



Guia Livre

Referência de Migração para Software Livre do Governo Federal

Versão Beta
V0.5

Disponibilizada para Comunidade Brasileira de Software Livre

**Software Livre:
um contrato aberto com o cidadão.**

Realização:



Guia Livre

Referência de Migração para Software Livre do Governo Federal

Versão Beta
0.5

Disponibilizada para Comunidade Brasileira de Software Livre

www.governoeletronico.gov.br
www.softwarelivre.gov.br

Guia Livre – Referência de Migração para Software Livre do Governo Federal

Organização:

Coordenação:

- Secretaria de Logística e Tecnologia da Informação – SLTI
Ministério do Planejamento, Orçamento e Gestão
- Comitê Técnico para Implementação do Software Livre – CISL
Comitê Executivo do Governo Eletrônico
- Comitê Técnico de Sistemas Legados e Licenças de Software – CTSLL
Comitê Executivo do Governo Eletrônico

Colaboração(*):

- DATAPREV
- SERPRO
- DATASUS
- RADIOBRÁS
- Ministério do Desenvolvimento Agrário
- Ministério da Integração Nacional
- Ministério das Comunicações
- Ministério da Ciência e Tecnologia
- Ministério da Defesa
- Ministério do Meio Ambiente
- Ministério da Saúde
- Ministério das Cidades
- Ministério do Desenvolvimento, Indústria e Comércio Exterior
- Ministério do Desenvolvimento Social e Combate à Fome
- Fundação Nacional de Saúde
- Instituto Nacional de Tecnologia da Informação
- Comunidade Brasileira de Software Livre.

(*): Instituições com representantes no Grupo de Trabalho de Migração para Software Livre

Agradecimentos

Este é um trabalho coletivo, união de esforços de 39 pessoas que integram os Comitês Técnicos para Implementação do Software Livre e de Sistemas Legados e Licenças de Software e a estrutura da Secretaria de Logística e Tecnologia da Informação.

Ficam aqui registrados os agradecimentos a cada um(a) deles(as).

Guia Livre – Referência de Migração para Software Livre do Governo Federal

O presente Manual foi elaborado por grupo de trabalho interinstitucional constituído em agosto de 2003 por deliberação conjunta de dois Comitês Técnicos do Governo Eletrônico: Implementação de Software Livre e Sistemas Legados e Licenças de Software, homologados em Decreto Lei no dia 29 de outubro de 2003 pelo Presidente da República. O Grupo tem como objetivo prioritário formular orientações para a migração para software livre de órgãos integrantes da Administração Pública Federal, em consonância com diretrizes dos comitês técnicos citados.

Embora originalmente o escopo das atividades do grupo estivesse restrito a definições referentes ao ambiente de estações de trabalho, foi percebido que para atender efetivamente às demandas dos órgãos seria necessário tratar a migração em todas as “camadas” dos ambientes computacionais. Desta forma, os integrantes do grupo – que representam percentual expressivo dos órgãos da APF que iniciaram a migração – se concentraram na elaboração do presente documento, tendo como referência básica o *Guia do IDA (Comunidade Européia) em sua versão 02* e com base empírica em experiências reais de mudança em que seus participantes estiveram ou estão envolvidos.

Este é o contexto geral de elaboração do presente documento, que visa ser uma referência para processos de Migração para Software Livre no Governo Federal, bem como em qualquer outro nível de governo ou esfera de poder, que porventura necessitem utilizar tal material como base ou desejarem planejar e executar seus processos de migração com base em “estratégias já implementadas”.

0.1. História do Documento

Data	Versão	Autor	Alteração
11/01/04	0.0	SLTI	Tradução para português do Guia com as Diretrizes do IDA para Migrações para Fonte Aberta
07/04/04	0.1	Grupo de Trabalho Migração para Software Livre (GT-MSL)	Alterações e reorganização da Parte 1 e 2
07/05/04	0.2	13 ^a . Reunião GT-MSL	Alterações e reorganização da Parte 1, 2 e 3.
21/05/04	0.3	14 ^a . Reunião GT-MSL	Alterações e reorganização da Parte 1, 2, 3 e 4.
28/05/04	0.4	Contribuições individuais das instituições	Alterações e reorganização da Parte 1, 2, 3 e 4.
02/06/04	0.5	15 ^a . Reunião GT-MSL	Reorganização e alteração para o Lançamento da Versão Beta no V Fórum Internacional do Software Livre
	0.6 a 0.8		Comentários, correções e inclusões enviadas pela comunidade
22/06/04	0.9	SLTI / ITI Cisl e CTSLL	Consolidação e publicação para Consulta Pública

Nota Inicial da Comissão de Redação

O conteúdo deste documento expressa a visão consensual de técnicos e gerentes de informática, que integram o Grupo de Trabalho “Migração para Software Livre”, formalmente instituído no âmbito dos Comitês Técnicos Implementação de Software Livre e Sistemas Legados e Licenças de Software.

As versões futuras serão publicadas depois de finalizados os processos de submissão a estes Comitês e ao Comitê Executivo do Governo Eletrônico. Após esta fase passarão a expressar a posição do Governo Brasileiro sobre o assunto. Consulta pública será realizada para viabilizar a complementação dos conteúdos apresentados pela comunidade de software livre brasileira.

Embora todos os cuidados tenham sido tomados para minorar as imprecisões nas informações publicadas, pede-se que na eventual identificação deste tipo de ocorrência a comissão de redação seja informada pelo e-mail guialivre@planejamento.gov.br.

A comissão de redação buscou atender a todos titulares de direitos autorais de partes de documentos originais utilizados, em especial os do Guia do IDA Versão 2, fonte primária para a elaboração do presente texto.

Nota técnica para edição da Comunidade Brasileira de Software Livre

Esta versão Beta (0.5), aberta a contribuições da comunidade até o dia 19/06/04, apresenta-se com possíveis inconsistências técnicas, em especial desatualizações quanto ao estado da arte das soluções livres, que possui uma dinâmica acelerada no desenvolvimento de soluções.

A equipe técnica responsável conta com a colaboração da comunidade para suprir tais lacunas, originadas pela complexidade e abrangência do conteúdo do Guia Livre.

0.2. Distribuição

Secretaria de Logística e Tecnologia da Informação.
Instituto Nacional de Tecnologia da Informação.

Versão 0.5
Versão 0.5

0.3. Marcas Registradas

Foram usadas marcas registradas neste documento somente com o propósito de identificação. Os autores reconhecem a propriedade dessas marcas registradas.

0.4. Direitos Autorais

Governo Brasileiro - a reprodução é autorizada desde que a fonte seja reconhecida, de acordo com as orientações da CC-GPL.

0.5. Índice

1	Prefácio	12
1.1	Abreviações e Terminologia	12
1.2	Público	12
1.3	Autores	13
1.4	Agradecimentos	13

PARTE 1 **Introdução e Sumário**

2	Introdução	15
3	Sumário	16
4	Porque o Software Livre é livre e quais as razões jurídicas para migração?	19
5	Metodologia	20

PARTE 2 **Diretrizes de Gestão**

6	Visão Geral da Migração	23
7	Questões Humanas	28
8	Facilitando a Vida	29
8.1	Introdução de novos aplicativos em um ambiente familiar	29
8.2	Faça primeiro as coisas fáceis	30
8.3	Pense Além	30

PARTE 3 **Diretrizes Técnicas**

9	Arquitetura de Referência	33
9.1	Arquiteturas Genéricas	33
9.2	Arquitetura Básica de Referência	36
10	Grupos Funcionais	37
10.1	Sistema Operacional	38
10.2	Estação de Trabalho	39
10.2.1	Gerenciadores de Janelas	39
10.2.2	Escritório	40
10.2.3	Correios	43
10.2.4.	Calendários e Groupware	44

10.2.5.	Navegador	44
10.2.6.	Banco de Dados	45
10.3	Servidores	46
10.3.1.	Serviço de Correio	46
10.3.2.	Serviço de Anti-Vírus	48
10.3.3.	Serviços de Calendário e Groupware	48
10.3.4.	Serviços de Web	52
10.3.5.	Serviço de Gestão do Documento	53
10.3.6	Serviço de Bancos de Dados	54
10.3.7	Serviços de Segurança	55
10.3.8	Serviços de Gestão	56
10.3.9.	Serviço de Backup e Recuperação	63
10.3.10.	Outros Serviços	63
11.	Migração de Aplicativo – Visão Geral	67
11.1	Aplicativos proprietários que possuem um software livre equivalente	67
11.2	Aplicativos proprietários que operam em um ambiente software livre	67
11.3	Software que pode ser acessado por uma exibição remota	67
11.4	Software que funcionará em um emulador	68
11.4.1	Emulação de Hardware	68
11.4.2	Emulação de Software	69
11.5	Software que pode ser recompilado em software livre	70

PARTE 4 Planejando a Migração

12.	Cenário 1 – Windows	73
12.1	Como Planejar a Migração	73
12.2	Domínios	73
12.2.1	Modelo de “grupo de trabalho” Windows	73
12.2.2	Domínio Windows NT	74
12.2.3	Domínio do Active Directory Windows 2000	74
12.3	Visão Geral de possíveis rotas de migração	74
12.4	Questões Gerais	76
12.4.1	Nomes de usuário e Senhas	77
	Questões de Nomes de usuários	77
	Questões de Senhas	77
12.4.2	Serviços de Autenticação	78

12.4.3	Arquivos	78
	Conteúdo e formato	78
	Nomes de arquivos	78
	Acesso Dual	79
12.5	Ferramentas	81
12.5.1	Samba	81
12.5.2	OpenLDAP	82
12.5.3	NSS e PAM	82
12.5.4	Acesso a arquivos GNU/Linux SMBFS	83
12.5.5	Winbind	83
12.6	Migrando o ambiente do sistema operacional	83
12.6.1	Acrescentar servidores GNU/Linux individuais a um domínio Windows NT existente	83
12.6.2	Executar estações de trabalho GNU/Linux em domínios Windows NT	84
	Configuração simples para um número pequeno de máquinas	84
	Configuração mais inteligente para distribuições mais amplas	86
12.6.3	Executar estações de trabalho GNU/Linux em domínios Active Directory	87
12.6.4	Substituir o Windows NT PDC/BDC por Samba+LDAP	88
12.6.5	Substituir o Active Directory Windows 2000 por LDAP	89
12.6.6	Executar uma infraestrutura GNU/Linux paralela e migrar usuários em grupos	89
	Substituindo todos os clientes Windows por GNU/Linux	89
	Mantendo alguns clientes Windows	90
12.7	Migrando aplicativos tipo servidor	91
12.7.1	Servidores de Web: mudando do IIS para o Apache	91
	Questões de Migração	91
	Migrando um sítio Internet estático	93
	Uma configuração WebDAV simples	94
12.7.2	Bancos de Dados: do Access e Servidor SQL para MySQL ou PostgreSQL	95
	Migrando Bancos de Dados Access	96
	Migrando Bancos de Dados do Servidor SQL	97
	Questões de migração de Bancos de Dados	97
12.7.3	Groupware: mudando do Exchange	98
	Questões Gerais	98
	Questões de Correio	98

Guia Livre – Referência de Migração para Software Livre do Governo Federal

	Questões de Catálogo de endereços	99
	Questões de Calendários	99
12.8	Migrando aplicativos de estação de trabalho para software livre	99
12.8.1	Office	99
	Conversão de Documento	99
	Conversão de Modelo	99
	Conversão de Macro	100
	Processamento de Texto	100
	Publishing	101
	Planilhas	101
	Gráficos de Apresentação	101
	Manipulação de gráficos e imagens	102
	Geração de PDF	102
12.8.2	Correio	103
12.8.3	Calendários e Groupware	103
	Calendários	104
	Gestão de Contatos	104
12.8.4	Navegação Internet	104
12.8.5	Bancos de Dados Pessoais	105
12.9	Migrando serviços de impressão para software livre	105
12.9.1	O modelo de impressão no ambiente proprietário	106
12.9.2	O modelo de impressão Unix e GNU/Linux	106
12.9.3	Configurando um serviço de impressão baseado em software livre	107
12.9.4	Imprimindo a partir de clientes Windows para impressoras GNU/Linux anexas	107
	Usando o protocolo lpr	108
	Usando cotas de impressoras	108
	Usando configuração Point and Print	109
12.9.5	Imprimindo esquemas de migração	109
12.9.6	Problemas Potências	110
12.9.7	Informações adicionais sobre impressão	110
12.10	Aplicativos legados	110
12.11	Proteção antivírus	110
12.12	Referências	111
13.	Cenário 2 – Unix	112

Guia Livre – Referência de Migração para Software Livre do Governo Federal

14	Cenário 3 – Mainframe	114
15	Cenário 4 – Cliente Leve	115

APÊNDICES

Apêndice A – Estudos de Casos	115
Apêndice B – Wine	127
Apêndice C – Sistemas de Correio	131
Apêndice D – Glossário	138
Apêndice E – Licença CC-GPL	145

Prefácio

1.1 Abreviações e terminologia

Sempre que possível, na primeira vez que uma abreviação for usada, será incluída também a versão por extenso. No Apêndice D encontra-se um glossário de termos. Quando um termo de glossário aparecer pela primeira vez, estará com uma breve descrição na nota da própria página, seguido pelo termo: **Glossário**.

Cada um dos termos Software de Fonte Aberta (Open Source Software) e Software Livre (Free Software) têm seus defensores. Neste relatório, usaremos o termo Software Livre na intenção de destacar as características que o diferenciam do Software de Fonte Aberta, especialmente a sua disponibilização em forma de Licença Pública Geral (GPL).

Dessa forma, os dois termos: Software de Fonte Aberta e Software Livre, serão tratados separadamente ou conjuntamente, conforme a necessidade deste Guia.

Para mais informação sobre esses termos, veja:

<http://www.fsf.org/home.pt.html>
<http://www.gnu.org/philosophy/free-sw.pt.html>
<http://www.gnu.org/licenses/license-list.pt.html>
<http://www.opensource.org>
<http://www.sourceforge.org>

Nomes de produtos serão apresentados desta forma: Nome de Produto.

Termos do Sistema Operacional, como nomes de arquivos, serão apresentados desta forma: **Nome de arquivo**.

Código de programa será apresentado desta forma: **Código**.

1.2 Público

Este documento é dirigido aos Gerentes de Tecnologia da Informação (TI) de todo Governo Federal Brasileiro, em todas as esferas: executivo, legislativo e judiciário, servindo também como referência para os governos estaduais e municipais. O termo **Administração** é usado em todo o documento com relação à Administração Pública, e o termo **Administradores** refere-se ao grupo de pessoas descrito acima.

1.3. Autores

O relatório foi produzido por grupo de trabalho Migração para Software Livre cujos integrantes já foram relacionados no início deste documento.

Teve como referência básica à estrutura e parte dos conteúdos constantes no guia “Diretrizes do IDA* para Migrações para Fonte Aberta”, desenvolvido pela *netprojetc*, para a Comunidade Européia.

1.4. Agradecimentos

O Governo Brasileiro gostaria de agradecer à Comunidade Européia pelo trabalho que subsidiou a elaboração do Guia de Referências das Migrações para Software Livre.

(* Intercâmbio de Dados entre Administradores

PARTE 1
Introdução
e Sumário

2. Introdução

As diretrizes apresentadas neste trabalho foram objeto de estudo do grupo de trabalho Migração para Software Livre do Governo Federal, com participação da comunidade Software Livre Brasileira, na intenção de construir um guia cujo respaldo alcançasse também qualquer entidade interessada em promover equivalentes projetos de migração.

Os objetivos destas diretrizes são:

1. Ajudar os Administradores a definir uma estratégia para migração planejada e gerenciada.
2. Descrever, em termos técnicos amplos, como pode ser realizada tal migração. As diretrizes pretendem ter um uso prático para Administradores e portanto, devem ser relevantes e precisas, além de acessíveis e compreensíveis. Este não é um manual de referências técnicas detalhadas. A estrutura pretende tornar possível e facilitar as mudanças à proporção em que os administradores adquiram experiência, tenham segurança e os produtos disponíveis atendam suas necessidades.
3. Orientar o conjunto de diretrizes e definições deste Guia aos Padrões de Interoperabilidade do Governo Brasileiro (e-PING), cujas informações detalhadas podem ser obtidas em <http://www.governoeletronico.gov.br>.
4. Criar condições para um maior detalhamento técnico destas migrações na página do governo federal do software livre: <http://www.softwarelivre.gov.br>.

Para alcançar esses objetivos, é imperativo que o conteúdo seja mantido atualizado e que quaisquer imprecisões sejam removidas. Para isto, os leitores são encorajados a tecer comentários e contribuições a qualquer item das diretrizes. Por favor, encaminhem seus comentários para guialivre@planejamento.gov.br

As diretrizes não se referem ao *software livre* em termos gerais ou aos méritos relativos das várias licenças. Estas informações, juntamente com uma grande quantidade de outras informações norteadoras, podem ser encontrados nos seguintes sítios do Governo Federal:

<http://www.governoeletronico.gov.br>
<http://www.softwarelivre.gov.br>

3. Sumário

Estas diretrizes dirigem-se aos Gerentes de TI e aos profissionais que estejam planejando ou realizando uma migração para o *software livre*. São baseadas nas experiências práticas dos autores e em um número crescente de estudos de casos publicamente conhecidos, que foram validadas com a migração concreta de cada órgão.

Há diversas razões para que instituições públicas estabeleçam programas de migração para software livre, em especial:

- necessidade de adoção de padrões abertos para o Governo Eletrônico (e-Gov);
- nível de segurança proporcionado pelo software livre;
- eliminação de mudanças compulsórias que os modelos proprietários impõem periodicamente a seus usuários, face à descontinuidade de suporte a versões;
- independência tecnológica;
- desenvolvimento de conhecimento local;
- possibilidade de auditabilidade dos sistemas;
- independência de fornecedor único.

Esses benefícios, agregados ao fato de que despesas referentes a licenças de uso não são aplicáveis a soluções baseadas em software livre e resultam em economia progressiva para seus usuários, sendo que consideramos ser de grande importância, que tais valores sejam reaplicados em investimentos para Tecnologia da Informação.

Isto posto, vale destacar a linha geral das diretrizes recomendadas para qualquer processo de migração para software livre:

- antes de começar, ter um claro entendimento sobre as razões para a migração;
- assegurar-se de que exista apoio ativo da equipe e dos usuários de TI para a mudança;
- certificar-se de que existam defensores da mudança, quanto mais altos na hierarquia da organização, melhor;
- formar peritos e construir relacionamentos com a comunidade do movimento Software livre;
- começar com sistemas não críticos;
- garantir que cada passo da migração seja administrável;
- criar canais de comunicação e bases de conhecimento internos na instituição.

Note-se que a mudança para software livre deve ser vista como um tipo de migração de sistemas de TI. Portanto, são aplicáveis estas migrações, desafios e possibilidades já experimentadas por todo gerente de informática. Em especial, a migração de sistemas de TI proporciona a oportunidade de realizar a reengenharia dos mesmos, para satisfazer às novas demandas a eles propostas. As questões pertinentes incluem:

- como garantir a interoperabilidade dos sistemas;
- como dar suporte aos usuários móveis;
- como identificar usuários remotos de forma segura;
- como construir sistemas administráveis.

Guia Livre – Referência de Migração para Software Livre do Governo Federal

Acima de tudo, como certificar-se de que a segurança seja planejada desde o início, e não acrescentada como uma questão posterior.

No caso específico de migrações para software livre é importante destacar que as decisões referentes a servidores têm diferenças significativas em relação às referentes a estações de trabalho. Afinal, para utilização em servidor, o *software livre* é bem estável e já empregado largamente. A migração de servidores para o *software livre* pode ser feita, em termos gerais, sem qualquer efeito adverso para os usuários. É, normalmente, por onde se começa.

Já o uso de distribuições baseadas em *software livre nas estações de trabalho* envolve maiores desafios, embora potencialize, para a maioria das organizações, uma maior economia de custos. Os principais desafios em migrar *estações de trabalho*, estão na necessidade de que aplicativos *livres* têm que interagir com os aplicativos existentes. Deve ser objeto de atenção, particularmente, a forma como ferramentas de trabalho em grupo (groupware) interagem com os ambientes baseados em *software livre* e proprietário.

Adicionalmente, na substituição da automação do escritório proprietário, os *documentos modelo* devem ser verificados para garantir que produzam o produto correto. As macros devem ser reescritas, preferivelmente como *scripts*. Aplicativos para os quais não haja equivalentes em *software livre* podem funcionar como *thin clients* (veja mais seção 9.1). Com o decorrer do tempo, aplicativos de estação de trabalho poderão ser crescentemente substituídos por equivalentes livres.

Embora as diretrizes pretendam uma mudança completa para o *software livre*, a possibilidade é que um ambiente heterogêneo seja construído, particularmente por que uma migração de milhares de estações de trabalho levará tempo. A mescla de aplicativos *software livre* e proprietários também é provável, porque aplicativos software livre para substituição total podem ainda não estar disponíveis ou não ser apropriados. Nesse momento, isto é particularmente verdadeiro na substituição da função intercâmbio de ferramentas de trabalho em grupo proprietário, em larga utilização nas organizações. No entanto, há soluções de software livre com qualidade suficiente para impulsionar a migração também destas ferramentas.

Em qualquer das decisões que se tome, no entanto, é importante assegurar-se de que as escolhas, mesmo que não estejam diretamente relacionadas à migração, não amarrem a Administração, no futuro, aos formatos proprietários, seja de arquivos ou de protocolos.

É recomendável refletir quanto ao caráter diferenciado do desenvolvimento de software livre, o qual possibilita uma mudança fundamental na forma como as organizações realizam serviços de TI. É uma mudança de uma indústria baseada no produto, para uma indústria baseada no serviço. O *software livre* tem nitidamente custo menor de instalação. A questão é como formar profissionais em larga escala para o suporte, pois estes estão acostumados com os produtos proprietários. Existe um certo número de companhias de prestação de serviços, bem como os representantes de distribuições. No entanto, se a sua postura diante da TI é “Quem eu vou processar se as coisas derem erradas?”, então talvez você precise refletir melhor sobre as reais condições de responsabilização por problemas que tem hoje, utilizando software proprietário. Os mitos de que ambientes de TI baseados neste modelo de licenciamento sejam “gerenciáveis” estão profundamente construídos em nossa sociedade.

No entanto, são mitos. Conforme assinala Hexsel*:

O simples fato de existir um proprietário do software, e portanto legalmente imputável, não provê necessariamente garantia quanto a prejuízos decorrentes de erros ou falhas nos sistemas. Pelo contrário, freqüentemente o proprietário se exime de qualquer responsabilidade por danos ou prejuízos decorrentes da utilização correta de seus produtos.

Guia Livre – Referência de Migração para Software Livre do Governo Federal

Por outro lado, é necessário um entendimento da dinâmica do movimento do software livre e seu funcionamento. É aconselhável saber como relacionar-se com a comunidade do software livre e usufruir dos benefícios de um novo modelo de negócios. Você verá que riscos existem nos dois modelos, mas que as condições efetivas para gerenciá-los – mesmo que não atendidas pelos modelos de comercialização instituídos e praticados – estão mais acessíveis em ambientes baseados em software livre.

HEXSEL, Roberto André. Propostas de Ações de Governo para Incentivar o Uso de Software Livre. Curitiba, UFPR 2002. Relatório Técnico RT-DINF 004/2002. Disponível em <http://www.inf.ufpr.br/~roberto>. Último acesso em 31 de maio de 2004.

4. Por que o Software Livre é livre e quais as razões jurídicas para migração?

Software livre não é um tipo diferente de software. Não é uma espécie distinta dentro do gênero software.

Internamente, em sua arquitetura, o que chamamos de software livre não tem uma substância técnica diferente daquilo que chamamos de software proprietário. O modelo do desenvolvimento do que chamamos de software livre - colaborativo, compartilhado - e da transmissão de direitos sobre ele é que são diferentes.

A lei diz que programa de computador é a expressão de um conjunto organizado de instruções em linguagem natural ou codificada, contida em suporte físico de qualquer natureza, de emprego necessário em máquinas automáticas de tratamento da informação, dispositivos, instrumentos ou equipamentos periféricos, baseados em técnica digital ou análoga, para fazê-los funcionar de modo e para fins determinados

Essa definição não muda, caso o software seja livre ou caso o software seja proprietário.

Então, se em qualquer momento durante sua leitura deste guia você se deparar com alguma referência a software livre como um produto distinto, entenda que essa referência não foi intencional e por favor nos avise, pois iremos mudá-la.

O que faz um software ser livre para o governo é a forma como os direitos sobre ele são adquiridos ou transmitidos pelo governo. Então, quando o governo “contrata software livre”, ele não está dando preferência a um tipo de programa ou a alguma empresa.

O que ele está fazendo é contratando de uma forma melhor para o cidadão, para o país e para todo o mundo que se beneficia do compartilhamento das informações que existem no código do programa. Contratando de uma forma que permitirá a transparência sobre inúmeros atos do governo que são praticados com o auxílio de programas de computador.

O governo ter acesso ao código do seu programa (além de pagar muito menos por ele) tem a ver com eficiência, com independência e com soberania. O cidadão ter acesso ao código do programa de computador que o governo usa tem a ver com democracia e com cidadania. E todo mundo poder usar as informações desse programa e criar com base nelas tem a ver com o desenvolvimento da economia e da sociedade.

E isso tudo só acontece por causa do contrato, que faz com que o software seja livre.

É importante então que você envolva os advogados ou procuradores do órgão ou instituição em que você trabalha no processo de migração. Converse com eles sobre este guia. Mostre a eles este capítulo, em particular. Sempre que você for instalar um programa de computador, converse com o jurídico sobre a licença.

Enfim, o que se deve perceber é que o governo terá sempre à sua frente duas formas de contratar distintas. Uma em que o governo e o cidadão preservam mais direitos - direitos inerentes à democracia - e outra em que o governo e o cidadão abrem mão desses mesmos direitos.

Guia Livre – Referência de Migração para Software Livre do Governo Federal

São dois modelos contratuais distintos. E adotar um ou outro modelo não é uma opção para o governo. É, ao contrário, um dever. O governo tem o dever de contratar preservando os valores de liberdade e de abertura. O governo tem o dever de contratar da forma melhor para o cidadão.

Se pudéssemos resumir a política governamental em relação a software livre em uma única frase, diríamos:

“Software Livre: Um Contrato Aberto com o Cidadão”

5. Metodologia

Qualquer exercício de migração deve constituir-se, em termos gerais, de:

1. Uma fase de coleta de dados e definição de projeto, incluindo:

A. Uma descrição das condições iniciais relevantes que consistem, por exemplo, de:

- a. arquitetura de sistemas,
- b. aplicativos e os dados a eles associados,
- c. protocolos e padrões usados,
- d. *hardware*,
- e. ambiente físico, como largura de banda da rede, localização,
- f. requisitos sociais tais como idioma(s) e conjunto de habilidades do pessoal.

B. Uma série de condições alvo detalhadas da mesma forma,

C. Uma descrição de como passar das condições existentes para as planejadas;

2. Uma justificativa para a migração, incluindo os benefícios e o custo a ela associado;

3. Uma ou mais fases-piloto, projetadas para testar o plano e as justificativas. Os dados desses pilotos podem ser realimentados no modelo de custo usado no plano;

4. Acompanhamento do plano;

5. Monitorar a experiência atual junto ao plano.

O conteúdo do item 1 define o que neste Guia chamamos de *Cenário* e as diretrizes descrevem como migrar para o *software livre* em tais circunstâncias.

No entanto, para que as diretrizes fossem legíveis e úteis na prática, foram necessários alguns pressupostos simplificadores, caso contrário, o número total de possíveis combinações inviabilizaria o trabalho.

Guia Livre – Referência de Migração para Software Livre do Governo Federal

Escolhemos uma das muitas condições alvo (1.B) e simplificamos a descrição das condições iniciais (1.A). O ambiente alvo é discutido na Seção 8.2. Com o ambiente alvo padrão assumido, fica definido um Cenário, por referência às condições iniciais simplificadas e pelo caminho de migração destas até o alvo.

Caso necessário, no futuro, novos capítulos podem ser adicionados a este documento. A partir da Parte 4, cada capítulo oferece uma descrição razoavelmente detalhada de um Cenário, descrevendo como migrar para o alvo, inclusive com a discussão de migrações parciais. Cada capítulo deve ser atualizado a partir da experiência com migrações reais.

O detalhamento disponível no levantamento de estudos de casos disponibilizados ao público era muito amplo. Foi encontrado um grande número de tais estudos (há uma lista no Anexo A), mas com pouca documentação, que oferecessem maior detalhamento, restando somente a divulgação de informação genérica. Isto significa que a maior parte das diretrizes é baseada nas experiências do Governo Federal Brasileiro e em outros níveis da federação, e suas discussões com pessoas/órgãos que realizaram uma migração.

O enorme número de combinações diferentes de condições iniciais e finais de cenários, juntamente com as muitas e variadas formas de passar de umas para outras, demonstra que é impossível, para qualquer conjunto de diretrizes, cobrir todas as possibilidades. As diretrizes devem então ser consideradas mais como indicativas e referenciais do que pode ser feito, do que prescritivas do que deveria ser feito. Elas devem ser usadas como um ponto de partida no processo de migração. Não se pode esperar que ofereçam uma resposta para todas as circunstâncias.

Parte-se do princípio que a migração tem um alvo que é um ambiente totalmente *software livre* onde for possível e sensato; no entanto, podem haver razões para que sistemas proprietários devam ser mantidos ou utilizados. A possibilidade de uma migração parcial também é discutida e justificada.

PARTE 2
Diretrizes e
Gestão

6. Visão Geral da Migração

Muito do que precisa ser feito para migrar de um ambiente proprietário para software livre, é semelhante a qualquer migração. Até mesmo na migração de um ambiente tecnologicamente idêntico e/ou de um mesmo fornecedor, não se pode pressupor que os formatos dos arquivos serão compatíveis e sempre haverá necessidade de testes apropriados antes de proceder a qualquer mudança mais difundida. Todas as migrações precisam ser baseadas em um cuidadoso planejamento.

As diretrizes aqui presentes não pretendem ser um manual sobre gestão de projeto e supõe-se que a Administração tenha habilidade suficiente para gerenciar a migração de forma apropriada. A descrição adiante pretende realçar os pontos importantes de uma migração para software livre.

O processo de migração deveria, em tese, consistir das partes que se seguem. Algumas delas podem ser feitas em paralelo, tais como a 2, 3 e 4, ou de acordo com o planejamento de cada instituição.

Note que a informação encontrada pode indicar que serão necessárias modificações no ambiente atual, antes de ser concebida a migração para o software livre. Esta é a razão pela qual até mesmo administrações que não têm planos imediatos para migração, mas que desejam manter esta opção disponível, são instruídas a solicitar somente padrões multiplataforma e abertos e a avaliar sua infra-estrutura em comparação a esses padrões (veja também Seção 7.3 - “Pense Além”).

1. Crie uma equipe habilitada e com apoio gerencial. É importante que haja apoio gerencial, caso contrário haverá resistência para sair do modelo dos sistemas proprietários. Esse suporte deverá possibilitar que se construa, no mínimo, pilotos representativos; portanto, terá que ser produzido um relatório de implementação / plano de trabalho, e talvez algum documento mais detalhado quando houver mais dados disponíveis.

2. Entenda o ambiente alvo, tanto o software de livre quanto a arquitetura básica (veja Capítulo 8), junto com as várias opções e escolhas disponíveis. Isto significa treinar a equipe existente, recrutar ou utilizar consultores, o que vai demandar um custo inicial e, portanto, requerer suporte gerencial suficiente. Há, por vezes, a expectativa de que o software livre pode ser compreendido e utilizado sem custos, o que pode causar inconsistências nos custos planejados.

3. A migração é uma oportunidade para rever a arquitetura base, bem como os aplicativos. A arquitetura recomendada no Capítulo 8 é baseada no controle centralizado e possui algumas vantagens discutidas no mesmo capítulo. Esta mudança pode implicar em custos, e estes devem ser considerados.

4. É muito importante que se entenda a “filosofia” do software livre. Algumas questões precisam ser bem consideradas antes de tomar-se qualquer decisão:

Onde houver várias opções para cada uma das funções - é necessário que os administradores conheçam os prós e os contras de cada produto, para que se possa optar pela solução que melhor atenda as suas necessidades. Neste Guia serão sugeridos os parâmetros que devem ser avaliados para cada uma das opções disponíveis, seguidos pelas opções de produtos que já foram conhecidas na prática em experiências da Administração e, finalmente, um ou mais casos de sucesso relacionados com estas ferramentas.

Guia Livre – Referência de Migração para Software Livre do Governo Federal

As diferenças entre as várias distribuições dos sistemas operacionais livres devem ser consideradas. Algumas são desenvolvidas por empresas que oferecem suporte e reparos. Outras têm características distintas para essas mesmas questões; as diferenças devem ser avaliadas antes de se fazer uma escolha.

Os Administradores devem determinar o nível de suporte necessário. Pode-se obter suporte comercial com os responsáveis pelo desenvolvimento do aplicativo ou com os distribuidores, no caso de oferecerem tal suporte. Caso não ofereçam, pode-se consegui-lo com serviços de terceiros, pois o código fonte é disponibilizado e há muitas companhias oferecendo seus serviços para tais distribuições.

A questão do suporte é uma diferença bem clara em relação ao mercado de software proprietário, onde somente as empresas que tem o privilégio de acesso à fonte podem fornecer o suporte com maior nível de profundidade – isto se torna importante caso o revendedor proprietário deixe o negócio sem liberar o código fonte. Com a adoção de Software Livre, as organizações têm acesso e controle ao código fonte, e dessa forma adquirem autonomia para negociar com qualquer empresa que preste este serviço.

Além disso, a maior parte desses aplicativos tem Listas de Discussão ativas, onde um pedido de ajuda será respondido por alguém interessado no aplicativo. A presença de uma Lista de Discussão ativa e de uma comunidade de usuários é, freqüentemente, um dos primeiros critérios na seleção de componentes do software.

5. Faça uma auditoria nos sistemas existentes. Essas informações serão necessárias, não somente para fazer a migração, mas em grande parte, para construir um modelo de custo de propriedade para um plano / relatório de migração detalhado.

Faça o inventário:

A. Para cada aplicativo usado:

- a. O nome do aplicativo, número da versão e contato para responder a questões relacionadas ao aplicativo;
- b. A quantidade de usuários que requerem acesso e acesso simultâneo ao aplicativo;
- c. Com quais sistemas operacionais o aplicativo pode ser usado, considerar todos os ambientes;
- d. Quais outros aplicativos são necessários, tanto no cliente como no servidor, para o aplicativo funcionar – pré-requisitos e/ou dependências;
- e. O hardware exigido, considerando se é necessário algum equipamento fora dos padrões ou de última geração – alta performance, etc.
- f. Os protocolos e portas usados para comunicar-se com outros aplicativos;
- g. os formatos de arquivos requeridos e gerados.

Guia Livre – Referência de Migração para Software Livre do Governo Federal

B. Requisição de Dados:

Deve ser interpretada no sentido amplo, o que inclui, por exemplo, documentos de processador de textos e planilhas, dados som/voz e de imagem, bem como os bancos de dados habituais. Em geral, qualquer coisa que se pretenda processar em um computador.

a. Quais são as dificuldades na interface com sistemas externos, ou usuários fora dos quadros de funcionários da Administração?

b. Quais os requisitos para guardar os dados e poder processá-los no futuro? Há um repositório de dados legados ao qual se tenha que dar suporte? Caso positivo, há necessidade de aplicativos específicos para acessá-los (caso estejam armazenados em mídias cuja acessibilidade é restrita a determinado sistema ou aplicativo) e processá-los?

i. Divida os dados nas seguintes categorias:

a) Dados que não precisam ser mantidos e podem ser eliminados. Descarte-os. Tenha cuidado de fazer o back-up para o descarte no período adequado.

b) Dados que precisam ser mantidos e que se encontram normalmente em formato aberto, como PDF ou Postscript, ou que podem ser facilmente convertidos para formato aberto. Neste caso, o custo da conversão deve ser avaliado.

c) Dados que precisam ser mantidos, mas que estejam em um formato proprietário fechado, que não permite fácil conversão para formato aberto. Esses dados podem necessitar de cópias do aplicativo proprietário específico para serem mantidos. O custo desses aplicativos deve ser avaliado. O número de cópias desse aplicativo pode ser determinado pelo grau de acesso necessário aos dados. Por exemplo, se os dados forem raramente acessados, uma única cópia em uma máquina central será suficiente. Também pode ser que haja necessidade de manter um hardware específico para usar esses aplicativos. Finalmente, para tanto, pode-se considerar a utilização de características do modelo cliente leve – thin client (veja na Seção 9.1).

C. Requisitos de Segurança

a. Qual é o sistema atual para criação de usuários e senhas? Há uma estrutura para os nomes dos usuários? Caso positivo, qual é ela? Está de acordo com a norma apontada na e-PING? Qual é a política para alteração de senhas?

b. Há sistemas que requerem alguma outra autenticação além de um simples nome de usuário e da senha?

c. Quais as políticas Administrativas e de Governo existentes com relação ao uso de computadores? Existem normas internas específicas? Por exemplo, há restrições ao uso da Internet e do correio eletrônico?

d. Há planos de segurança que requerem o uso de hardware ou software específico?

6. Faça um cenário para migração detalhado. Ele deverá basear-se nos dados compilados nas etapas sugeridas nos itens anteriores e consistirá de algumas seções, inclusive:

Guia Livre – Referência de Migração para Software Livre do Governo Federal

- A. O custo do ambiente existente durante um período razoável de tempo, tal como 3 anos, com os pressupostos apropriados à Administração.
- B. O custo de ambientes alternativos, bem como o custo da migração para cada um, ao longo do mesmo período.
- C. Os pontos fortes e fracos do ambiente atual e as várias alternativas.

7. A consulta aos usuários pode ser um elemento favorável. Explique as razões da migração e os efeitos sobre eles. Considere as preocupações informadas com seriedade e permita que eles utilizem a tecnologia o mais breve possível. Quanto mais rápido se envolverem, melhor. Isto pode ser uma exigência em alguns órgãos, porém deve ser realizado em qualquer caso, para facilitar a introdução do que pode vir a ser uma mudança significativa nas práticas de trabalho.

Crie uma central de atendimento que responda às dúvidas dos usuários. Mais tarde, quando a migração estiver estabelecida, essa Central poderá responder a problemas e tornar-se um centro de excelência e boas práticas. Crie um sítio na rede interna com uma Seção de Dicas e um “Como Fazer”, que pode ser atualizado pelos próprios usuários (existem aplicações livres próprias para permitir esta interação). Isto é importante para que os usuários sintam-se incluídos e também porque o sítio dará ao pessoal do suporte técnico uma idéia dos tipos de problemas mais enfrentados por seus clientes.

8. Assumindo que o cenário foi feito, comece com projetos piloto de acordo com sua capacidade de atender às demandas geradas pelos projetos. Isto vai proporcionar, entre outras coisas:

- A. Dados para modelos mais refinados de Custo de Propriedade e Serviços.
- B. Opinião do usuário, que pode ser usada para facilitar a introdução de outros sistemas.
- C. Validação ou modificação da arquitetura alvo e do modelo de negócios.
- D. Aquisição de experiência ao longo do tempo.

9. Decida a forma /o modelo do processo de migração a partir do momento em que ele começar. As principais opções são:

A. Big bang: Todos os usuários mudam do sistema antigo para o novo ao mesmo tempo. Na prática, isto quer dizer, provavelmente, que a mudança deverá ser marcada para um final de semana ou feriado nacional. A vantagem é que não é preciso disposições para acesso a mais de uma plataforma e que o pessoal não precisará ficar trocando de um sistema para o outro, ficando o ambiente homogêneo. As desvantagens incluem o alto risco e a necessidade de recursos muito altos durante a mudança. Este esquema de migração provavelmente só será atrativo para pequenas Administrações.

De qualquer forma, se for possível, EVITE A MIGRAÇÃO BIG BANG. As migrações Big Bang têm tantas variáveis para se controlar que quase sempre falham, provavelmente por uma falha de gestão, e não do software livre, o que pode não ficar transparente.

Evitar a migração Big Bang não significa que você pode protelar indefinidamente a migração. Ela deve ser feita de maneira progressiva nos estritos limites em que esta progressividade for indispensável à continuidade das atividades do seu órgão ou instituição

Guia Livre – Referência de Migração para Software Livre do Governo Federal

B. Transição em fases por grupos: Os usuários mudam do sistema antigo para o novo em grupos. É provável que grupos funcionais completos sejam movidos juntos, para minimizar compartilhamento (perda de segurança..) de dados e problemas do trabalho em grupo. Os riscos podem ser contidos e os recursos administrados através da escolha do tamanho apropriado dos grupos. É possível aproveitar este momento para fazer alterações necessárias no hardware, com substituição gradual da estação de trabalho, ao mesmo tempo, fazendo a atualização das máquinas removidas de um grupo e instalando-as depois no lugar das máquinas antigas do outro grupo. Existe a desvantagem para ambientes heterogêneos: às vezes pode ser necessário tratar de cada ambiente separadamente.

C. Transição usuário por usuário: Essencialmente igual à opção de transição por grupo, porém com grupos de uma pessoa. Este método de alimentação gota a gota, requer poucos recursos, permite dimensionar o problema, mas cria ilhas no ambiente e é ineficiente e de interesse pouco provável para grandes Administrações. Pode, no entanto, ser uma forma apropriada de conduzir projetos piloto.

É provável que os sistemas antigos e novos tenham que funcionar lado a lado por algum tempo. É importante possuir uma estratégia de transição que possibilite aos sistemas antigos e novos trabalharem juntos, de forma que as atividades de produção possam seguir adequadamente durante o período de transição. A substituição da última máquina pode levar um longo tempo, portanto, é provável que a coexistência venha a ser muito importante.

10. Faça a migração chegar a toda a Administração. Isto envolverá treinamento adicional dos usuários e do pessoal técnico. Considere o treinamento do pessoal técnico primeiro, com repasse posterior do aprendizado aos demais, visando diminuir os gastos e as dificuldades.

11. Acompanhe o feedback do usuário e procure resolver quaisquer problemas que aparecerem. Algumas necessidades de usuários podem ser tão desconhecidas, que não será possível prevêê-las com antecedência ou descobri-las durante projetos piloto. Esteja certo de que haverá recursos suficientes para lidar com tais necessidades após a transição.

7. Questões Humanas

Estas diretrizes não têm o objetivo de ser um guia para Gestão de Recursos Humanos, e as Administrações já terão enfrentado muitas dessas questões em outras áreas antes. Elas deverão ter habilidade interna considerável para superá-las de forma solidária, e portanto, o pessoal de Recursos Humanos precisa estar envolvido no processo desde o início. A intenção aqui é simplesmente realçar os tipos de questões que surgiram em outras instituições que migraram para o software livre.

É muito importante que todo o pessoal seja consultado e mantido atualizado sobre o desenvolvimento do processo. Uma forma de fazer isso é criar uma intranet que possibilite ser atualizada facilmente e que possa ter uma seção para feedback do usuário. Novamente, existem diversas soluções em software livre que possibilitam este tipo de interação, com sistemas de votação, livro de visitas, etc.

O acesso ao treinamento é muito importante. Algumas instituições permitem aos usuários decidir se desejam prestar algum serviço, enquanto outros solicitam atendimento. A escolha dependerá da cultura da Administração e do assunto do curso de treinamento. Manuais e documentos em geral são normalmente em inglês e isso pode significar um problema para algumas pessoas da equipe. A tradução para a língua portuguesa pode ser considerada como um custo de migração, mas haverá o problema da tradução contínua das atualizações.

As interfaces de usuário do software livre, em particular Gnome e KDE, oferecem opção de idiomas, porém a tradução pode não ser completa, com alguns itens ainda em inglês. Além disso, nem todos os aplicativos terão suporte à *localização* plena. No entanto, tudo isso está mudando rapidamente e *a estrutura que permite o uso de outro idioma além do inglês* está disponível, se a Administração quiser usá-la. Quanto aos usuários com necessidades especiais, o Gnome, particularmente, tem boas características de acessibilidade para os deficientes visuais.

Há algumas reações clássicas a qualquer mudança nas práticas de trabalho, para as quais deverá haver um planejamento:

i) Medo do Desconhecido

O uso do software livre será completamente novo para a maioria dos usuários e para a equipe técnica. O medo natural do desconhecido, e a tendência em manter os sistemas existentes, fará com que as pessoas resistam ao software livre.

Haverá os usuários mais curiosos, que poderão ficar felizes por estarem experimentando o novo ambiente e são precisamente esses que deverão ser apresentados ao novo sistema em primeira instância. A experiência, até agora, indica que, a partir do momento em que as pessoas superam suas reservas, elas descobrem que o software livre não é usado de forma significativamente diferente do que o software proprietário, ficando bastante contentes por estarem usando-o. É provável, portanto, que esse grupo inicial de usuários mude para nova plataforma de forma entusiástica. De qualquer forma, também é provável que essas pessoas sejam aquelas que vão fornecer feedback útil.

O primeiro grupo de usuários poderá ser usado em projetos piloto, e tendo passado pela experiência, poderão incentivar e orientar seus colegas. Pode-se pensar na criação até mesmo de um grupo para estes usuários, com destaque dentro da Administração, para servirem como exemplo e quebrarem a barreira das pessoas mais relutantes.

Guia Livre – Referência de Migração para Software Livre do Governo Federal

De qualquer forma, na segunda fase, os usuários mais reservados terão que receber mais recursos de apoio, através de central de atendimento, intranet, usuários locais experientes e treinamento presencial.

O mesmo processo pode ser usado para a equipe técnica, porém é provável que o nível do treinamento seja significativo se o ambiente proprietário existente não for compatível com UNIX/Linux. A equipe técnica, particularmente, precisa ter seus temores logo apaziguados. Eles serão um ponto focal para todos os problemas que estão por acontecer e, se não acreditarem no projeto, não poderão encorajar os usuários de forma positiva.

ii) O efeito diluição de Currículo

Tanto a equipe técnica quanto os usuários poderão sentir que, por não usarem o software modelo da indústria, terão prejuízo em sua habilidade de desenvolverem-se na carreira. Este é um problema, que demanda um cuidadoso gerenciamento. A Administração não deve parecer distante dessa realidade em sua abordagem. Mas até que o software livre seja largamente utilizado, as Administrações deverão enfrentar este problema com bastante frequência.

iii) Conhecimento é poder

As pessoas que conhecem os sistemas e configurações existentes, tem um certo poder e podem demonstrar relutância em abrir mão do mesmo, se o ambiente do software livre for muito diferente do existente. Novamente, o problema requer gerenciamento cuidadoso já que essas pessoas cumprem um papel crítico no funcionamento dos sistemas existentes. Podem existir prioridades de treinamento nos novos sistemas, de forma que sua posição na organização se mantenha.

8. Facilitando a vida

Há algumas considerações que podem facilitar a introdução do software livre:

8.1. Introduza novos aplicativos em ambiente estável

Muitos dos aplicativos do software livre irão trabalhar em sistemas operacionais proprietários e isso proporciona a oportunidade de introduzir esses aplicativos sem ter que trocar totalmente o ambiente.

Por exemplo, o pacote de automação de escritório OpenOffice.org, o navegador Mozilla e o servidor web Apache trabalham no Windows e, portanto, podem ser utilizados como substitutos do Microsoft Office®, do Internet Explorer® e do IIS®, respectivamente. Além de ser menos radical, essa abordagem permite que a reação do usuário seja avaliada em uma pequena escala e os planos para treinamento do mesmo podem ser baseados na experiência real. Além disso, problemas como conversão de formatos de arquivos, macros e modelos podem ser facilitados se o aplicativo corrente permanecer por alguns momentos.

Esta abordagem permite dizer que a escolha do aplicativo no ambiente alvo final estará limitada aos que trabalham no aplicativo corrente. Por exemplo, o navegador web alvo pode ser o Galeon, porém o Mozilla é o único que funciona tanto no Windows como no GNU/Linux.

8.2. Faça primeiro as coisas fáceis

Em primeira instância, faça as mudanças que não dividem os usuários. Isto significa proceder às mudanças primeiro no servidor, o que vai fornecer uma plataforma para a introdução das mudanças do cliente, mais tarde. Muitas das mudanças do cliente serão compatíveis com o ambiente corrente, portanto, o efeito de divisão será minimizado.

Por exemplo, servidores de nomes DNS, servidores DHCP e servidores de base de dados com softwares proprietários, são todos candidatos à substituição por um software livre equivalente e ainda estabelecer interface com o resto dos sistemas correntes como antes.

O mesmo efeito pode ser verificado na adoção de ferramentas livres para autenticação de usuários e compartilhamentos de arquivos, cuja utilização podem ser bastante efetivos na divisão do ambiente em partes gerenciáveis.

8.3. Pense Além

Faça agora o possível para minimizar o que possa vir a dificultar a migração futura. Por exemplo:

1. Insista para que todo desenvolvimento feito para web, construído na instituição ou contratado, produza um conteúdo que possa ser visto pelos navegadores definidos pela e-PING, ou seja, sem utilização de padrões proprietários. Essa deve ser uma boa prática em qualquer caso, pois as Administrações não precisarão de um software específico para ver seu conteúdo. Existem ferramentas disponíveis para ajudar a verificar a compatibilidade das páginas da web.

2. Desencoraje o uso indiscriminado de macros e scripts em documentos e planilhas; encontre outros meios de prover funcionalidade. Novamente, esta deve ser uma boa prática, já que este uso indiscriminado é, comumente, um meio através do qual os vírus infectam os sistemas. As macros também podem ser facilmente usadas para roubar dados e subverter documentos, pois podem fazer o documento “dizer” coisas diferentes, dependendo de quem o está lendo, ou “dizer” outra coisa quando impresso. Finalmente, muitas das coisas feitas através de macros e scripts deveriam receber uma sistematização mais robusta.

3. Insista no uso de formatos de arquivos padrão abertos, por exemplo, Postscript e PDF. Sabe-se que há uma discussão sobre serem, o Postscript e o PDF, modelos abertos ou não. Isso é mais um debate sobre definições estritas e, em particular, de quem controla o modelo. Na realidade, esses são os únicos formatos de arquivos amplamente utilizados no momento, que tem definições disponíveis publicamente, e que podem ser usados sem restrições significativas. Diversas soluções em software livre produzem com facilidade esses formatos de arquivos.

Particularmente, não use formatos de arquivos proprietários para arquivos cujo objetivo é somente serem lidos e não editados pelo destinatário. Novamente, esta deve ser uma boa prática, pois tais arquivos são um meio comum de disseminar vírus. Ao usar tais formatos, a Administração estará amarrada ao vendedor por um tempo considerável. Além disso, esses formatos proprietários podem incluir quantidades consideráveis de metadados, em particular textos apagados previamente, os quais, caso vistos, podem tornar-se embaraçosos para a Administração. Não é difícil acessar essa metadados.

Guia Livre – Referência de Migração para Software Livre do Governo Federal

4. Ao escrever documentos em colaboração com outros, use um formato de maior aproximação possível (ou melhor acessibilidade). Isso aumentará a possibilidade de uso de aplicativos do software livre.
5. Use protocolos de padrão aberto. Definem-se protocolos de padrão aberto como aqueles que são livres de patentes e que tem implementação do software livre. Os padrões adotados pelo Governo Federal estão no e-PING.
6. Desenvolva sistemas baseados em um modelo, pelo menos, em três camadas (veja Seção 8.1), em que o código do aplicativo é independente da interface humana e dos métodos de acesso aos dados. Por exemplo, se possível, tenha uma interface que possa ser acessada em um navegador web de software livre. A construção de aplicativos desta forma, modular, tornará mais fácil a migração aos poucos. Isso reduzirá não somente a escala de qualquer fase de migração, como também o risco de fracasso. Os aplicativos de clientes monolíticos tradicionais são, notoriamente, difíceis de manipular.
7. Insista que todos os novos aplicativos sejam feitos para serem portáteis. Evite linguagens de arquitetura específica e APIs. Evite construir aplicativos que exijam a presença de outros aplicativos proprietários
8. Substitua os leitores de correios que usem formatos de armazenagem proprietários e/ou que se comuniquem com servidores usando protocolos não padronizados pela e-PING. Se possível, utilizar uma solução para armazenar listas de endereços e calendários em um formato aberto.

PARTE 3
Diretrizes
Técnicas

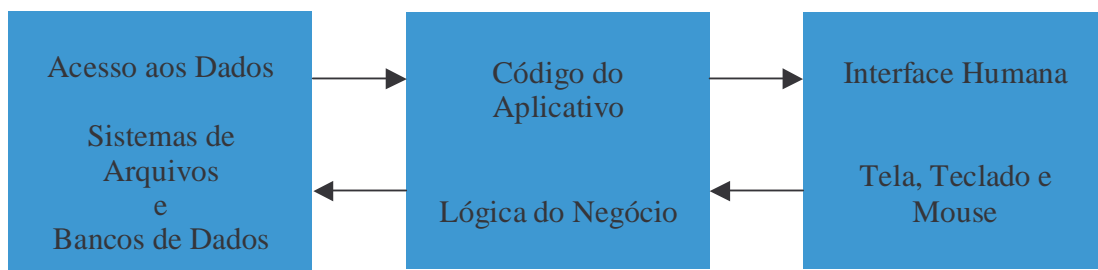
9. Arquitetura de Referência

9.1. Arquiteturas genéricas

Tem sido largamente usado o modelo de 3 camadas para se desenvolver softwares de computadores. Esse modelo separa os aplicativos em 3 partes:

- 1- Camada de acesso aos dados, sistemas de arquivos e bancos de dados.
- 2- Camada de código do aplicativo e lógica do negócio.
- 3- Camada da interface humana, tela, teclado e mouse.

Mais detalhes sobre este modelo podem ser encontradas em <http://www.corba.ch/e/3tier.html>.



Cada camada do aplicativo deve preocupar-se em realizar suas tarefas específicas deixando as outras tarefas e funções para as outras camadas do software. Isso traz o benefício de que o código do aplicativo pode ser mais simples e pode trabalhar mais facilmente em ambientes diferentes, porque sua dependência em relação ao acesso específico à máquina é reduzida. As setas indicam o fluxo da informação entre as diversas camadas do aplicativo. É interessante que estes fluxos de informação utilizem padrões abertos e bem definidos. Desta forma se torna mais fácil portar, ou migrar o aplicativo para diversas arquiteturas de hardware e software, e além disso é possível alcançar a interoperabilidade entre diversos sistemas.

Este modelo em 3 camadas foi generalizado para n-camadas, onde os componentes são ainda mais refinados, e é tipicamente desenvolvido utilizando-se a tecnologia de objetos ou de componentes.

Infelizmente muitos aplicativos Cliente/Servidor, no passado, só usaram um modelo de 2 camadas, onde o código do aplicativo e a interface humana se fundem. Isto significa que migrar tais aplicativos é mais trabalhoso que com os modelos 3/n-camadas, porque provavelmente a interface humana irá requerer alterações, pois o modelo de 2 camadas terá em sua grande maioria o código da interface humana inteiramente combinado com a lógica do negócio.

A comunicação entre as 3 partes de um modelo de três camadas, normalmente usa protocolos que permitem a cada camada trabalhar em uma máquina ou sistema diferente das outras duas camadas, caso isso seja desejado. Algumas vezes, as partes também são divididas entre outras máquinas. A escolha de como esta divisão é feita dá origem a várias arquiteturas genéricas.

No ponto de vista da estação de trabalho, onde no mínimo alguma parte do código da interface humana é realizada, são usualmente utilizadas as arquiteturas:

1- Cliente Leve (Thin Client)

Nesta arquitetura a estação de trabalho não precisa de dispositivos de armazenamento, tais como drive de CDROM, de disquete, disco rígido, etc. O cliente normalmente é um terminal burro, um computador antigo e/ou um dispositivo específico para Cliente Leve. As vantagens desta arquitetura são o baixo custo e a possibilidade de se reutilizar hardwares antigos e tecnologicamente obsoletos. Além disso, como um servidor remoto armazena todos os dados e executa todas as aplicações, existe maior facilidade de manutenção e atualização dos sistemas. Alguns exemplos desta arquitetura são: terminais gráficos, Terminal Services, WebService, terminais VT100, um dispositivo de navegador embutido, emuladores de terminal 3270, etc.

2- Cliente Pesado (Fat Client)

O código e os dados são mantidos na estação de trabalho sem conectividade de rede. As aplicações são executadas localmente, exigindo uma maior capacidade de processamento, memória RAM, armazenamento em disco rígido, e ainda, utilização de drive de disquete e CDROM. Vantagens: independência de outros computadores e servidores. Desvantagens: custo elevado, maior dificuldade em atualizar e dar manutenção ao sistema por ser descentralizado.

3 – Arquiteturas intermediárias

Existem diversas formas de se desenvolver, executar e utilizar aplicações intermediárias às arquiteturas Cliente Leve e Cliente Pesado, tais como:

Armazenar a aplicação em um servidor e depois transferir para a estação de trabalho para que ela seja executado quando for necessário. Por exemplo, esta é a forma como trabalham applets Java.

Uma outra variante é o acesso remoto a servidores de arquivos. Os dados ficam armazenados remotamente, mas são acessados e executados localmente na estação de trabalho. Como exemplo desta variante temos a tecnologia de compartilhamento de arquivos NFS.

4 – Escolha da arquitetura

A escolha da arquitetura para algum aplicativo ou até mesmo a estação de trabalho dependerá dos seguintes fatores:

a) **A largura de banda da rede para os servidores e o que essa largura de banda terá que carregar.** Se as estações de trabalho não forem “pesadas”, a rede terá que integrar os controles de interface humana, dados, ou código de aplicativo executado, além do tráfego normal da rede (TCP, UDP, etc). Em algumas circunstâncias, o volume de dados, gerados por uma única estação de trabalho ou um conjunto delas, pode ser excessivo para a capacidade da rede. Neste caso alternativas têm que ser avaliadas.

b) **A latência aceitável no uso do aplicativo.** Quando alguém interage com o aplicativo, ou estação de trabalho, apertando teclas ou movendo o mouse, o tempo que o

aplicativo leva para reagir e mostrar o resultado na tela, é conhecido como latência. Para alguns aplicativos pode-se tolerar uma latência maior, já para outros é necessário que os resultados sejam em tempo real. A latência vai depender da largura de banda disponível entre a interface humana e o aplicativo, entre o cliente e o servidor, e da capacidade da máquina de executar o código do aplicativo. Para atingir uma menor latência o aplicativo deve operar na mesma máquina da interface humana, e essa máquina deve ser poderosa o suficiente para executar o aplicativo.

c) **A política de segurança da Instituição.** Se os dados residem nas estações de trabalho distribuídas por toda a Instituição, isto significa que se qualquer máquina for roubada, ou estiver acessível em um ambiente inseguro, os dados podem ser perdidos ou divulgados a terceiros ou a pessoal desautorizado. Do outro lado se os dados estiverem centralizados em um servidor é preciso checar a segurança e privacidade da conexão dos clientes ao servidor. Pois se os dados estiverem trafegando de forma não criptografada e com falhas no controle ao acesso, é possível que alguma pessoa não autorizada veja seus dados enquanto eles trafegam pela rede. Neste momento é preciso rever as normas de segurança da instituição e analisar qual a melhor forma de adequar as soluções a essas normas.

d) **A Política de Backup da Instituição.** Se os dados residem nas estações de trabalho distribuídas por toda Instituição, será necessário algum mecanismo de backup centralizado ou a responsabilidade pelo backup deve ser distribuída entre muitas pessoas, provavelmente os próprios usuários. Um esquema centralizado de backup poderia vir a ser complexo e iria requerer uma alta largura de banda de rede e cooperação com os usuários das estações de trabalho (os quais, por exemplo, devem lembrar-se de não desligarem suas máquinas nos períodos em que estiverem marcados os backups). Já em um sistema de Cliente Leve como os dados estão centralizados em um servidor ou em um conjunto de servidores, o backup se torna mais fácil.

e) **O desenho do aplicativo.** Se for necessária interação humana com o aplicativo, ele precisará ter uma interface com o usuário que pode estar na estação de trabalho ou em um servidor. Por exemplo: um terminal IBM 3270 ou DEC VT100 tem todo o código de exibição processado no servidor. Por outro lado, um terminal baseado em um navegador, metaframe, Terminal Server, X-window system, terminal gráfico, VNC divide o código de exibição entre o servidor e o cliente.

f) **A capacidade da estação de trabalho de executar o aplicativo.** Na arquitetura de Cliente Pesado, quanto mais capacidade de processamento os aplicativos exigirem, mais poderosa (e, portanto, mais cara) a estação de trabalho deverá ser.

g) **A capacidade da estação de trabalho de armazenar dados.** Alguns aplicativos necessitam ter acesso a grandes banco de dados, os quais só podem ser sustentados por servidores especializados.

h) **O desempenho dos servidores disponíveis.** Se um aplicativo é executado remotamente em um servidor e não na estação de trabalho, o servidor deve ser suficientemente poderoso para executar todas as instâncias do aplicativo necessárias, no momento em que o número máximo de clientes estiver em uso.

i) **Custo total de implementação.** Da mesma forma que qualquer problema de engenharia, não há uma solução aplicável a todas as situações, e uma estação de trabalho particular pode operar de uma determinada forma para um aplicativo e de outra para um aplicativo diferente.

9.2. Arquitetura Básica de Referência

A Arquitetura Básica de Referência (ABR) usada nestas diretrizes é escolhida de forma a ser aplicável na maioria das situações. Ela pode se tornar mais “leve” ou mais “pesada” para aplicativos ou perfis específicos, caso necessário. Muitas vezes a arquitetura utilizada em uma Instituição tende a ser uma combinação de várias arquiteturas, cada uma escolhida para situações específicas.

A ABR é caracterizada como uma estação de trabalho sem condição específica, na qual:

- a) Todos os Aplicativos são executados sempre que possível na estação de trabalho e são armazenados nela mesma.
- b) Nenhum dado contínuo é guardado na estação de trabalho. Estes dados são armazenados em um servidor central.
- c) Toda a autenticação e autorização são controladas por servidores centrais.
- d) A gestão do sistema é centralizada.
- e) O objetivo é que as estações de trabalho fiquem operacionais e que não necessitem de suporte local.

Os aplicativos funcionam localmente para reduzir quaisquer problemas de latência, e a ABR pressupõe a existência de banda larga suficiente para que os dados sejam mantidos centralmente. Além disso, ela pressupõe que todas as estações de trabalho sejam essencialmente idênticas, permitindo que qualquer pessoa conecte-se a qualquer máquina que lhe seja permitido usar. Deve haver um regime de gestão de sistemas eficiente, para manter as instalações dos softwares das estações de trabalhos em harmonia.

A ABR concentra todos os dados importantes da Organização nos servidores centrais, visando facilitar a gestão da informação e os processos de backup, e disponibiliza estações de trabalho individuais, reduzindo o impacto na eventual avaria de uma máquina cliente.

A manutenção dos dados de forma local significa que há uma identificação da máquina com o usuário. Isto causa problemas quando o usuário muda de lugar ou deixa a organização. Também transforma o lugar da estação de trabalho em local específico de um usuário, dificultando a implementação da sistemática de mobilidade, onde o usuário tem suas personalizações em qualquer estação que vier a utilizar (conceito *hot-desking*).

A manutenção dos dados de forma central elimina essas dificuldades e flexibiliza a utilização da estação de trabalho. Também permite que se mantenha o menor tamanho possível de armazenamento local na estação de trabalho. Com isso, a estação transforma-se em um dispositivo do tipo “ligar e funcionar”, facilitando a instalação. Dessa forma, a ABR é considerada uma boa escolha para diversas situações.

A ABR não é recomendada para laptops, ou para estações de trabalho que não sejam permanentemente conectados à rede da Instituição. Tais dispositivos necessitariam de mais recursos para poder armazenar todos os dados localmente.

10. Grupos Funcionais

Para efeito de organização, os sistemas foram separados em três grandes grupos:

- Sistema Operacional – uma vez que a maioria das considerações acerca do ambiente operacional é pertinente tanto a estações de trabalho como a servidores, elas serão tratadas a parte;
- Estação de trabalho – considerações quando a aplicações voltadas para utilização em estações de trabalho.
- Servidores – abordagem dos módulos de servidores de diversos tipos.

Por sua vez, o segundo e o terceiros itens são organizados dentro de grupos funcionais, definindo os tipos característicos de atividade genérica de computador em uma Administração. Isto significa que atividades específicas, como Gestão de Projetos ou Sistemas de Informação Geográfica não são consideradas nesta versão do documento. As atividades não consideradas devem ser, em geral, as usadas por uma pequena proporção da população de usuários.

A grande extensão de software livre disponível significa que, para muitas funções, há vários aplicativos diferentes disponíveis. A escolha do aplicativo a ser usado não é sempre evidente e deve ser pautada por diretrizes e especificações técnicas.

O modelo de referência usado nestas diretrizes deve, portanto, ser tratado como exemplo de um sistema que, reconhecidamente, funciona, mais do que como uma recomendação de sistema para ser usado em todas as circunstâncias.

As diretrizes discutem os assuntos que os que tomam as decisões devem levar em conta e pode acontecer dessas pessoas chegarem a conclusões diferentes, mas igualmente válidas. Em qualquer caso, restrições locais ao Administrador podem tornar necessária a escolha de um modelo diferente. As escolhas possíveis para cada grupo são discutidas em detalhes no capítulo 11, para os grupos Principais e no capítulo 12 para os grupos Auxiliares.

Há alguns sítios de referência úteis, que contém listas de aplicativos de software livre, mostrando o que há disponível, e quais deles são candidatos à substituição de quais aplicativos proprietários. O sítio <http://linuxshop.ru/linuxbegin/win-lin-soft-en/> é um exemplo.

O sítio <http://www.osafoundation.org/desktop-linux-overview.pdf>, que contém detalhes de muitos dos aplicativos discutidos abaixo e é, portanto, um outro local onde os Administradores podem obter mais informações.

Uma dos pontos fortes do software livre é o fato de ser modular e poder ser montado de várias formas diferentes, permitindo que os sistemas sejam talhados para satisfazer a necessidades específicas. Essa modularidade é possível porque o software livre se adapta a interfaces abertas e disponíveis publicamente.

Infelizmente, essa flexibilidade pode, às vezes, traduzir-se como uma dificuldade, já que os Administradores podem assustar-se com a grande quantidade de opções disponíveis. Há muitas organizações que podem prover ajuda e suporte, da mesma forma como existe no mercado proprietário.

10.1 – Sistema Operacional

Há vários sistemas operacionais do software livre, e várias distribuições (explicado adiante) do GNU/Linux, entretanto, muitas pessoas só tenham ouvido falar do GNU/Linux, e geralmente pelo nome Linux.

Um sistema operacional consiste de um *kernel* que opera em modo supervisor, junto com programas de suporte que operam sob o controle do *kernel* no modo usuário. O Linux é um *kernel*, mas demanda carregadores de suporte, compiladores, *drivers* etc. A maior parte desses programas de suporte são fornecidos pelo projeto GNU da *Free Software Foundation* e, portanto, todos devem ser chamados de GNU/Linux, que é o termo usado neste relatório.

O *kernel* do Linux é fornecido em pacote junto com um conjunto de programas de suporte e aplicativos, por algumas companhias como Conectiva, Red Hat, SuSE e Mandrake, como uma Distribuição. Os componentes do conteúdo de uma Distribuição devem interagir e o *kernel* pode ser compilado para possuir características específicas não disponíveis nas Distribuições. Portanto, a escolha da Distribuição deve ser considerada, já que cada um tem seus pontos positivos e negativos.

Há outras Distribuições como o Kurumin, Debian, Slackware e o Gentoo, que não são preparados por uma organização comercial e isto tem implicações na forma como o suporte é fornecido. O suporte para essas distribuições vem de terceiros ou de acessos a listas de correio na Internet. Ambos podem oferecer níveis aceitáveis de cobertura.

O Debian tem reputação de solidez e sua seção estável contém um código que foi cuidadosamente testado por muitas pessoas no mundo todo. Há mais duas seções que fornecem níveis crescentes de *software* de ponta. A seção estável também é reputada como desatualizada, o que, até certo ponto é injusto porque a maioria dos usuários comerciais está interessada, principalmente, em estabilidade e ausência de defeitos, entretanto, por estar desatualizado geralmente não possui suporte para os últimos periféricos.

O Gentoo é uma distribuição única do código fonte, o que significa que a Administração pode construir seus próprios binários facilmente, talhando a distribuição ao seu ambiente e seu *hardware*. Construir uma distribuição do zero demanda tempo, porém, construídos os binários, eles estarão disponíveis de um modo geral. Este é uma nova distribuição com características próprias, e vale a pena considerar. Pelo fato de que a maior parte das outras distribuições são fornecidos com código de fonte completo, é possível talhar qualquer um deles da mesma forma, porém o Gentoo talvez seja mais receptivo a tal tratamento.

As distribuições comerciais vêm em pacotes diferentes, com diferentes níveis de suporte disponíveis. A distribuição disponível via Internet tem suporte por algo em torno de um ano, invariavelmente, e então, os usuários têm que fazer um *upgrade*. A maior parte das companhias oferecem uma versão Enterprise que tem garantia de suporte por cinco anos ou mais e que é baseada em versões estáveis. Tais versões também possuem um contrato de suporte associado a elas, que às vezes, é chamado de licença, embora o código seja licenciado usando o GPL ou LGPL e não pode ser licenciado de outra forma. O que muitos Administradores desejam é a disponibilidade de tais distribuições estáveis e que as mesmas possuam suporte. Na verdade, uma razão para mudar para software livre é a inexistência de pressão para *upgrade* de forma constante e desnecessária. As companhias prometem um conserto de defeitos de *backport*.

Nossa opinião é de que GNU/Linux é a plataforma preferida para estações de trabalho, já que oferece ferramentas de configuração melhores e uma variedade de pacotes de configurações que são mais apropriados para uso em estações de trabalho.

Além disso, alguns produtos de estações de trabalho populares ainda não funcionam em qualquer sistema. (por exemplo, o navegador Mozilla atualmente não trabalha em OpenBSD).

Para servidores, a situação é bem menos clara. O OpenBSD tem, de longe, o melhor registro de segurança de qualquer sistema operacional do software livre. Deve ser a plataforma preferida para qualquer situação que requeira mais do que a média de segurança (tal como *firewalls* e servidores de Zona Desmilitarizada).

Aplicativos de servidores geralmente funcionam bem em todas as plataformas BSD e GNU/Linux, embora muitos tenham sido escritos para GNU/Linux e depois adaptados. O *software* proprietário freqüentemente só é disponível para GNU/Linux./Linux.

10.2. Estação de Trabalho

O sistema operacional é o GNU/Linux com interface do usuário baseada no KDE, mas deve-se considerar a interface Gnome e o Ximian, baseado no Gnome.

Sugerimos que os sistemas de arquivos contendo binários (tais como “/usr”) podem ser montados em formato “*somente para leitura*”, para evitar que os usuários alterem seu conteúdo, e os remanescentes estão montados em formato “*não executável*”, para evitar que o código seja executado a partir deles. Para reforçar, a interface do usuário só deve permitir aos usuários executarem programas através de interfaces pré-definidas. Isto significa que o acesso à linha de comando ou a habilidade de criar ou mudar itens de menu ou ícones, deve ser removida. Os sistemas de arquivos que contém dados voláteis baseados no usuário são montados a partir de um servidor NFS central. A autenticação do usuário é feita junto a um serviço de diretório LDAP.

Sugere-se ainda que servidores centrais podem fornecer endereço IP e configuração de rede através de DHCP e resolução de nome por um servidor DNS no momento em que o computador é ligado e começa a carregar o sistema operacional.

10.2.1 Gerenciadores de Janelas

Há várias opções, que vão dos gestores leves de janelas muito simples, como o *icewm*, até gestores de sessão completos como os incluídos no Gnome e KDE. A escolha depende do uso pretendido.

Dos gerenciadores de janelas, o KDE é o mais maduro mas o Gnome está aproximando-se rapidamente. O Gnome tem o suporte da Sun Microsystems e de membros da *Gnome Foundation*.

A Ximian lançou uma estação de trabalho baseado no Gnome chamado XD2. Ele trabalha sobre várias distribuições de base diferentes, inclusive o Red Hat e SuSE. A Ximian dedicou-se especialmente a integrar os vários diferentes aplicativos, para certificar-se de que trabalham de forma similar. Isso significa que eles incluíram suas próprias versões de alguns produtos como OpenOffice.org.

A escolha de cada Administração será provavelmente de preferência pessoal; ambos os ambientes são muito capazes. Aplicativos desenhados para trabalhar em um ambiente normalmente trabalharão em outro, porém recursos mais específicos como gerenciadores de janelas, podem não trabalhar de forma adequada.

10.2.2 Escritório

A criação, modificação e impressão de arquivos contendo dados do negócio em formato padrão, tais como cartas e relatórios. Também a criação, modificação e impressão de planilhas e apresentações. É preciso que haja utilitários para gerenciar esses arquivos. Deve ser possível ler e escrever formatos proprietários da melhor maneira possível, quando isto se tornar necessário. Quanto aos formatos abertos, obviamente, estes devem ser passíveis de leitura e escrita sem nenhuma dificuldade. Deve ser disponibilizado o idioma português brasileiro e, preferencialmente, outros possíveis idiomas a serem utilizados, como espanhol e inglês, além de configurações de moedas e alfabetos.

10.2.2.1 Escritório - Opções

Um pacote de escritório muito utilizado na atualidade, principalmente na administração pública, é o Microsoft Office. Ele inclui os aplicativos proprietários de editor de texto, planilha, apresentação e correio eletrônico, junto com seus formatos também proprietários. Esses formatos não são abertos e mudam de uma versão do MS-Office para outra. Até mesmo os próprios produtos da Microsoft não podem garantir a capacidade de ler e escrever um arquivo com 100% de precisão, a não ser que os arquivos tenham sido criados na mesma versão do produto.

Os aplicativos software livre atualmente são capazes de ler formatos fechados com tal precisão que os problemas encontrados não são diferentes dos vistos com o uso das diferentes versões dos próprios produtos proprietários. Quanto mais antigo o formato, melhor os aplicativos do software livre lidam com o mesmo. Os aplicativos software livre tendem a ser melhores na leitura de arquivos em formato proprietários, do que na escrita de arquivos do mesmo formato. Em termos gerais, os aplicativos software livre podem ser usados com confiança.

A exceção é quando é requerido algum tipo de trabalho colaborativo e, no mínimo, uma das partes insista em usar um formato proprietário. A leitura, a alteração e a reescrita dos arquivos nesses formatos podem introduzir anomalias que o uso de um aplicativo proprietário único não faria. No entanto, deve-se ter em mente que esse tipo de erro também pode ocorrer caso sejam usadas diferentes versões do *software* proprietário. Sugere-se nestes casos o uso do formato RTF (Rich Text Format) ou mesmo texto puro, até que formatações sejam requeridas. Para arquivos que só permitem leitura, sem atualização, o formato *PDF* pode ser usado.

Pode-se julgar também que a interface do usuário deve ser tão similar quanto possível ao *software proprietário*, para minimizar custos de re-treinamento. Alguns scripts de configuração fazem isto de forma automática, por exemplo, no OpenOffice.org, podendo ser utilizadas para instalação em larga escala.

Modelos e Macros *Visual Basic (VB)* são comuns em muitas Administrações. Elas têm um formato proprietário similar e terão que ser reescritas.

Há três conjuntos de escritório principais em software livre: OpenOffice.org, KOffice e Gnome Office, e todos devem ser considerados.

Está disponível na Internet um estudo piloto que compara a capacidade dos vários conjuntos de escritório em software livre de lidar com os arquivos proprietários:

<http://www.acmqueue.com/modules.php?name=Content&pa=showpage&pid=55>

a. OpenOffice.org

O OpenOffice.org é um conjunto de aplicações para escritórios do software livre baseado no StarOffice, produzido por uma empresa alemã chamada StarDivision. A Sun Microsystems, comprou a StarDivision e franqueou o código à comunidade de software livre. Continua a colocar no mercado uma versão do OpenOffice.org, ainda chamada de

Guia Livre – Referência de Migração para Software Livre do Governo Federal

StarOffice, que vende a um preço muito mais barato do que os pacotes proprietários correspondentes.

O StarOffice e o OpenOffice.org são essencialmente idênticos, a não ser por:

- A Sun Microsystems fornece suporte comercial para o StarOffice.
- O StarOffice possui um sistema de banco de dados construído internamente (Adabas).
- O StarOffice tem alguns filtros extras para migração para/de outros pacotes de *office* (no entanto, o filtro do Wordperfect não é disponível em GNU/Linux por razões de licença).
- O StarOffice possui algumas fontes proprietárias.
- O StarOffice só é disponível em um conjunto de idiomas um pouco mais restrito (Alemão, Francês, Italiano, Inglês, Espanhol e Sueco).
- O OpenOffice.org é atualizado mais frequentemente do que o StarOffice.

Ambos os aplicativos são comparáveis ao Office, porém não incluem um cliente correio eletrônico. Ambos lidam com a maior parte dos arquivos Office até e inclusive os do Office XP, embora a compatibilidade piore progressivamente com as versões do Office posteriores ao Office 97. Não lidam com arquivos protegidos por senhas (exceto para proteção de planilhas página a página) e tem alguns problemas com objetos gráficos com *link* com OLE). Ainda assim, têm algumas das melhores integrações de formatos de arquivos de outras ferramentas de Escritório.

Se a Administração está migrando de um ambiente proprietário, o OpenOffice.org possui uma versão para esse sistema operacional, proporcionando aos usuários um contato inicial com o novo *software* em um ambiente familiar.

Eles têm sua própria implementação do Basic e não lidam com macros do Visual Basic, que precisam ser traduzidas à mão. Embora a tradução seja um custo de migração, a falta de suporte macro impede que sejam transmitidos vírus de macro.

A *Sun Microsystems* está estabelecendo ligações com companhias, para traduzir macros e modelos proprietários para uma forma compatível com o StarOffice.

Eles também oferecem uma interface Java, porém só reconhecem atualmente o JDK da Sun Microsystems. A Sun Microsystems anunciou um projeto para desenvolver um tradutor de *Visual Basic for Applications* (VBA) para Java.

A versão disponível para cópia do OpenOffice.org.br contém o dicionário português brasileiro, porém dicionários estão disponíveis para diversos idiomas. Já existem versões pré-construídas em mais de 25 línguas diferentes.

Embora o OpenOffice.org não ofereça atualmente um pacote de banco de dados, ele tem interfaces ODBC e JDBC com muitos sistemas comuns de banco de dados, inclusive com os populares do software livre. Além disso, não há filtros de conversão para Wordperfect, porém o lançamento futuro está planejado. Ambos trabalham em uma faixa de sistemas operacionais que incluem ambiente proprietário e GNU/Linux.

Revisores estão começando a avaliar o OpenOffice.org quanto à funcionalidade e estabilidade. Veja:

<http://www.raycomm.com/techwhirl/magazine/technical/openofficewriter.html>

Guia Livre – Referência de Migração para Software Livre do Governo Federal

O OpenOffice.org possui uma versão em português brasileiro sendo constantemente desenvolvida, disponível para cópia no endereço <http://www.openoffice.org.br>.

Neste sítio do OpenOffice.org.br – Projeto Brasil, há uma série de documentações sobre instalação, uso e funcionalidades avançadas do produto. Veja mais no endereço supracitado, dentro da área de Ajuda – Documentação.

b. Koffice

Este é o componente *office* da estação de trabalho KDE. É um pacote integrado que oferece processador de textos, planilhas, mapas e gráficos, apresentações, ilustrações, geração de relatórios e ferramentas para fluxogramas, com um *desktop* opcional chamado Workspace.

Os filtros de arquivos proprietários não são tão bons quanto os oferecidos pelo OpenOffice.org. Ele não tem uma linguagem macro, mas há *scripting* disponível.

O Koffice trabalha bem uma boa interface intuitiva.

c. Gnome Office

É uma coleção de programas escritos nos padrões Gnome e podem, portanto, integrar-se uns aos outros, tem uma interface com o usuário similar e devem ser capazes de embutirem-se uns nos outros.

O OpenOffice.org é considerado agora parte do Gnome Office, muito embora não se prenda aos padrões do Gnome. A Ximian, particularmente, está trabalhando para tornar o OpenOffice.org mais compatível com o Gnome e incluiu sua própria versão no mais recente produto para estações de trabalho, XD2. Veja <http://www.gnome.org/gnome-office/> para maiores detalhes.

O Gnome Office tem muitos componentes, inclusive o AbiWord (processador de texto), o Gnumeric (planilhas), Sodipodi e Sketch (desenho de gráficos vetoriais), o Gimp (edição de imagens), o Olho do Gnomo (exibição de imagem), o Dia (gráficos de vetores, similar ao Visio) e o Agnubis (gráficos para apresentações).

Estes componentes apresentam graus variados de utilização, por exemplo, o Abiword é muito bom para processamento básico de texto, porém tem problemas com tabelas, e o Agnubis é muito limitado, enquanto o Gnumeric é uma planilha muito boa. O trabalho de desenvolvimento em quase todos os componentes ainda é muito ativo.

O Gnumeric, em particular, tem o objetivo de desenvolver uma planilha que pode fazer tudo que o Excel faz, e ainda bastante mais. O último lançamento de teste beta do Gnumeric (1.1.20 quando foi escrito) suporta todas as funções de trabalho da versão americana do Excel, e é esperado, para setembro 2003, um comunicado incluindo este nível de funcionalidade. Os desenvolvedores vêm de uma formação financeira e incluíram várias características que tornam o Gnumeric especialmente útil para aplicações financeiras. É nessa área que eles acreditam que o Gnumeric excede o Excel.

O Gnumeric usa o “XLS” como formato de arquivo nativo, enquanto o OpenOffice.org converte planilhas em um formato baseado em XML.

A gama de produtos disponíveis é interessante e, junto com o OpenOffice.org, oferecem um bom número de soluções diferentes. No entanto, caso seja necessário um conjunto integrado, o atual OpenOffice.org é a única solução real.

10.2.3 Correios

A criação, o recebimento e a apresentação de correio eletrônico, incluindo suporte para correio seguro.

10.2.3.1 Cliente de e-mail - Opções

Há um grande número de clientes de e-mail (MUA) baseados em texto e com interface gráfica disponíveis no campo do software livre.

Para os que estão acostumados a usar o Outlook ou Outlook Express e desejam ter algo similar, a escolha óbvia é o Evolution. O Evolution não é somente um cliente de correio, mas também um Gestor de Informação Pessoal (*Personal Information Manager - PIM*). Ele possui integração LDAP e pode, portanto, acessar dados de nomes e endereços da Administração, desde que permaneça no próprio projeto do Evolution. Está sendo desenvolvido ativamente pela Ximian como seu “produto âncora”. A Ximian tem um produto chamado Connector que permite ao Evolution conectar-se ao Exchange (mas não na versão 5.5). O Connector era proprietário mas teve seu código aberto e disponibilizado sob a licença GPL recentemente.

O *Evolution* tem uma interface com o usuário muito similar ao *Outlook*, tornando-se fácil para as pessoas aprenderem. Também possui alguns aspectos úteis como as Pastas Virtuais.

O Kmail e o Sylpheed são alternativas aos clientes de e-mail. Ambos são muito bons e integram-se com os principais ambientes de estações de trabalho do software livre. Se a estação de trabalho for KDE, usa-se o Kmail, enquanto que se a estação de trabalho for Gnome, usa-se o sylpheed.

O Evolution dá suporte ao GPG, porém não ao S/MIME, embora se espere que o faça em breve.

O Mozilla dá suporte ao S/MIME, mas não ao GPG ou PGP, apesar de que o fará em breve.

O Kmail dá suporte ao S/MIME, ao GPG e PGP, e foi desenvolvido recentemente para funcionar como um cliente junto ao servidor *groupware Kontact*.

Muitos pacotes de *groupware* também incluem clientes compatíveis com IMAP e POP3. Em geral, eles não são tão bons quanto o Evolution, mas podem ser suficientes, se integrarem-se bem com as outras funções do *groupware*.

Em alguns casos, pode ser melhor migrar algumas categorias de usuários de correio a uma interface de usuário baseada na *web*. O SquirrelMail é uma particularmente boa, que pode ser encontrada em <http://www.squirrelmail.org/>.

A *Open Systems Application Foundation* tem um produto chamado Chandler, que está em fase inicial, mas vale a pena monitorar para o futuro. É um competidor potencial do Evolution. Já que a maioria dos usuários irá, provavelmente, requerer algo parecido com o Outlook da *Microsoft*, é provável que o Evolution seja a escolha preferida, e é a escolha de referência. No entanto, se o S/MIME for requerido imediatamente, então o Kmail terá que ser usado, o que pode significar o uso do KDE ao invés do Gnome.

10.2.4. Calendários e *Groupware*

A criação e gestão de calendários pessoais e de grupos e catálogos de endereços. Os calendários também devem permitir a organização de reuniões e a reserva de salas. Os catálogos de endereços devem integrar-se com os outros grupos funcionais, como o correio eletrônico.

O Evolution é um para calendário pessoal e gestão de contatos. No momento parece difícil encontrar um *groupware* com software livre, somente soluções baseadas na *web* estão realmente disponíveis, embora recentemente o projeto *Kontact* tenha produzido uma solução usando o *Kmail* como cliente. Portanto, para uma verdadeira solução de software livre, teria que ser utilizado um *navegador* para acessar o *groupware*.

10.2.5. Navegador

A habilidade de acessar os protocolos de acesso à *Internet* e apresentar os resultados. Isto é normalmente feito com um *navegador*. Além disso, a habilidade para criar conteúdo e disponibilizá-lo interna e externamente.

10.2.5.1 Opções de Navegador

Os principais *navegadores* do software livre são Mozilla, Galeon e Konqueror. Há outros como Lynx, que é somente texto, e é frequentemente usado como base para *navegadores* para pessoas com deficiência física, e o Mozilla Firefox (anteriormente conhecido como Phoenix e depois como Firebird), um variante leve do Mozilla.

O Mozilla é o principal projeto software livre baseado em código, distribuído pela Netscape, e é a base para a Netscape 7. Ele contém correio e componentes de notícias junto com catálogo de endereços e uma ferramenta de autoria de páginas de Internet. Grande parte do código do Mozilla é usado por outros projetos, inclusive o Galeon e o OpenOffice.org.

O *Mozilla* é uma alternativa no caso de se requerer um produto que inclua os recursos de leitor de correios e catálogo de endereços. O *Mozilla* também seria a opção no caso da Administração estar usando no momento estações de trabalho com ambiente proprietário, então seria necessário que o *navegador* trabalhasse no ambiente existente, para permitir aos usuários um contato inicial com o novo *software*, em um ambiente familiar.

O Galeon é somente um *navegador* e é desenhado para ser pequeno e rápido. É baseado no Gecko, o motor de renderização no qual se baseia o projeto do Mozilla, junto com uma interface de usuário Gnome. Ambos, Galeon e Mozilla dão suporte a todos os padrões de Internet aberta e podem executar Java e Javascript escritos apropriadamente. O *Galeon* é um navegador rápido, de uma única função, que possui uma boa interface com o usuário

Alguns conteúdos requerem um *plugin* que só é disponível para Ambiente proprietário, como Shockwave Director. O produto proprietário CodeWeavers *CrossOver Plugin* permite que *plugins* que trabalham em Ambiente proprietário, trabalhem no GNU/Linux.

O Konqueror é o navegador escrito para estações de trabalho KDE e também é usado como um gerenciador de arquivos “arraste e solte”. Baseia-se no motor de renderização KHTML, com Mozilla Gecko como opção, junto com uma interface de usuário KDE.

10.2.6. Banco de Dados

A gerência de informações estruturadas em banco de dados pessoais.

10.2.6.1 Bancos de Dados Pessoais ou Locais

Bancos de dados pessoais que são utilizados de forma local, ou seja, na própria estação de trabalho; tem suporte através de alguns produtos que também já são considerados *SGBD's* centrais e completos.

Para que tenham funcionalidades similares a um banco de dados *ad hoc*, utilizam-se ferramentas externas de acesso a estes mecanismos. Tais ferramentas disponibilizam funcionalidades que incluem: manutenção (criar, alterar e remover) de tabelas, de índices, consultas específicas (utilizando o SQL como linguagem de consulta), relatórios para impressão ou visualização, e a possibilidade de se construir formulários para a automação do processamento dos dados armazenados no banco de dados, como uma tela amigável para inclusão de dados ou uma grade (ou matriz) dos dados existentes para uma navegação interativa que facilite a escolha de registros para alteração/exclusão. Como tudo em UNIX, as soluções são construídas em blocos, que sabem como fazer bem o seu trabalho, ao contrário das soluções fechadas proprietárias que tem um grande número de funcionalidades, aumentando o escopo funcional do software ou solução.

Bancos de dados pessoais são baseados no *MySQL* ou em um produto de *groupware* baseado na *web*, como *phpGroupware*.

10.2.6.2. Conectividade de Bancos de Dados

A maior parte dos produtos *SGBDs* dão suporte direto a APIs com ligações de linguagem C. Alguns também dão suporte naturalmente a C++. Todos oferecem conectividade ODBC ou JDBC. Alguns também oferecem conectividade .NET.

Há um produto chamado Unix-ODBC que provê conectividade do tipo ODBC aos programas Unix e GNU/Linux.

10.3. Os Servidores

O sistema operacional é o *GNU/Linux*. Para máquinas altamente seguras como *firewalls*, onde o OpenBSD fosse usado juntamente com o GNU/Linux, a escolha seria outra.

As principais funções dos servidores são fornecidas por:

10.3.1. Serviço de Correio

O Correio é uma área complexa, com muitos componentes lógicos, e tem riqueza de aplicativos software livre, alguns dos quais sobrepõem funcionalidades. Também é estreitamente ligado a outras questões, incluindo controle de vírus e de *spam*.

A escolha dos aplicativos apropriados é complexa, e está incluída no Apêndice C uma discussão detalhada sobre essas questões, junto com uma definição de todos os termos usados aqui.

10.3.1.1 Opções de MTA

Os principais MTAs do software livre são Sendmail, Exim, Courier-MTA e Postfix. Há muitos outros, mas estes são considerados os principais, por serem usados em larga escala.

Tradicionalmente, os *sítios* do Unix e do software livre usam o Sendmail como seu MTA. Infelizmente, ele apresentou um registro de segurança pobre, e também é notoriamente difícil de configurar.

Todos os outros têm boa reputação em nível técnico e há pouco a escolher entre eles. No entanto, há uma diferença significativa no padrão da documentação disponível, com o Postfix apresentando uma melhor documentação em nível inicial e o Exim em nível especializado.

O Courier-MTA faz parte de uma família com um MTA, MDA (Mail Delivery Agent), MAA e pacote *webmail* (*squwebmail*) disponível. Cada parte pode ser usada por si só, ou integrada com o resto da família.

Uma escolha de referência é o Exim, porque é tão capaz quanto o Sendmail, e ao mesmo tempo, mais fácil de configurar, e provavelmente mais seguro. Ele também entende as opções do Sendmail e pode, portanto, funcionar como substituto compatível com o Sendmail. Os outros dois são, talvez, menos capazes de lidar com grande volume de mensagens. A escolha não é evidente e os Administradores devem realmente tomar suas próprias decisões com base nas necessidades locais.

10.3.1.2 Opções de MAA - Depósito de Correio

Muitos Administradores preferem que os clientes usem o armazenamento central de correio, em vez de descarregar as mensagens para armazenamento local na estação de trabalho do cliente. Por essa razão, nós recomendamos fortemente o uso do IMAP.

Há três servidores IMAP do software livre bem conhecidos: UW-IMAP (às vezes chamados somente de IMAP), Courier-IMAP e Cyrus.

O UW-IMAP tem uma história de segurança pobre e não o recomendamos.

Dos outros dois, o Courier-IMAP é largamente conhecido por ser o mais fácil de configurar. Ele tem um tamanho pequeno e trabalha bem com o Postfix e com o Courier-MTA. É a parte MAA da família Courier. Ele necessita do *maildir* como formato de armazenamento de correio.

O Cyrus usa seu próprio formato de armazenamento de correio, que é similar ao *maildir* e necessita de seu próprio MDA para completar o armazenamento de correio.

Guia Livre – Referência de Migração para Software Livre do Governo Federal

Ambos, Courier-IMAP e Cyrus suportam o **TLS** (um protocolo padrão de autenticação e privacidade).

Há vários MDAs, por exemplo, o procmail, *maildrop* do Courier, que vem como parte da família Courier e fornecimento da Cyrus. Os MDAs também possuem a habilidade de filtrar o correio de acordo com regras sofisticadas, o que é útil se o MUA usado não tiver dispositivos de filtro.

Uma escolha de referência é o Courier-IMAP sem MDA. Não é preciso um MDA, porque o Exim é capaz de escrever diretamente para as estruturas *maildir* e o Evolution tem os seus próprios filtros com muita capacidade.

10.3.1.3. Outras Ferramentas de Correio

Existe um grande número de filtros de e-mail livres, ferramentas Anti-Spam, que evitam que anexos executáveis sejam baixados juntamente com os e-mails, e diversos outros tipos de opções. Atualmente existem 342 projetos desse tipo cadastrados no sourceforge.net, Mais informações em: http://sourceforge.net/softwaremap/trove_list.php?form_cat=29

O **SpamAssassin** é provavelmente o mais utilizado de todos os filtros anti-spam. Seu princípio de funcionamento buscar por assinaturas heurísticas típicas de SPAMS no cabeçalho e no corpo das mensagens e atribuir a cada assinatura encontrada uma pontuação, após isso a pontuação é checada, se exceder um limite do que é considerado normal, a mensagem é declarada como SPAM. Ele pode rodar diretamente no MTA ou no MUA. O sítio oficial é www.spamassassin.org nele podem ser encontradas diversas informações, como configurá-lo e utilizar em diversos MTA's diferentes e MUA's.

O **Anomy Sanitizer** é um conjunto de filtros que permitem se procurar por vírus nas mensagens de e-mail, desabilitar códigos HTML e Javascript potencialmente perigosos, bloquear ou “mangle” anexos baseado em seus nomes de arquivo. Desta forma você não precisa receber, por exemplo, visual basic scripts, então você não precisa se preocupar com o risco que receber este tipo de arquivo implica. (O vírus ILOVEYOU era um programa visual basic). Mais informações em <http://mailtools.anomy.net>

MailScanner é um e-mail vírus scanner, protetor de vulnerabilidades, e marcador de spam. Ele utiliza o SpamAssassin para uma detecção muito boa dos spams, e é desenhado para tratar com ataques do tipo Denial Of Service. Ele vai detectar arquivos zip protegidos por senhas e aplicar uma checagem de nome de arquivos aos seus conteúdos. É muito fácil de instalar, e suporta em uma grande quantidade de servidores de e-mail, Postfix, Sendmail, Exim, Qmail, Zmailer e diversos antivírus: Sophos, MacAfee, F-Prot, F-Secure, DrWeb, ClamAV, BitDefender, RAV, Panda, e muitos outros antivírus. Ele pode ser integrado em qual sistema de e-mail. Mais informações: <http://www.mailscanner.info/>

Fetchmail é um completo, bem documentado, robusto aplicativo para fazer cópia de emails remotos e realizar encaminhamento entre servidores. Ele suporta quase todos os protocolos de acesso remoto a e-mail em uso atualmente na Internet, tais como: POP2, POP3, RPOP, APOP, KPOP, todos os tipos de IMAP, ETRN, e ODMR. Ele também IPv6 e IPSEC. Suporta diversos tipos de autenticação tais como NTLM, IMAP RFC1731 encrypted method. Como ele puxa o correio para uma máquina (a transferência é iniciada pela máquina receptora), ele é útil onde, por razões de segurança, os administradores não querem abrir uma porta para a internet em sua máquina, para permitir que o correio lhes seja imposto (onde o remetente inicia a transferência), como aconteceria com o modelo SMTP normal. Mais informações podem ser encontradas em: <http://www.catb.org/~esr/fetchmail/>.

Estes são alguns das dezenas de aplicativos filtros de e-mail LIVRES. Existem atualmente 2174 projetos de softwares livres relacionados a e-mail, desde clientes, a servidores, aplicativos web, filtros anti-spam, webmail, antivírus, etc cadastrados no sourceforge.net, todos estes projetos podem ser visualizados em

http://sourceforge.net/softwaremap/trove_list.php?form_cat=28

10.3.1.4. Problemas Encontrados

Para armazenar dados em um servidor LDAP é preciso escolher um esquema. O esquema deve ser compatível com todos os clientes que possam requerer acesso aos dados. Felizmente, alguns pacotes vêm com um esquema que, não somente dá suporte às suas necessidades, como também às necessidades de vários outros pacotes em geral.

O Courier-IMAP vem com um esquema, mas o Exim não. Descobriu-se que o esquema do Courier dá suporte ao Exim também. É preciso verificar se este suporte abrange toda a capacidade do Exim. Foram descobertos alguns problemas com o arquivo de configuração do LDAP para o Courier. Situações críticas foram retornadas ao desenvolvedor mas ainda não foram liberadas.

Para usar o Courier com o Whoson são necessárias algumas correções no Courier. Algumas delas foram disponibilizadas no sítio do Whoson, porém estavam velhas e necessitavam de atualizações significativas para trabalhar com a versão selecionada do Courier.

10.3.2. Serviço de Antivírus

Se os sistemas de software livre estiverem configurados corretamente, os vírus terão efeito limitado. No entanto, há o problema de passar vírus para os locais que executam outros sistemas operacionais. Assim, o controle de vírus é necessário principalmente para evitar a transmissão de vírus para outros locais que não sejam baseados em software livre. Embora o correio eletrônico seja uma das principais formas de transmissão de vírus, não é a única, portanto é necessário fazer uma varredura geral dos arquivos para evitar a transmissão por outros meios. Em termos de correio eletrônico, a melhor maneira de executar tais produtos é como parte do MTA. O Postfix e o Exim, fornecem meios de incorporar tais filtros.

Um exemplo de software livre que pode atuar como uma solução de antivírus para servidores de Correio Eletrônico, é o Clam Antivírus (<http://clamav.sourceforge.net>). No entanto, é possível garantir que só os arquivos executáveis tenham sido instalados pelo sistema administrador, através da configuração dos sistemas de arquivos, tanto nos servidores quanto nas estações de trabalho. Portanto, é importante que os administradores de sistemas tenham certeza de que os arquivos que estão instalando são confiáveis, por exemplo, através da checagem da assinatura do vendedor da Distribuição nos arquivos.

Há a escolha de produtos proprietários. O Sophos, o RAV e o Vexira, todos possuem boa reputação. A tendência na checagem dos vírus é boa, mas deve se executado com um super acesso ao usuário, o que o transforma em um problema de segurança potencial maior, caso seja comprometido.

10.3.3. Serviços de Calendário e *Groupware*

Calendário é um assunto mal definido no software livre. Isto é devido à ausência de padrões abertos para comunicação entre os clientes e o servidor central. Conseqüentemente, os produtos desenvolvidos até agora, usam a distribuição baseada na *web*, e isso pode não proporcionar às pessoas a cara e o jeito aos quais estão habituadas com o uso do Exchange e do Outlook. Esta área é significativamente frágil no portfólio do software livre.

Guia Livre – Referência de Migração para Software Livre do Governo Federal

Pode-se assumir que os produtos listados na tabela a seguir usam distribuição baseada na *web*, a não ser por afirmação em contrário. Todos eles são parte de conjuntos de *groupware* que possuem uma grande variedade de outros aspectos.

A maior parte dos produtos de *groupware* são escritos em PHP ou Perl, e podem, portanto, ser customizados.

Foram feitas algumas integrações interessantes de recursos nesses produtos.

O *phpGroupware* (<http://www.phpgroupware.org/>) tem uma boa reputação.

Detalhes de Produtos *Groupware*

Produto	Correio	Calendário	Gestão de Documentos	Bate-papo	Lista de Tarefas	Gestão de Contato	Banco de Dados	Cronograma	Programação	Outras facilidades	Comentários
NullLogic	S	S	S	[1]	S		S				
Twiki		S	S	S			S				Uma estrutura mais que um produto
phpGroupWare	S	S	S		S	S	S		S		Difícil encontrar informações adicionais no site
phProject	S	A, B	S		[2]	S		S		Gestão de Projetos, Favoritos, Lembretes, sistemas de busca, sistema de votação, rastreador de solicitações	
Tutos	S	B	S			S		S			
Twiggi	S	S	S		[2]	S			S	Para fazer, Favoritos	

NOTAS

- 1 Padrão IN mais *fórum* (BBS)
 - 2 Funcionalidade de bilhete curto adesivo
- A Registro de cronograma; rastreador de Solicitações, um formulário de gestão de fluxo de trabalho, Lista “para fazer”; projetos dinâmicos
- B Recursos (por exemplo, salas de reuniões, projetores overhead) e gravação de eventos (futuros, bem como os já passados).

O *Horde* é uma estrutura para rodar outros aplicativos. Por exemplo, o *Imp*, um servidor da web, o *Turba*, um gestor de contatos, e o *Kronolith*, um calendário. Veja <http://www.horde.org/>.

O NullLogic aparece somente para oferecer interfaces com o idioma inglês, porém o phProject, o Tutos, o Twigg e Twiki, todos eles dão suporte a uma série de idiomas.

Um produto muito recente do software livre é o *OpenGroupware* da <http://www.opengroupware.org/>. Ele é o aplicativo SKYRiX, anteriormente proprietário, que foi transformado por seus donos em software livre. Tem o objetivo de substituir o Exchange. Ainda não houve tempo suficiente para investigá-lo, porém, tudo indica que irá tornar-se muito influente.

Um outro produto recente é o Kolab (Veja <http://kolab.kroupware.org/>). Esse produto tem um cliente com base no Kmail, e vale a pena pesquisá-lo, particularmente se o KDE for escolhido como interface com o usuário ou se o Kmail for escolhido como MUA para dar suporte a S/MIME.

10.3.3.1. Calendários Pessoais e Agendas

Todos os produtos podem manter calendários pessoais e listas de tarefas, a não ser por disposição em contrário.

10.3.3.2. Calendários de Grupo

O Tutos, o Twigg e o NullLogic, todos eles dão suporte a calendários.

O Tutos permite controle em níveis de faixas, do individual ao grupo de trabalho e do grupo de projeto para todos.

No NullLogic, não se pode manter calendários privados em relação a outros membros do grupo, mas as tarefas sim.

10.3.3.3. Organização de Reuniões

Muitos dos produtos incorporam itens de agendamento de recursos, que podem ser usados para planejar reuniões.

O Tutos permite a alocação automática de pessoas, junto com notificação de correspondência automática para aqueles que não constam do calendário compartilhado (como os que estão em outras organizações). Ele mantém uma lista de aceitações e manda lembretes via correio, caso desejado. O phProject é similar, e lida com notificações de texto SMS.

O NullLogic dá suporte a todos os recursos acima, exceto para alocação de salas.

10.3.3.4. Sincronização de PDA

O phProject tem um recurso adicional que sincroniza com os PDAs baseados em PalmOS. A sincronização do PDA também recebe suporte como parte do Gnome e do Evolution. A maioria do PDAs populares pode ser sincronizada.

10.3.4. Serviços de *Web*

10.3.4.1 Servidores da *Web*

O servidor de *web* software livre mais popular é o Apache, o qual, de acordo com a pesquisa Netcraft (<http://www.netcraft.com/>), possui mais de 60% do mercado e sua porção está crescendo. Uma combinação de produtos cada vez mais popular está sob o nome LAMP: Linux, Apache, MySQL e PHP. Ela fornece uma estrutura para os *sítios* acessarem bancos de dados SQL através da linguagem PHP. Todos os componentes são software livre.

O *Apache* conta com uma extensa gama de ferramentas e suporte associados. Outros servidores poderiam ser utilizados para tarefas específicas, como por exemplo, o *Zope* (veja no item 10.3.4.2) poderia ser usado para gestão de conteúdo.

O projeto Apache contém vários subprojetos, um dos quais é chamado de Jakarta e cobre o lado servidor do uso do Java. O Jakarta em si, consiste de subprojetos, dois dos quais são o Tomcat e o Slide. O Tomcat oferece um produto para *servlets* Java, em conformidade com o padrão JSP, e a habilidade de usar tecnologias como a *Websphere* da IBM. O Slide é implementação baseada em Java da *WebDAV*, que permite gerenciamento de conteúdo. Veja <http://www.apache.org/> para maiores detalhes.

Outros servidores de software livre que vale a pena considerar são o *Zope* e o *Tux*. O *Tux* é um desenvolvimento Red Hat, atualmente chamado de Acelerador de Conteúdo Red Hat. Usa um *kernel* especial e pretende-se que ofereça resposta muito rápida para páginas estáticas.

O *Roxen* é uma outra combinação de servidor de *web* e sistema de gerenciamento de conteúdo, embora seja necessário comprar *add-ons* proprietários para obter total funcionalidade do gerenciamento.

De todos esses, o Apache é, de longe, o mais popular. Ele trabalha atualmente em 63% dos *sítios* públicos e está ganhando fatia de mercado da IIS de forma estável, portanto há muita experiência a considerar ao planejar uma migração. O Apache é um servidor modular, com um motor de protocolo nuclear e uma grande seleção de módulos para propósitos específicos.

10.3.4.2. Portal / Conteúdo

O *Zope* (<http://www.zope.org/>) é desenhado para prover suporte dinâmico de conteúdo da *web* e baseia-se em um modelo orientado para o objeto. É um pacote interessante, pois combina um sistema de gerenciamento de conteúdo com um servidor de *web* e um sistema de modelos em um pacote. O *Zope* também dá suporte a *add-ons* (chamados produtos) e baseia-se na linguagem Python orientada para objeto. É comum encontrar o *Zope* colocado “atrás” do Apache, em uma configuração multiservidor, onde o mesmo serve conteúdo estático e atua como acelerador baseado em *cache* para as partes do *sítio* geridas pelo *Zope*. Um projeto interessante baseado no *Zope* é o *Plone* (<http://www.plone.org/>).

O *JB* software livre (<http://www.jbsoftware livre.org/>) é um servidor de aplicativo baseado em Java. Tem uma boa reputação e está sendo desenvolvido ativamente.

Há atualmente muitos produtos de gerenciamento de conteúdo de software livre, como mostra o *sítio* <http://www.oscom.org/matrix/index.html>.

10.3.5. Serviço de Gestão do Documento

10.3.5.1. Registro e Recuperação

A Gestão do Documento pode, e talvez deva, ser pensada como uma forma de gestão de conteúdo e de fluxo do trabalho. Recomenda-se que seja adotada uma solução com base nas soluções de gestão de conteúdo disponíveis. Em particular, os que usam a *WebDAV* podem prover as soluções mais úteis.

Alguns dos produtos de *groupware* provêm suporte para gestão de documento:

- O Tutos inclui um sistema de gestão de documento que também possui gestão de versão.
- O NullLogic tem capacidade de armazenar, indexar e baixar arquivos simplesmente. Não parece oferecer um sistema de gestão de mudança. Possui um mecanismo generalizado de *query* que pode ser instalado para oferecer indexação.

10.3.5.2. Trabalho colaborativo

Esta função pode ser implementada *ad-hoc* pela simples troca de documentos entre as pessoas.

Para que haja colaboração, as partes precisam acordar quanto ao formato do documento, e atualmente, muitas pessoas usam o **.doc* da *Microsoft* por *default*. Esse *default* significa que as partes precisam confiar umas nas outras, porque esses formatos podem ser eficientes portadores de vírus. Além disso, o **.doc* não é ideal como padrão, porque muda constantemente; o formato usado pelo Office 2000, não é idêntico ao usado no Office 97. Isto significa que as partes também precisam concordar quanto à versão do *software* a ser usado.

Há pressão para que se adote formatos de documentos baseados em padrões, particularmente os baseados no XML OpenOffice.org, que provê um padrão de documento baseado no XML, que poderia ser usado como base para a colaboração.

Uma abordagem mais estruturada seria a de adotar uma solução de gestão de conteúdo/fluxo do trabalho, como descrito acima.

O produto de *groupware* do Tutos permite que os documentos sejam sujeitos ao controle de uma única pessoa ou por todos dentro de um grupo definido. O NullLogic e o Twiki também possuem controles sofisticados.

10.3.6 Serviço de Bancos de Dados

Para sistemas que necessitam de um banco de dados rápido como as aplicações com interface WEB, o *MySQL*; para sistemas que necessitam de maior robustez, o *PostgreSQL*.

10.3.6.1. Bancos de Dados Centrais - baseados em aplicativos

Os sistemas de banco de dados do software livre disponíveis incluem MySQL, PostgreSQL e Firebird. Eles têm características e aplicabilidade significativamente diferentes.

O MySQL é um banco de dados SQL leve favorecido para aplicações *web* e aplicativos similares. É adequado em situações em que a leitura predomina sobre a escrita.

O PostgreSQL é um SGBD completo, comparável aos mais robustos do mercado, com os recursos necessários para se lidar com grandes volumes de dados. Apresenta-se como uma opção para substituir os SGBDs mais voltados às soluções corporativas.

O Firebird é uma versão do banco de dados Interbase da Borland, liberado sob uma licença do software livre. Uma grande parte do código é comum com o Interbase e, como tal, deve ser considerado maduro. Juntamente com o PostgreSQL, o Firebird também se apresenta como uma solução para aplicações corporativas.

O OpenOffice.org 1.1 suporta diretamente o MySQL ou qualquer outro SGBD através das ferramentas de conectividade ODBC e JDBC.

10.3.6.2 Desempenho

O desempenho do banco de dados depende muito do tamanho das tabelas envolvidas e da complexidade das *queries*. Alguns SGDBs do software livre já se mostraram robustos o bastante para

Os produtos proprietários Oracle, DB2, Ingres, Informix, Progress, Mimer e Sybase estão disponíveis para rodar em GNU/Linux e podem ser considerados opções para aplicativos de bancos de dados pesados, para os quais os produtos do software livre não são apropriados. As ferramentas de desenvolvimento do Oracle têm suporte no GNU/Linux.

10.3.6.3 Conectividade (ODBC, JDBC, etc...) – veja item 10.2.6.2 na seção de Estações de Trabalho

10.3.7. Serviços de Segurança

Todos os grupos funcionais devem ser configurados tendo em conta a segurança. A segurança, em nível de *software*, só pode trabalhar se também estiver presente na estrutura maior da gestão de segurança.

10.3.7.1. Criptografia

10.3.7.1.1. Dados em trânsito

Dados confidenciais em LANs internas devem ser criptografados sempre que possível. Dados sensíveis enviados pela Internet, devem estar sempre criptografados. Isto pode ser feito através da canalização de conexões em produtos como *ssh* e *stunnel*.

10.3.7.1.2. Dados armazenados

Dados confidenciais mantidos em dispositivos móveis devem ser criptografados em disco. O ideal é que todos os dados sejam criptografados, porém isso imporia ônus significativos que nem sempre serão aceitáveis. Há vários sistemas de arquivos. Por exemplo, em <http://koeln.ccc.de/archiv/drt/crypto/linuxdisk.htm> há uma discussão dos vários métodos disponíveis.

10.3.7.2. Autenticação

Métodos seguros podem identificar, de modo único, uma pessoa ou máquina que sejam parte de uma comunicação com outras pessoas ou máquinas. Isso inclui assinaturas e infra-estruturas PKI. Nenhum sistema PKI foi testado como parte deste projeto. Todas as autenticações foram feitas em contraposição à senha padrão de um banco de dados LDAP.

10.3.7.3. Autorização

Uma vez autorizadas, determinam o que uma pessoa ou máquina pode fazer e em que circunstâncias. Isso normalmente é parte do sistema operacional ou do código do aplicativo. O *Role Based Access Control* ou *RBAC*, foi definido pelo *NIST* nos Estados Unidos e está disponível para o Linux. (Veja <http://csrc.nist.gov/rbac/>).

10.3.7.4. Servidor Proxy

Há disponível uma gama de servidores proxy do software livre, inteligentes ou semi-inteligentes. Dentre os servidores proxy da *web*, o Squid é o mais popular. Ele possui um produto associado (o squidguard), que evita o acesso a uma lista de *sítios* banidos.

10.3.7.5. Firewalls

Todos os sistemas operacionais atuais do software livre possuem *firewalls* com pacotes de filtragem interna, dos quais a maioria é *stateful*. Os *firewalls* do tipo *stateful* são aquelas que mantêm a informação sobre as conexões em curso e seus fluxos de dados no *firewall*, e permitem a passagem de pacotes que são associados a essas conexões, enquanto eliminam pacotes que não o são. *Firewalls* que não são *stateful*, examinam cada pacote por seus próprios méritos, sem guardar qualquer registro de pacotes anteriores. *Plugins* especializados estão disponíveis para protocolos tais como *ftp* e telefonia H.323, que usam formas de conexão fora dos padrões.

Iptables, atualmente incluídos no GNU/Linux, e *ipfilter*, incluído no FreeBSD são ambos bons produtos *firewall*. O Packetfilter está incluído no OpenBSD, também tem uma boa reputação. Uma boa prática para *firewalls* externos é ter dois diferentes entre a conexão da rede pública e os servidores internos. Não recomendamos a implementação de um modelo apenas.

10.3.7.6. Redes Virtuais Privadas (VPN)

10.3.7.6.1. VPN Abertas

Disponível para a maioria dos sistemas Unix, esta é uma oferta madura e poderosa. Os recursos incluem codificação de chave pública, compressão dinâmica para gestão de largura de banda, e a capacidade de trabalhar com NAT (*Network Address Translation*). Veja também <http://openvpn.sourceforge.net/> para maiores informações.

10.3.7.6.2. FreeSWAN

Esta é uma implementação GNU/Linux dos padrões IPSEC e IKE, e significa que vai interoperar com dispositivos concordantes, incluindo rotas especiais e outros sistemas operacionais. Desde que IPv6 dá originalmente suporte ao IPSEC, o FreeSWAN pode ser preferido se esse padrão for usado. Para beneficiar-se da extensão única do FreeSWAN denominada “Opportunistic Encryption”, que pode automatizar a segurança, os registros DNS devem estar atualizados, o que pode ser limitador. Além disso, entende-se que o IPSEC também pode ter problemas com NAT. Veja também <http://www.freeswan.org/> para maiores informações.

10.3.7.6.3. CIPE

Este é menos maduro do que os outros dois e o suporte de chave pública ainda é experimental. No entanto, pode funcionar com NAT, está disponível para Ambiente proprietário, e navega com Red Hat Linux (você pode até configurá-lo com a ferramenta Controle de Dispositivo de Rede desses sistemas. Mais informação disponível em <http://sites.inka.de/~W1011/devel/cipe.html>.

10.3.8. Serviços de Gestão

Apesar de não ser apenas um serviço do tipo provido por um servidor, a Gestão pode ser vista como uma questão estruturadora, assim como os demais serviços aqui citados. Por este motivo ela está sendo tratada aqui.

O sítio <http://www.infrastructures.org/> provê considerável detalhamento em como gerir uma rede de máquinas, de servidores e estações de trabalho, e tem várias ferramentas de software livre para uma série de tarefas de manutenção.

O sítio mostra que a gestão do Unix, e por extensão, GNU/Linux, tende a ser feita por ferramentas a serem agrupadas a partir de unidades de função única menores. Essa abordagem modular é extremamente poderosa e é o que permite aos administradores do sistema Unix e GNU/Linux serem muito eficientes e efetivos. Também significa que o mercado para *kits* de ferramentas é pequeno, já que os administradores de sistemas tendem a construir, cada um, suas próprias ferramentas.

10.3.8.1. Gestão do Usuário

A manutenção de usuários e de grupos de usuários, incluindo produtos de gestão de senhas como Directory Administrator e gq, permite que os bancos de dados LDAP sejam mantidos.

10.3.8.2. Gestão de Configuração

Embora um cliente bem projetado, gerido de forma central, deva ter um mínimo de instalação local, a atualização de sua configuração sem reinstalação do zero é ainda desejável para grandes redes, das quais se espera que fiquem ativas durante algum tempo. Por exemplo, se um serviço essencial central é mudado, os clientes podem precisar ser re-configurados para usá-lo.

10.3.8.2.1. Manutenção da Configuração Manual

Os Administradores podem manter atualizações de configuração manualmente, já que poderiam programar atualizações. No entanto, problemas similares de sincronização se aplicam. A modificação manual dos arquivos de configuração, geralmente armazenados em arquivos *plain text*, é particularmente sujeita a apresentar erros de digitação.

10.3.8.2.2. Cfengine

O GNU Configuration Engine (<http://www.cfengine.org/>) automatiza a configuração remota de clientes de rede. Ele dá suporte a uma grande variedade de preferências. UNIX e seu poderoso conceito de classe permitem que diferentes grupos de clientes sejam geridos com um mínimo de configuração. Agentes autônomos nos clientes podem manter arquivos de texto, interfaces de rede, *links* de arquivos e permissões, armazenamento temporário e sistemas de arquivos montados.

Alguns dos primitivos que podem ser automatizados usando o *cfengine* são:

- Checar e configurar a interface da rede.
- Editar arquivos de texto.
- Fazer e manter *links* simbólicos, inclusive *links* múltiplos a partir de um único comando.
- Checar e estabelecer permissões de propriedade dos arquivos.
- Remover lixo de arquivos que confundem o sistema.
- Montagem automatizada e sistematizada de sistemas de arquivos (em Unix).
- Checar a presença de arquivos importantes e sistemas de arquivos.
- Execução controlada de *scripts* de usuários e comandos de embalagem.

O Cfengine segue uma estrutura de decisão baseada em classe.

10.3.8.2.3. Configurador de Sistema

O Configurador de Sistema (<http://sisuite.org/systemconfig/>) é parte do *System Installation Suite*, e é usado pelo System Installer. Ele pode configurar e manter muitos componentes de uma instalação GNU/Linux através de muitas distribuições tais como redes, armazenamento, zona de tempo e *booting*.

10.3.8.3. Gestão de Software

Esta seção cobre a manutenção de sistemas de clientes desde a configuração inicial do novo *hardware* até atualizações de *software* em andamento e configuração, e algumas tecnologias para facilitar sua gestão.

10.3.8.3.1. Instalação do Sistema

A instalação do sistema é a configuração inicial do *software* e a configuração necessária para manter uma máquina. Máquinas construídas em fábricas podem não possuir qualquer sistema operacional, ou chegar pré-instaladas com *software*. Máquinas velhas com *software* indesejado também podem ser reutilizadas pela instalação de um sistema novo em seu lugar.

A primeira tarefa de um instalador de sistema é carregar a máquina alvo. O método mais antigo é o de carregar via disquete. Os disquetes estão sendo eliminados, pois são lentos, não confiáveis e oferecem espaço muito pequeno para o *software* de instalação de sistema nos padrões modernos. Muitas máquinas construídas desde 1997 suportam inicialização a partir de CD-ROM pela emulação do setor de *boot* do disquete. Isto é mais rápido e oferece mais espaço para o *software* de *boot* inicial e qualquer outro *software* requerido. O modo mais sofisticado de carregar a máquina é através da rede. Nem todos os *firmware* de BIOS ou placas de rede suportam esse novo recurso. O Pré Execution Environment (PXE) é parte do padrão da indústria *Wired for Management (WIM)* e habilita a maior parte das máquinas compradas desde 1998 a serem carregadas pela rede local.

O instalador deve acessar os meios de instalação apropriados contendo *software* do mais alto nível para ser executado depois que a máquina tiver sido carregada. Caracteristicamente isso terá que ser armazenado em um CD-ROM local ou um servidor de arquivo de rede. Um único disco compacto pode ser usado para armazenar uma imagem de *software*, e a capacidade de um CD-ROM deve ser suficiente para uma estação de trabalho de Administração básica, usando compressão de arquivo regular. Essa imagem pode ser apropriada se for pouco provável que o *software* mude, ou se só for necessária uma base de instalação estável para a adição de *software* adicional. Geralmente, uma instalação de rede é mais flexível, pode ser mais rápida, oferece maior capacidade, e escala melhor as instalações paralelas e múltiplas, do que compartilha discos de instalação entre clientes.

O instalador de sistema transfere o *software* da mídia selecionada para o disco rígido local da máquina alvo, e prepara-a para a inicialização. Isso vai envolver conhecimento do *hardware*, checagem da capacidade do disco e configuração de detalhes de rede.

Alguns dos possíveis métodos de instalação são discutidos abaixo:

1. Instalação Manual

A instalação mais básica se dá através de um administrador de sistema. O *software* vem em discos compactos, inclusive em um disco de instalação *bootable*. Algumas indicações automáticas podem guiar o administrador, mas em última instância, toda a customização é manual. Já que a seleção do pacote, a divisão do disco rígido, a configuração do *hardware* e os detalhes da rede devem todos ser inseridos manualmente, este processo é demorado e tende ao erro humano. A maior parte das distribuições têm seu próprio programa de instalação.

2. Clonagem de Imagem

Se clones quase idênticos são adequados, uma origem pode ser manualmente instalada e depois replicada. Distribuições como Knoppix e o Kurumin (que carrega um ambiente GNU/Linux completo de um único CD-ROM - Veja <http://www.guiadohardware.net/kurumin/>) e outros,

podem ser usados para copiar as imagens do sistema de arquivos da origem para outras máquinas. Configurações e customizações podem ser adicionadas por *scripts* executados antes ou depois da instalação. Uma vez que sistemas de arquivos inteiros podem ser copiados para o disco em vez dos arquivos neles contidos, isto pode oferecer um tempo mais rápido de instalação. No entanto, configurar clones não idênticos é menos eficiente e requer habilidade maior no sistema.

3. Instalação Totalmente Automática

A FAI (<http://www.informatik.uni-koeln.de/fai/>) instala a distribuição Debian automaticamente. Os pacotes de *software* são acessados de um *sítio* Debian, que pode ser espelhado localmente para obter velocidade ou customização. O *kernel* de instalação disponibilizado pode ser carregado a partir da rede ou de disquete, porém o carregamento por CD-ROM ainda está sendo aperfeiçoado atualmente. Embora o FAI tenha sido desenhado para replicação idêntica de máquinas *clustered*, o *software* cfengine descrito acima é usado para configuração de sistema e permite extrema flexibilidade, se necessário.

4. System Imager

O System Imager (<http://www.systemimager.org/>) faz instalação de sistema, configuração e manutenção para grandes redes de máquinas, de preferência com *hardware* similar, através de várias distribuições. Ele pode carregar a partir do disquete, do CD-Rom ou de servidores de rede PXE. Ambas as instalações, tanto Debian quanto Red Hat foram testadas, porém o *software* System Configurator usado ajuda a dar suporte a todos os GNU/Linux distribuições.

Uma origem é instalada e configurada manualmente. Então seus sistemas de arquivos são espelhados para um servidor de imagem, de onde as máquinas alvo são instaladas. Se a origem estiver atualizada, essas mudanças serão propagadas para clientes usando rsync. Embora rsync envie minimamente as diferenças de arquivos pela rede, ele pode requerer memória significativa para fazê-lo. Como as modificações são relativas à origem, o System Imager é mais apropriado para clientes alvo com *hardware* idêntico ou muito parecidos.

5. Kickstart da Red Hat

O Kickstart (<http://www.tldp.org/HOWTO/KickStart-HOWTO.html>) é o *software* de instalação automatizada da Red Hat. Ele instala distribuições da Red Hat de CD-ROM, disco rígido ou rede, e carrega a partir de rede, CD ou disquete. O instalador Anaconda oferece texto ou interfaces gráficas e pode ser interativo ou totalmente automatizado por um arquivo de configuração. O *software* de detecção de *hardware* Kudzu fornece uma gama de dispositivos automaticamente. Opções de instalação generalizadas podem ser montadas no arquivo de configuração e extensões adicionadas com *scripts* de instalação pré e pós.

Com o seu *software* inteligente de configuração e detecção, o kickstart pode ser utilizado para automatizar instalações similares entre uma variedade de alvos de *hardware*. A seleção de pacotes do Red Hat distribuição padrão é direta, porém as atualizações ou extensões também podem ser incluídas pela customização do processo kickstart.

10.3.8.3.2. Manutenção do Software

Instalações de *software* não permanecem estáticas durante seu tempo de vida. Serão lançadas atualizações de *software* como o de segurança ou de conserto de problemas, após a instalação inicial. Além disso, será requisitada a remoção ou adição do pacote para gerenciar o *software* sem a reinstalação de um sistema inteiro.

Guia Livre – Referência de Migração para Software Livre do Governo Federal

Onde for possível, devem ser realizadas atualizações utilizando técnicas de *pull* em vez de *push*. A decisão de *pull* atualizações deve ser tomada por uma máquina, seja servidor ou estações de trabalho, depois de ter verificado a si mesma em comparação a um servidor principal. As atualizações não devem estar sob o controle dos usuários. Desta forma, as máquinas podem ser mantidas no mesmo nível de revisão.

1. Manutenção do Software Manual

Os administradores do sistema podem manter o *software* atualizado manualmente. Isso pode envolver acessar o cliente alvo de forma remota, copiar pacotes de atualização e instalá-los com o gerenciador de pacote original da distribuição. No entanto, embora isso ofereça um rígido controle ao administrador, tende a erros e dificulta a sincronização de grandes coletas de máquinas. Algumas distribuições oferecem ferramentas de atualização para manter seus pacotes padrão, mas ainda requerem intervenção manual e podem não disponibilizar extensões à distribuição básica.

2. Debian APT

O APT é um jogo de ferramentas padrão fornecido com a distribuição Debian GNU/Linux, que permite atualizações automatizadas ao *software* instalado em uma máquina. Ele pode checar dependências entre pacotes de *software* instalados na máquina e disponíveis dos *softwares* depositários, os quais ele foi configurado para checar, e para recuperar e instalar atualizações relevantes disponíveis de um depositário. As organizações podem montar e manter seus próprios depositários do *software* a ser instalado em seus clientes (O Debian inclui ferramentas para montar e manter tais depositários), podem usar depositários disponibilizados pelo Debian e outros, ou usar qualquer combinação dessas fontes de *software* atualizado. O APT foi transportado para trabalhar em sistemas operacionais baseados em RPM, tais como Conectiva, Red Hat Linux e Mandrake, onde provê funcionalidade similar, e de algumas formas aperfeiçoadas por comparação ao Red Carpet.

3. Novell Ximian Red Carpet

Novell Ximian Red Carpet Daemon (<http://www.ximian.com/products/redcarpet/>)

É uma ferramenta de atualização de software disponibilizada gratuitamente pela Ximian. Começou como um gerenciador de pacote gráfico para o Novell Ximian Desktop 2 (pacote de software e interface da Ximian), porem oferece agora acesso seguro à linha de comando remota e mais canais de software inclusive atualizações de distribuição. Mandrake, SuSE e Red Hat são suportados atualmente. Ele oferece administração remota fácil e automação possibilitando que um número bem grande de clientes possa ser mantido de forma central. Pode ser configurado para atualizar software a partir de canais customizados.

Novell Ximian Red Carpet Enterprise

É um produto servidor proprietário, utilizado para facilitar a gestão de grandes coletas de software. O Novell Ximian Red Carpet Enterprise oferece a vantagem de controle centralizado de políticas de atualização de pacotes, permitindo o agendamento de tarefas e mantendo o tráfego de rede para busca de atualizações dentro da LAN. Sem o Novell Ximian Red Carpet Enterprise todas as estações de trabalho utilizam a diretamente o

acesso a internet para buscar atualizações. Em ambientes corporativos esse comportamento não é desejado. O Novell Ximian Red Carpet Enterprise consolida no servidor corporativo todas as atualizações necessárias, reduzindo o consumo do link de internet.

A interface gráfica não deve ser usada, pois permite aos usuários controlar as atualizações. A interface com a linha de comando deve ser incorporada aos scripts que atualizam as máquinas automaticamente.

4. Red Hat Enterprise Network

A Red Hat oferece uma gama de serviços de atualização de *software* como parte de seu Enterprise Network (http://www.redhat.com/software/rhen/software_delivery). O mais poderoso é seu Satellite Server, que permite *customização* completa de atualizações e errata. Todos os servidores dão suporte aos seus clientes padrão, os *Update Agent*, para distribuição. Os comentários contra a permissão de uso da interface gráfica podem ser aplicados aqui da mesma forma que os do Red Carpet acima.

10.3.8.4. Gestão de *hardware* e monitoramento de sistema

O *hardware* pode ser monitorado com relação a falhas e falhas potenciais, por exemplo, utilizando o SMART – discos rígidos habilitados e *hardware* de checagem de saúde do sistema. Sistemas de *hardware* e *software* também deveriam ser monitorados com relação às falhas, falhas potenciais, ausência de serviço e falta de capacidade.

10.3.8.4.1. MRTG e Snmpd

O MRTG (Multi-RouterTraffic Grapher, <http://people.ee.ethz.ch/~oetiker/webtools/mrtg/>) é uma ferramenta de monitoramento desenhada originalmente para rastrear e representar em gráfico o uso da capacidade dos links de rede. No entanto, desenvolveu-se como uma ferramenta capaz de rastrear virtualmente qualquer mudança de quantidade, e pode ser usada para monitorar variáveis como processador, memória e uso do espaço do disco, uso dos serviços da rede incluindo estatísticas sobre volumes de correspondência processada, páginas da *web* servidas, etc, temperatura do sistema e velocidades da ventilação, e outras variáveis.

O Snmpd (Simple Network Management Protocol Daemon, <http://net-snmp.sourceforge.net/>) é um servidor de gestão de sistema que pode operar em qualquer estação de trabalho de uma organização. Disponibiliza informação de gestão de sistema para clientes: tipicamente para um cliente SNMP central, que agrega estatísticas a partir de algumas máquinas. O MRTG pode atuar como tal cliente e desempenhar essa função, fornecendo uma visão geral gráfica de um grande número de máquinas de clientes.

10.3.8.4.2. Nagios

O Nagios (anteriormente conhecido como NetSaint <http://www.nagios.org/>) é um hospedeiro customizável, provê serviço e monitoramento de rede e sistema de gestão. É capaz de monitorar serviços de rede e desempenhar vários procedimentos de recuperação, caso descubra que um serviço não está disponível ou está apresentando problemas, inclusive invocando *scripts* de recuperação automática e alertando os administradores do sistema para o problema. O Nagios também pode fornecer relatórios e visões gerais do status corrente e passado dos serviços que monitora.

10.3.8.4.3. smartd

O conjunto de ferramentas SmartMon Tools (<http://smartmontools.sourceforge.net/>) inclui um *daemon* chamado *smartd*, que é desenhado para monitorar o SMART (Self-Monitoring, Analysis and Reporting Technology), função dos modernos *drives* de disco rígido. Como esses dispositivos são um dos componentes que mais falham em um computador moderno, o SMART pretende monitorar parâmetros de *drive* e notificar um administrador de sistema sobre falhas potenciais antes que elas aconteçam. O *smartd* é desenhado para receber essas notificações e agir, através do alerta ao administrador do sistema.

10.3.8.5. Serviço de Impressão

10.3.8.5.1. LPRng

LPRng (<http://www.lprng.com/>) é uma implementação desenvolvida do antigo sistema BSD padrão *lpr/lpd*. Contem alguns aprimoramentos que o fazem muito mais robusto e fácil de manusear do que os produtos originais. O autor é particularmente entusiasmado ao garantir que o LPRng é seguro. Até pouco tempo atrás, essa era provavelmente a escolha para gestão de impressora, porém recentemente o CUPS progrediu e a escolha agora é menos evidente.

10.3.8.5.2. Sistema de Impressão Unix Common

O *Common Unix Printing System* ou CUPS (<http://www.cups.org/>) é desenhado para ser um sistema de impressão Unix “pronto para a empresa”. É baseado no padrão *Internet Printing Protocol* ou IPP e incorpora uma função navegação, que disponibiliza detalhes dos nomes e características de impressoras, para serem distribuídos automaticamente pela rede. O CUPS também incorpora uma interface usuário baseada na *web* para administrar e configurar as impressoras. Os *drivers* são disponíveis para as impressoras mais comuns.

10.3.8.5.3. Kprint e GnomePrint

Tanto KDE quanto Gnome incorporam seus próprios subsistemas de impressão, que podem ter interface com aplicativos do usuário com a maior parte dos sistemas cilíndricos de impressão utilizados, incluindo LPRng e CUPS.

10.3.9. Serviço de Backup e Recuperação

Supõe-se que todos os dados do usuário e da Administração estejam em um ou mais servidores. É necessário ser capaz de incluir arquivos de forma incremental, encontrar itens removidos com arquivos específicos e restaurar arquivos individuais ou sistemas inteiros de arquivo. Fazer o *backup* dos dados do usuário tende a ser mais fácil em sistemas de software livre do que no ambiente proprietário, porque os arquivos de dados do usuário, incluindo seus dados de configuração, estão normalmente contidos em um único diretório.

10.3.9.1. Remoção e Recuperação

Esses dois programas são disponibilizados como parte da maioria das distribuições e são utilizados algumas vezes junto com *tar* e *cpio* em *scripts* *customizados* para *backup* e recuperação de máquinas individuais.

10.3.9.2. Amanda

Amanda (Veja <http://www.amanda.org/>) é um produto cliente-servidor desenhado para *backup* de múltiplas máquinas em um dispositivo individual. Também é capaz de fazer *backup* de ambientes proprietários através do Samba.

10.3.10. Outros serviços

10.3.10.1. Serviço de Tempo

É essencial, em um ambiente altamente interligado por rede, que todas as máquinas, tanto servidores como estações de trabalho, tenham a mesma noção do tempo corrente. Um ou mais servidores são designados como servidores mestres e eles obtêm seu tempo de um relógio anexo ou de servidores de tempo externos na Internet. Todas as outras máquinas são escravas, sincronizadas a esses mestres.

A sincronização do tempo pode ser feita executando o *ntp* (<http://www.ntp.org/>) nas máquinas. Ele pode manter, facilmente, uma rede de máquinas no intervalo de um segundo de uma para outra.

O Chrony (<http://go.to/chrony/>) é uma alternativa ao *ntp*. Possui alguns recursos que o fazem mais apropriado para cruzamentos NTP altamente estratificados do que o *ntp*, enquanto o *ntp* é melhor para cruzamentos de estratificação baixa, que podem ter que fazer interface diretamente com coisas tais como receptores GPS e relógios atômicos. Há também produtos de software livre para ambiente proprietário, que são úteis em um ambiente misto, como Automachron e nettime. O *sítio* <http://go.to/chrony/> fornece detalhes de ambos.

10.3.10.2. Serviços de Infra-estrutura de Rede

Estes são os serviços necessários para operar uma rede baseada em um TCP/IP.

10.3.10.2.1. Roteamento

Roteadores possibilitam a divisão de uma grande rede em redes menores interconectadas. Os roteadores têm a função de direcionar os pacotes de uma sub-rede para outra, para habilitá-las a chegarem a um destino. A construção de roteadores requer um bom entendimento dos protocolos básicos e muitas Administrações provavelmente desejariam comprar roteadores proprietários.

Para os que desejam construir seu próprio roteador, existem dois produtos: Bird (<http://bird.network.cz/>) e GNU Zebra (<http://www.zebra.org/>).

10.3.10.2.2. DNS

Uma rede TCP/IP necessita de alguns meios para traduzir endereços IP para nomes significativos do domínio humano e vice-versa. O DNS é um protocolo junto com alguns servidores de intercomunicação, cada um com dados. O DNS é básico para o funcionamento da Internet. Há alguns programas para construção de servidores DNS, inclusive o BIND (<http://www.isc.org/products/BIND/>), o MyDNS (<http://mydns.bboy.net/>), e o MaraDNS (<http://www.maradns.org/>). O BIND é o mais utilizado.

10.3.10.2.3. DHCP

O DHCP é um protocolo descrito em <http://www.dhcp.org/>, que permite às máquinas obterem seus detalhes de rede do computador ao ser ligado, a partir de um servidor central ou de vários servidores. O DHCP permite o uso eficiente dos endereços IP e re-aloca os endereços onde for possível fazê-lo. Também permite a administração centralizada de vários endereços globais tais como *gateways* e nomes de servidores. O principal produto vem de <http://www.isc.org/products/DHCP/> e consiste de um aplicativo cliente-servidor. O cliente deve operar em todas as máquinas de clientes participantes. Os dois produtos vêm com a maioria das distribuições padrão.

10.3.10.3. Servidores de Arquivos

Servidores de arquivos de rede permitem às máquinas anexas à rede acessar o armazenamento de arquivos em uma máquina remota, como se a mesma fosse local.

10.3.10.3.1. NFS

Esse é o padrão de fato e está em uso há muitos anos. O subconjunto implementado comumente não oferece grande segurança, embora seja definida uma variante de segurança e implementada em algumas variantes comerciais Unix.

O NFS consiste de um serviço que compartilha arquivos de um servidor para clientes conectados a rede. Há uma autenticação mínima dos usuários anexos na versão Linux.

O NFS é parte padrão de muitas distribuições.

10.3.10.3.2. Samba

O Samba é um produto que implementa o protocolo SMB. Veja 14.5.1 para uma descrição mais detalhada. É crucial para a integração do software livre e dos sistemas com base em ambiente

proprietário e vem com a maior parte das distribuições padrão. Seu uso está descrito com algum detalhamento no capítulo 14.

10.3.10.3.3. Netatalk

Para os que possuem máquinas Apple Macintosh, o netatalk oferece a implementação do protocolo Apple Talk

(Veja <http://netatalk.sourceforge.net/>).

10.3.10.3.4. OpenAFS, CODA e Intermezzo

Esses produtos implementam um sistema de arquivo distribuído em vários graus. Com tal sistema, o acesso aos arquivos pode continuar quando a rede falha, porque o *caching* local possibilita a aparência de estar conectado. Esse não é um problema trivial e os produtos oferecem soluções de diversas formas. Esse tipo de sistema de arquivo é realmente necessário em *laptops* ou máquinas anexas a uma conexão transitória. A outra forma de prover a mesma funcionalidade, é possuindo armazenamento local sincronizado com um servidor central, periodicamente. Veja <http://www.openafs.org/>, <http://www.coda.cs.cmu.edu/> e <http://www.inter-mezzo.org/> para detalhes de cada produto.

O *sítio* <http://www.inter-mezzo.org/docs/bottlenecks.pdf> contém uma discussão detalhada das características de todos acima.

10.3.10.4. Serviço de Diretório

Provisão de consulta rápida de nomes, endereços e dados associados.

O padrão mais popular de serviços de diretório é o LDAP. Esse é um protocolo aberto implementado em muitos produtos, por exemplo, Evolution e OpenOffice.org. O LDAP trabalha com definições de dados chamados esquemas e é possível às Administrações desenvolver seus próprios esquemas *customizados*. Infelizmente os esquemas usados pelos aplicativos não são sempre compatíveis uns com os outros, o que significa, por exemplo, que é difícil para o OpenOffice.org ler os dados do Evolution e vice-versa.

O serviço de diretório OpenLDAP adapta-se ao padrão LDAPv3 e à versão 2.1 e mais tarde poderá ser configurado com uma gama de bancos dados (tais como flat file, SQLs).

A maior parte dos conjuntos de *groupware* fornecem alguma forma de serviço de diretório, porém poucos estão atualmente compatíveis de fato com o LDAP. O Open LDAP ainda possui algumas peculiaridades, principalmente com relação ao seu uso em grandes redes que possuem milhares de objetos em seus Diretórios; e essas existem devido ao fato deste Diretório ainda não possuir algumas facilidades e funções administrativas que atualmente se encontram disponíveis em Diretórios proprietários. Porém, já estão em andamento projetos objetivando a implementação dessas funcionalidades, de forma que sua administração seja mais amigável e facilitada para esses ambientes. -

O OpenOffice.org, o Evolution e o Mozilla oferecem funções integrais de catálogos de endereços. No entanto, os formatos de armazenamento usados não são passíveis de trocas. É necessária alguma adaptação para possibilitar o trabalho interativo.

10.3.10.5. Suporte a legados

10.3.10.5.1. Emulação terminal

O uso do *xterm* com um jogo variável de ambiente TERM apropriado, pode emular a maior parte dos tipos terminais baseados em caracteres, por exemplo VT220 e VT100. Há um emulador 3270 específico chamado x3270. Pode-se encontrar emulações baseadas em páginas em produtos proprietários.

10.3.10.5.2. Exibição remota

Há uma discussão a respeito na Seção 11.3 abaixo.

10.3.10.5.3. Emulação

Há uma discussão a respeito na Seção 11.4 abaixo.

11. Migração de Aplicativo – Visão Geral

Uma vez redigida a lista de aplicativos, ela pode ser agrupada nas categorias a seguir:

11.1. Aplicativos proprietários que possuem um software livre equivalente

Alguns aplicativos, por exemplo, o Office, o Lotus SmartSuite, o WordPerfect, o Framemaker, o Quark Express e o Photoshop, têm equivalentes que operam originalmente no software livre, inclusive o OpenOffice.org, o Gnumeric, o Evolution e o GIMP. Nesse caso, é preciso testar o produto software livre para garantir que forneça a funcionalidade necessária.

11.2. Aplicativos proprietários que operam em um ambiente software livre.

Alguns aplicativos, como o Acrobat Reader da Adobe, têm uma versão que funciona originalmente no software livre. Se não houver alternativa em software livre para o aplicativo, tudo que é preciso é garantir que todos os recursos necessários sejam implementados na versão proprietária. Se houver uma alternativa software livre e for aceitável uma migração parcial, há que se fazer uma escolha baseada nos recursos oferecidos pelo proprietário e pelos aplicativos software livre.

11.3. Software que pode ser acessado por exibição remota.

Uma outra abordagem é fazer operar os aplicativos em um servidor e transportar a tela para a estação de trabalho; essa é a abordagem cliente leve. Produtos como Windows Terminal Server, Citrix e Graphon permitem que os aplicativos funcionem em um servidor operando com software proprietário em uma plataforma multiusuário. Isso significa que um aplicativo feito para operar em uma estação de trabalho no modo cliente individual, poderá ter que ser alterado para funcionar nesses produtos. Isso não será possível sem o código fonte, e vendedores externos podem não desejar ajudar.

O mais sofisticado desses produtos, o Citrix, tem sua própria linha de protocolo, ICA, que é extremamente boa, particularmente em conexões de banda estreita. Ele pode operar múltiplos servidores com carga balanceada e possui outros recursos. Existem clientes gratuitos do ICA que operam em GNU/Linux.

Todos esses produtos contam com *software* proprietário de fonte fechada e o Citrix é particularmente caro. Requer uma licença de servidor Windows, uma licença de Citrix e uma licença de Servidor de Terminal Windows, caso seja usado um cliente não Windows. Adicionalmente, serão necessárias Licenças de Acesso ao Cliente para cada estação de trabalho que use o *software*. A licença do Citrix é baseada em usuários concorrentes, portanto essa abordagem pode reduzir custos se houver muitos usuários que necessitem acesso a um aplicativo desde que o acesso do concorrente seja baixo. Há estudos de casos documentados em <http://www.citrix.com/press/news/profiles/>, que mostram que a economia realizada utilizando estações de trabalho em cliente leve é suficiente para justificar a mudança dos aplicativos para um servidor. O Citrix também possui produtos que permitem o transporte dos aplicativos Unix da mesma forma usando o ICA e exibidos em uma tela de cliente leve.

O Windows Terminal Server oferece funcionalidade similar ao Citrix, exceto por usar seu próprio protocolo, RDP. O cliente GNU/Linux para RDP é bom, porém ainda é considerado por

Guia Livre – Referência de Migração para Software Livre do Governo Federal

alguns como código beta. O RDP costumava ser muito ineficiente em comparação ao ICA, porém agora a diferença é pequena, se não insignificante.

O Citrix tem vários recursos como equilíbrio de carga, o que faz dele a melhor opção para instalações de larga escala, onde o custo extra pode ser justificado. Tanto o Citrix quanto o Windows Terminal Server podem introduzir latência no aplicativo, se o tamanho dos servidores não estiver dimensionado corretamente e a rede não for suficientemente rápida.

O Tarantella (<http://www.tarantella.com/>) fica em um servidor, entre a estação de trabalho e os servidores aplicativos. Ele agrega o produto do Citrix no Windows e em outros aplicativos, funcionando no computador central Unix e IBM, e envia o resultado para um navegador na estação de trabalho. Ele usa sua própria linha de protocolo proprietário, AIP, que é aparentemente razoável em bandas estreitas. No entanto, ele aumenta a latência porque fica entre o usuário e o aplicativo e, portanto, torna mais lenta a conexão entre os dois.

Conforme mencionado acima, o CodeWeavers produz agora uma versão servidor de seu produto *CrossOver Office*. Ele trabalha com o cliente conectado com segurança ao servidor central e tem uma sessão X exibida de volta a ele. Isto significa que a comunicação com o servidor central é codificada e comprimida, porém também requer largura de banda suficiente para suportá-lo, já que é baseado em X. Não foram feitos testes no requisito largura de banda, porém é provável que seja maior do que para o ICA (Citrix) ou AIP (Tarantella).

O VNC é um produto software livre desenvolvido pela AT&T, desenhado para exibir uma sessão, funcionando em outra máquina. Consiste de um servidor e um cliente, os quais são disponibilizados para ambiente proprietário, Unix e GNU/Linux. O VNC permite que os aplicativos funcionem em um ambiente e a exibição seja feita em outro. Usa seu próprio protocolo aberto, RFB, sobre o TCP/IP, o que não é tão eficiente quanto o ICA (Citrix) ou AIP (Tarantella), e portanto, necessita de uma alta banda larga de rede (como 100 Mb/s) para trabalhar bem. Infelizmente o servidor VNC para ambiente proprietário também não é tão eficiente quanto a versão Unix e pode necessitar de mais poder de processamento do que se poderia esperar. O VNC pode ser muito útil para uso ocasional de administrador de sistemas, permitindo que uma pessoa central controle uma estação de trabalho. Nessas circunstâncias, poderia ser aceita uma alta latência.

11.4. Software que funcionará sob um emulador

Se nenhum caso acima oferecer um meio de fazer operar o aplicativo ou um substituto, então pode ser possível fazê-lo operar na forma original, porém com seu ambiente operacional normal emulado em cima de um sistema operacional software livre. Uma boa discussão dos assuntos relativos a esta abordagem pode ser encontrada em

<http://www.linuxmednews.com/linuxmednews/967526746/index.html>. Todas essas técnicas têm implicações de licença porque podem envolver a operação de múltiplas cópias do aplicativo proprietário e/ou do sistema operacional.

É provável que a maior parte desta seção seja aplicável Windows, porém como as técnicas podem aplicar-se a outros Cenários, são discutidas aqui, ao invés do Capítulo 14.

Há dois tipos de emulação:

11.4.1. Emulação de *hardware*

Produtos como Vmware e Win4lin fazem emulação de *hardware*. Eles possibilitam que um sistema operacional de PC normal funcione como um aplicativo nível usuário imitando o *hardware*

Guia Livre – Referência de Migração para Software Livre do Governo Federal

Intel PC em interfaces de *software* e provendo, portanto, uma máquina virtual. Isso permite que um sistema operacional legado e seus aplicativos operem em cima de uma plataforma software livre.

O VMware não é rigorosamente um emulador. Ele permite que a maior parte das instruções passem através do processador, o que significa que somente funcionará em uma máquina de arquitetura x86. É a opção mais completa porém é proprietário e pode consumir muitos recursos da máquina.

O Win4lin é similar ao VMware, e também é um produto proprietário, mas é menos caro. Pode ser uma boa solução em casos simples, por exemplo, para operar aplicativos *office* somente. É um componente do produto Lindows, que está sendo vendido em *hardware* de baixo custo para usuários domésticos. (Pelo fato de aparentemente não usar contas de usuários não privilegiados para manter a segurança, o Lindows não deve ser recomendado para Administrações, sem uma consideração cuidadosa das implicações de segurança).

Pelo fato de que a abordagem de emulação de *hardware* requer licenças completas para o sistema operacional e o aplicativo proprietários, junto com o custo do emulador, deveria ser visto como uma forma de operar um número pequeno de aplicativos legados que são difíceis de migrar.

Há produtos de servidores VMware e Win4lin que podem reduzir custos de licença se o *software* proprietário permitir uma licença de usuário **concorrente**, ao invés de **licença de usuário potencial**.

Há aplicativos software livre que emulam completamente um ambiente Intel IA-32, por exemplo, o Bochs, mas provavelmente ainda não estão prontos para serem usados pela Administração.

11.4.2. Emulação de software

A emulação de *software* permite que programas escritos para um ambiente proprietário trabalhem diretamente no sistema operacional software livre. Quaisquer chamadas ao sistema feitas por eles são mapeadas na interface do sistema do software livre equivalente. Isso significa que não é mais necessária uma cópia do sistema operacional proprietário.

O Wine permite que aplicativos escritos para Windows funcionem em GNU/Linux através da emulação de *software*. O Wine é descrito em detalhes no Apêndice B. O principal problema que o Wine precisa resolver é o grande número de chamadas ao sistema (inclusive *bugs*) que precisa suportar.

O código Wine software livre está disponível em <http://www.winehq.org/> ou em CodeWeavers em <http://www.codeweavers.com/technology/wine/download.php>.

A CodeWeavers produz dois produtos proprietários, o CrossOver Office e o CrossOver Plugin, que são baseados no Wine e desenhados para dar suporte a aplicativos Windows específicos. Embora os produtos sejam proprietários, periodicamente são enviadas modificações de código de volta à versão software livre do Wine.

O CrossOver Office é desenhado para permitir que aplicativos como Office e Lotus Notes funcionem originalmente em GNU/Linux. Há alguns assuntos que ainda estão por resolver, mas o produto está em ativo desenvolvimento. No entanto, esta abordagem pode ser apropriada para alguns usuários dependendo de suas necessidades. O CrossOver Office está disponível agora como produto servidor, o que significa que não precisa ser totalmente instalado na estação de trabalho e que pode prover funcionalidade similar ao Citrix.

O CrossOver Plugin é desenhado para permitir que *plugins* de navegador que normalmente só trabalham em ambiente proprietário, funcionem em Netscape, Mozilla e Galeon em GNU/Linux. Este produto está disponível a mais tempo que o CrossOver Office e trabalha muito bem.

Usando essas técnicas, remove-se o custo da licença do sistema operacional proprietário, mas não da licença do aplicativo. A licença do aplicativo precisa ser examinada minuciosamente para se ter certeza de que não proíbe o funcionamento do aplicativo sem o ambiente proprietário. Essa restrição é usada em alguns aplicativos proprietários como uma tática para trancar, embora a imposição legal seja questionável.

11.5. Software que pode ser recompilado em software livre

Para aplicativos escritos em casa ou em nome da Administração, e para os quais haja código fonte disponível, o *software* pode ser transferido para funcionar em uma plataforma software livre. Em geral, o problema de transferir código fonte em qualquer linguagem não é a compilação, mas o uso pelo código de bibliotecas do sistema, incluindo o ambiente gráfico e o sistema operacional. Isso pode significar muita intervenção manual para migrar o código. Adicionalmente, quaisquer suposições sobre o ambiente base, tal como nomeação do arquivo, fará necessária a mudança do código fonte ou a replicação do ambiente, independentemente da linguagem usada.

1. **Java.** Se o *software* Java foi escrito de acordo com a especificação Java, o programa deve funcionar sem quaisquer problemas. No entanto, caso quaisquer extensões proprietárias tenham sido utilizadas, o código terá que ser mudado e deverão ser usados módulos padrões.

2. **Visual Basic.** Um produto proprietário chamado DeLux (<http://www.deluxsoftware.com/>) pode ser usado para converter o código Visual Basic para Kylix (veja item 4 abaixo) e pode trabalhar em GNU/Linux originalmente. As ferramentas de desenvolvimento da *Microsoft* podem converter o código Visual Basic para **.NET** e produzir código **CIL**. O projeto software livre Mono permite que esse código funcione em GNU/Linux. O Mono está sendo desenvolvido muito rapidamente no momento e qualquer aplicativo poderá ou não funcionar, dependendo da forma como vai interagir com as bibliotecas, como o exibidor de tela.

3. **C#.** Está recebendo suporte crescente GNU/Linux, e a Ximian produziu um compilador como parte do projeto Mono, acrescentando as ligações C# a componentes cruciais da estação de trabalho Gnome. O projeto Mono inclui um intérprete que permite que o código CIL produzido por ferramentas de desenvolvimento proprietárias trabalhem em GNU/Linux sem alterações. O projeto Mono e o uso da estrutura de desenvolvimento **.NET** é uma área muito viva do software livre atualmente e a posição muda rapidamente.

4. **Pascal and Delphi.** O Pascal, como linguagem livre, é menos usado atualmente, porém é o componente de codificação essencial da ferramenta de desenvolvimento rápido Borland Delphi. Borland tem um equivalente nativo GNU/Linux da Delphi chamado Kylix. O Kylix 2 e o Delphi 6 são feitos para usarem sintaxe de código compatível e possuem ambientes de suporte idênticos.

5. **C e C++.** Programas escritos para padrões ANSI devem ser recompilados e funcionar enquanto as bibliotecas do sistema base usadas forem compatíveis. Por exemplo, sistemas escritos especialmente para ambiente proprietário, em geral, não irão recompilar e funcionar corretamente em GNU/Linux, por conta de jogos de chamadas muito diferentes do sistema operacional e das bibliotecas *runtime*, como o sistema de janelas. Pode-se resolver essa falta de combinação, freqüentemente, com a compilação do código com Winelib, parte do projeto Wine.

PARTE 4
Planejando
a Migração

12. Cenário 1 - Windows

A Administração tem um ou mais domínios interconectados de Grupos de Trabalho Windows, Windows NT PDC/BDC ou Windows 2000 Active Directory. Todos os usuários possuem estações de trabalho Windows. Todos os aplicativos centrais funcionam em servidores Windows.

Ao longo deste capítulo, a palavra Windows significará uma versão do *Microsoft Windows*. Onde a versão precisa for importante, será expressa. Os exemplos de códigos são baseados em um sistema Red Hat Linux; outras distribuições podem conter diferenças sutis.

O conteúdo deste cenário deve ser lido em conjunção com os comentários gerais feitos nos Capítulos anteriores.

12.1. Como Planejar a migração

Recapitulando o que foi dito no Capítulo 5, o planejamento para a fase de transição é muito importante, o sucesso de um projeto software livre será julgado tanto pela forma que foi executada a transição, bem como pela qualidade final do serviço. É provável que qualquer transição prática de um sistema para outro dure um período de meses ou mesmo anos. Durante esse tempo, dados precisam ser movidos, pessoas treinadas, *software* instalado, e o negócio da Administração precisa continuar sem interrupções.

Será necessário um planejamento cuidadoso, e grandes administrações devem passar por uma fase piloto para testar o plano antes de colocá-lo em funcionamento em larga escala.

12.2. Domínios

Este Cenário pode ser dividido na seguinte forma:

12.2.1. Modelo de “grupo de trabalho” do Windows

Um grupo de computadores Windows co-operando de forma solta na rede, declarando-se parte do mesmo “grupo de trabalho”. Não há aspectos de segurança para grupos de trabalho - eles servem somente a máquinas de grupo, convenientemente em listas de navegadores.

Usuários que desejarem compartilhar arquivos com outros, podem permitir o compartilhamento, de partes ou total da hierarquia de seu diretório, para acesso generalizado ou com requisição de senha.

Não há coordenação de nomes de usuário e senhas neste modelo. Na verdade, em algumas versões do Windows, não há o conceito de propriedade do arquivo.

A migração de um esquema de grupo de trabalho para outro envolve a coleta de arquivos importantes à mão, uma máquina de cada vez.

12.2.2. Domínio Windows NT

Neste modelo, um ou mais computadores atuam com controladores do domínio para coordenar nomes de usuário e senhas. Uma dessas máquinas servidoras é designada *Primary Domain Controller ou PDC* (Controladora de Domínio Primário), e todas as mudanças são conduzidas por essa máquina. Também pode haver um ou mais *Backup Domain Controllers ou BDCs* (Controladores de Domínio Backup), para prover redundância e compartilhamento de carga.

Os domínios Windows NT usualmente incluem um ou mais servidores de arquivos (que podem ser as mesmas máquinas que estão operando funções PDC ou BDC). Os servidores de arquivo provêm armazenamento para perfis itinerantes (estações de trabalho de usuários, documentos e ambientes) e podem prover também espaço para diretório pessoal, armazenamento de arquivos compartilhados e serviços de fila de impressão.

Em um domínio bem administrado, os usuários normalmente recebem instruções para manter todos os seus arquivos de trabalho no servidor de arquivos, de forma que nenhuma informação importante seja guardada em PCs individuais. A migração de dados a partir de ambientes bem administrados para novos sistemas, é relativamente simples, já que os administradores do sistema sabem onde encontrar todos os arquivos importantes.

12.2.3. Domínio do Active Directory do Windows 2000

O modelo do domínio Windows NT torna-se muito difícil de ser administrado efetivamente para grandes números de usuários, por isso o Windows 2000 introduziu um modelo de domínio hierárquico. É chamado de Active Directory ou AD (Active Directory) e usa idéias do Internet Domain Name System ou DNS e do Lightweight Directory Access Protocol ou LDAP.

Da mesma forma que no Domínio Windows NT, o AD normalmente provê servidores de arquivos para guardarem perfis itinerantes e diretórios domésticos, de forma que seja fácil encontrar arquivos importantes ao planejar o processo de migração.

Pelo fato de AD permitir acesso a LDAP, há mais opções de migração disponíveis para um local que usa AD. Por exemplo, deveria ser possível usar servidores AD para guardar dados de nomes e senhas para software livre e clientes; isso pode ser conveniente onde uma pequena parte de toda a base do usuário for mudar para software livre, já que o processo de gerenciamento do usuário pode permanecer quase sem alterações.

12.3. Visão geral de possíveis rotas de migração

As duas principais rotas consideradas aqui são:

1. Agregue máquinas software livre a domínios Windows existentes e mova os dados e usuários gradualmente para o outro lado, e em seguida remova os servidores proprietários antigos. É possível migrar clientes e servidores independentemente.

Agregar servidores ao domínio Windows é um dos meios mais rápidos de beneficiar-se do software livre. Por exemplo, a combinação do Sistema Operacional GNU/Linux com Samba dá um

Guia Livre – Referência de Migração para Software Livre do Governo Federal

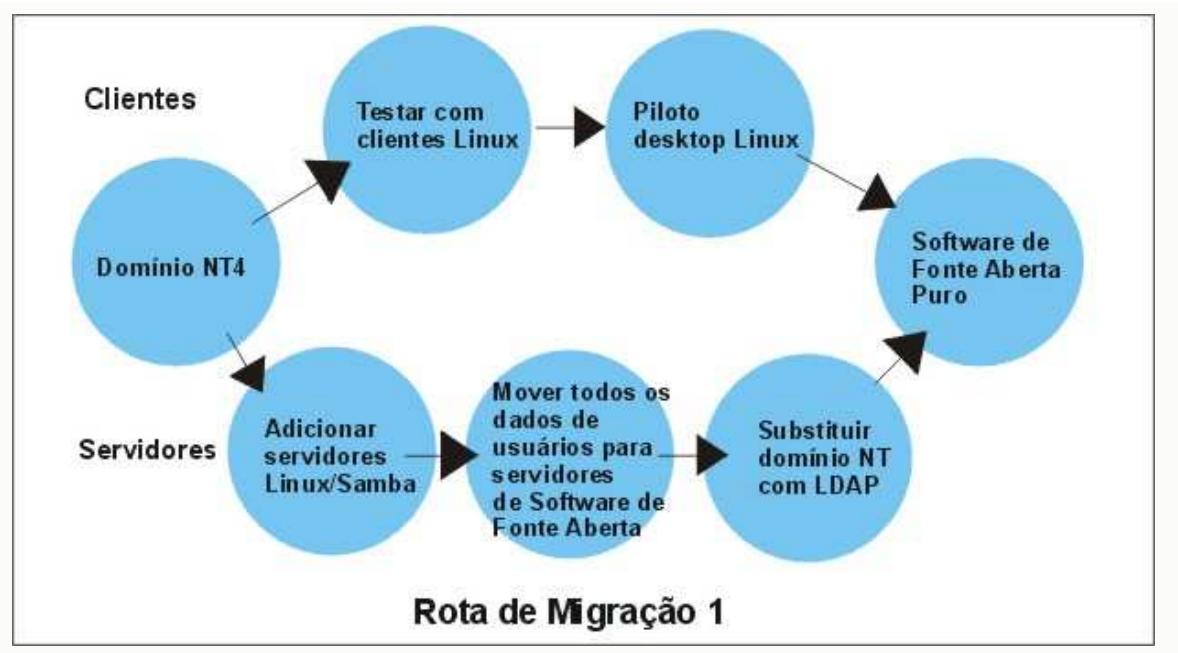
servidor arquivo/impressão poderoso e de baixo custo, que pode ser usado no lugar de um sistema Windows sem quaisquer mudanças no ambiente cliente existente.

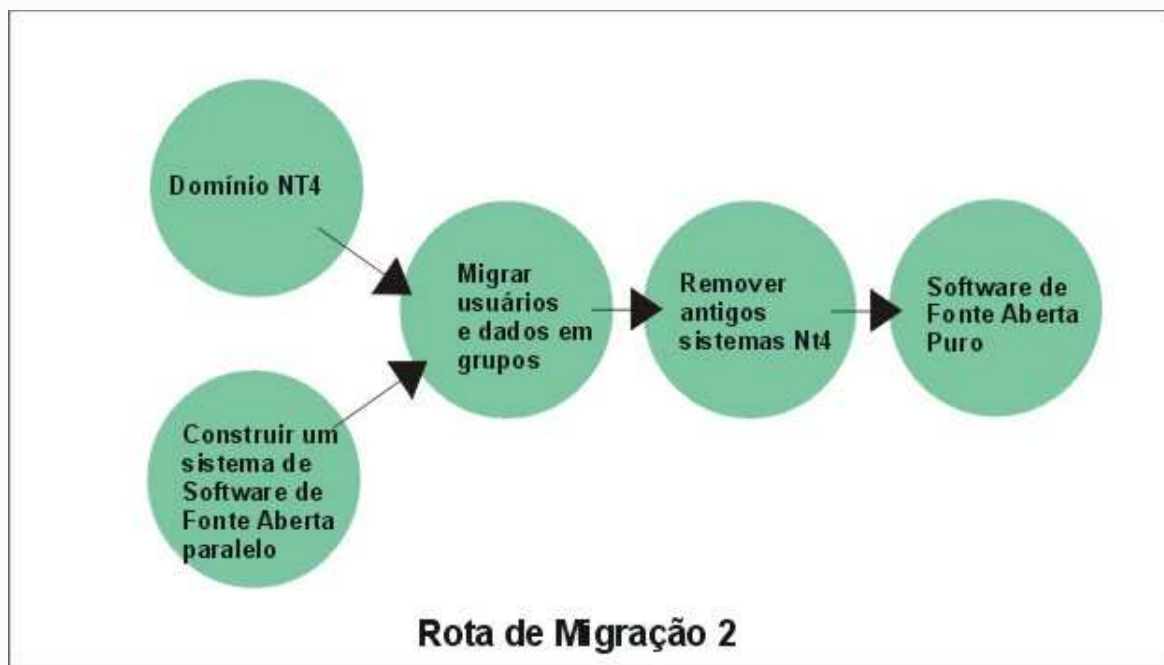
Fazer operar clientes software livre em um domínio Windows é uma forma de coexistência de baixo risco, já que não são necessárias mudanças nos servidores. Pode-se fazê-lo onde um pequeno número de pessoas usa estações de trabalho software livre em um ambiente, ao contrário, somente Windows

2. Construir uma infra-estrutura baseada em software livre paralela e migrar usuários e seus dados em grupos, com mínima interação entre os antigos e novos sistemas. Isso é mais simples do que operar um sistema misto Windows/software livre, mas a cooperação entre as pessoas que usam Windows e as que usam sistemas software livre fica mais difícil.

Ambas as rotas estão resumidas em diagramas abaixo. A primeira rota promove uma integração mais estreita entre os sistemas antigo e novo durante a transição, mas requer um esforço significativamente maior de planejamento e implementação.

Uma dificuldade na escolha da rota será a forma em que Administração está organizada e como isso influi na estrutura lógica e física da instalação do computador.





Os primeiros estágios da maioria das rotas de migração incluem uma fase de coexistência, onde ambos, Windows e sistemas software livre estão em uso, frequentemente acessando os mesmos dados. Estes podem ser modelos particularmente úteis onde está planejada uma migração parcial, com alguns grupos mudando para software livre e outros permanecendo no sistema antigo.

Os detalhes técnicos da operacionalização destas mudanças estão na Seção 12.6 à frente. Porém, discutimos primeiramente o *background* técnico e as ferramentas necessárias.

12.4. Questões Gerais

Há muitas similaridades entre os sistemas proprietários atuais e os sistemas de Fonte Aberta que podem ser escolhidos para substituí-los. Particularmente, as interfaces gráficas de usuário vêm tendendo a convergir para um padrão razoavelmente “cara e jeito”, que reduz problemas para os usuários finais que estão mudando de um sistema para outro. Ainda será necessário treinamento do usuário final, para ajudar as pessoas a lidar com o que é diferente e tirar o melhor proveito do novo sistema.

Atrás da aparência similar dos GUIs, há várias diferenças importantes entre o Windows e os sistemas software livre. Elas são particularmente aparentes no nível da administração do sistema. Aqui é onde será necessária a maior parte de treinamento e planejamento. Os sistemas software livre, como GNU/Linux, possuem GUIs de gerenciamento, mas grandes instalações são gerenciadas mais comumente através das ferramentas da linha de comando, já que lhes fornecem *scripting*, automação de processo, gerenciamento remoto e controle avançado. A habilidade de automatizar tarefas é o que faz do Unix e do sistema software livre ferramentas tão produtivas.

Além das diferenças em processos de gestão, também há algumas importantes diferenças no serviço prestado. Isto deve ser planejado, bem como se deve lidar com isso durante a transição.

12.4.1. Nomes de usuário senhas

Usuários de computadores identificam-se usando nomes de usuário e senhas. Em algumas Administrações, também podem usar cartões inteligentes ou outros dispositivos criptográficos para obter uma prova maior de identificação.

12.4.1.1. Questões de nomes de usuário

Algumas administrações podem ter nomes de usuário “estruturados”, que codificam informação sobre o usuário. Por exemplo, o nome de usuário *cfg27* pode pertencer à 27ª pessoa a ser registrada no Corporate Finance Group. Outras permitem às pessoas escolherem seu próprio nome de usuário, ou simplesmente usarem seu nome real. Esquemas estruturados de nomes de usuário podem ser normalmente usados em sistemas software livre em alteração. Nomes de usuário em software livre não podem começar com um caractere numérico, o que pode causar dificuldades com nomes de usuário estruturados, onde a estrutura inicial pode bem ser numérica.

Há algumas questões que podem afetar os sistemas ad-hoc. Nomes de usuário nos sistemas Windows geralmente preservam a caixa da fonte e são indiferentes à caixa da fonte. Significa que, se alguém receber o nome “Maria”, ela pode digitar “maria”, “MARIA” ou até “mArIa” na hora do *login*, sem problemas. Também significa que, sempre que o sistema exibe um nome de usuário (como o dono de um arquivo), ele vai usar a forma digitada originalmente pelo administrador, quando o nome de usuário foi criado – neste caso, “Maria”.

Por outro lado, nomes de usuário em Unix e software livre são indiferentes à caixa da fonte. O usuário precisa digitar seu nome de usuário exatamente no formato em que foi originalmente registrado. Convencionalmente, nomes de usuário são feitos inteiramente em caixa baixa e números, sem qualquer outro caractere, e com uma largura máxima de oito caracteres.

Essas restrições foram bem amenizadas recentemente, e os sistemas modernos permitirão nomes de usuário mais longos, com um conjunto de caracteres mais amplo.

Alguns esquemas de autenticação e autorização implementam agora nomes de usuário indiferentes à caixa da fonte: o esquema baseado em LDAP proposto neste documento, é tal que, nomes de usuário como “Maria” e “Controle Financeiro” são possíveis. Deve-se tomar cuidado, no entanto, porque pode haver outros pacotes em uso, que baseiam-se em pressupostos das antigas regras. Particularmente, seria muito pouco inteligente deixar espaços ou outros tipos de caracteres de pontuação nos nomes de usuário.

Seria uma boa prática limitar os nomes de usuário aos caracteres permitidos nos nomes de correio, de forma que nomes de usuário possam ser usados como nomes de correio também.

12.4.1.2. Questões de Senhas

Sistemas software livre modernos permitem senhas de quase todos os comprimentos, com um grande conjunto de caracteres. É uma boa prática incentivar o uso de senhas longas (10 ou mais caracteres) com uma boa variedade de letras, números, pontuação, caixa baixa e caixa alta. Os utilitários para estabelecimento de senhas geralmente recusam-se a aceitar senhas muito fracas, a não ser que forçados por um administrador, e muitos locais podem até decidir reforçar regras mais rígidas.

Algumas variantes comerciais Unix ainda “truncam” as senhas para até oito caracteres, de forma que se for planejado um ambiente misto, isso deve ser levado em conta.

A migração de senhas de sistemas proprietários existentes, para novos sistemas software livre nem sempre é possível, já que as senhas são normalmente mantidas de forma criptografada e misturada. O plano de transição poderá ter que incluir a reemissão de senhas para todos os usuários, ou possivelmente uma fase de coleta e sincronização de senhas.

12.4.2. Serviços de autenticação

Qualquer rede, mesmo com um número pequeno de computadores, necessita de um serviço de autenticação e nomeação de rede. No *Windows NT* isso é conhecido como *Domain Controller*. Em sistemas *Windows* posteriores chama-se *Active Directory*. O *Novell NDS* também é largamente instalado, e outros sistemas proprietários possuem seus próprios sistemas de autenticação e nomeação.

A maior parte dos sistemas Unix e software livre podem interagir com quase todos os serviços comuns de autenticação e nomeação. O GNU/Linux é particularmente forte com respeito a isso. O serviço proposto neste documento é baseado em LDAP, mas também é possível usar múltiplos sistemas de autenticação e nomeação ao mesmo tempo, o que pode ser útil durante a fase de transição.

12.4.3. Arquivos

Uma parte muito importante de qualquer plano de transição, concerne à migração de dados do sistema antigo para o novo. Se for planejada uma migração “*big bang*”, então será uma operação exclusiva, porém, se no momento mais provável, for conjeturado um funcionamento paralelo, então será necessário acesso a um arquivo de plataforma cruzada. Deve-se tomar muito cuidado para evitar perda de dados, e para evitar a confusão que pode resultar da separação de cópias modificáveis de um arquivo nos ambientes “antigo” e “novo”.

12.4.3.1. Conteúdo e formato

Este é o assunto mais óbvio de migração, e é tratado em detalhes na Seção 12.8 abaixo. A abordagem normal é usar aplicativos software livre que possam ler os arquivos escritos pelo aplicativo proprietário que estão substituindo, embora em alguns casos seja apropriado planejar uma conversão massiva como parte do processo de migração.

É provável que dados especiais como macros e *scripts* necessitem da atenção de programadores experientes durante a migração.

12.4.3.2. Nomes de arquivos

Como os nomes de usuário, os nomes de arquivos Windows são insensíveis à mudança de caixa de fonte e em alguma extensão preservam a caixa e a fonte. Alguns aplicativos tendem a transformar em maiúscula a primeira letra dos nomes dos arquivos, bem como fazer outras alterações das quais o usuário pode estar ciente ou não. O ambiente Windows também carrega a herança do formato de arquivo DOS 8.3, que ainda aparece em alguns utilitários. Nomes de

Guia Livre – Referência de Migração para Software Livre do Governo Federal

arquivos Windows comumente contêm espaços, e normalmente usam o conjunto de caracteres Unicode. O Windows usa `.\.` como separador de diretório.

Embora seja menos óbvio para usuários GUI, a totalidade dos nomes dos arquivos Windows deve incluir uma letra *.diretório*, indicando o dispositivo físico que contém o arquivo ou eles devem ter o nome verdadeiro do servidor, caso o arquivo esteja em um “diretório rede”. Essas restrições podem ser um problema para gerentes de grandes sistemas Windows, que tentam oferecer um serviço sem emendas ao enfrentar mudanças de *hardware*.

Outros sistemas proprietários tratam nomes de arquivos de formas diferentes. O VMS, por exemplo, tem nomes de arquivos insensíveis à mudança de caixa de fonte que usualmente incluem um ponto, e podem incluir um número de versão após um ponto e vírgula.

Nomes de arquivos em Unix e software livre têm regras diferentes. Aqui, os arquivos são totalmente sensíveis à mudança de caixa de fonte e o sistema não faz quaisquer alterações nos nomes fornecidos pelo usuário. Os nomes usam um conjunto de caracteres 8-bit, determinado pelo uso corrente da localidade. Os únicos caracteres que o GNU/Linux não permite em nomes de arquivos são o separador de diretório (`./.`) e o caractere nulo. No entanto, na prática, não é inteligente incluir caracteres não-imprimíveis, por exemplo, o sistema de arquivos Windows FAT32 não pode armazenar os 32 primeiros códigos ASCII ou qualquer um dos seguintes `"`, `*`, `:`, `<`, `>`, `?`, `\` or `|`. São permitidos espaços nos nomes dos arquivos, embora sua presença requeira que os usuários de linha de comando sejam mais cuidadosos com aspas.

Sistemas Unix e software livre não usam letras de diretório e não requerem que o nome real do servidor de arquivo faça parte do nome absoluto do arquivo, onde o acesso ao arquivo de rede é usado. Ao invés disso, o sistema apresenta todos os arquivos como parte de uma hierarquia sem remendos. Junto com o uso de *links* simbólicos no sistema de arquivo e *data-driven automounters*, isso dá aos administradores do sistema grande flexibilidade na separação do nome absoluto de um arquivo de seu lugar de armazenamento físico.

Quase todos os nomes de arquivos Windows podem ser migrados diretamente para os servidores software livre sem alterações. É possível que se encontre na prática a única exceção nos nomes de arquivos que contêm o caractere `./.`, que terá que ser modificado durante a transição. Usuários de ferramentas GUI provavelmente nunca perceberão que os nomes dos arquivos tornaram-se sensíveis à mudança de caixa de fonte, já que só digitam tais nomes quando criam o arquivo.

12.4.3.3. Acesso dual

Muitos planos de migração provavelmente incluirão um período de funcionamento paralelo onde algumas pessoas usam sistemas software livre e outras ainda usam sistemas paralelos. Onde os arquivos forem acessados por membros dos dois grupos, pode-se precisar tomar algumas providências.

O compartilhamento de arquivos em sistemas Windows usa o protocolo SMB (*Server Message Block*), que é uma tecnologia muito complexa com muitos níveis de compatibilidade para trás. É usado por servidores de arquivos consagrados e também no modo “par-a-par”, onde PCs individuais disponibilizam partes de seus sistemas na rede. Ambientes bem gerenciados da Administração irão basear-se mais provavelmente em servidores consagrados do que em compartilhamento *ad-hoc*. Arquivos de usuários não compartilhados em um ambiente Windows podem ser mantidos em vários lugares diferentes:

Guia Livre – Referência de Migração para Software Livre do Governo Federal

1. Em um disco local do *PC* estação de trabalho ou *laptop* do usuário, por exemplo, o referido como diretório C.

2. No perfil *roving/roaming* do usuário. Isto inclui a maior parte dos conjuntos de preferência e também o conteúdo da estação de trabalho Windows e (normalmente) a pasta “Meus Documentos”. O perfil *roving* é mantido localmente em qualquer PC que o usuário estiver utilizando, e é sincronizado de volta para um depósito de perfis na hora do *logout*. Isso oferece a facilidade de *backup* acessível, porém pode ter sérias implicações de desempenho, com usuários relatando *logouts* muito lentos.

3. Em um diretório doméstico de um servidor de arquivo gerenciado de forma central. Essa é uma opção comum em grandes redes de sistemas de estações de trabalho, por ser fácil de manejar *backups* apropriadamente. Não é sensato tentar prover acesso dual a arquivos mantidos em estações de trabalho individuais ou servidores de arquivos *laptops* em um estágio inicial do processo de migração.

O principal mecanismo de acesso ao arquivo de rede do Unix e dos sistemas software livre é o *Network File System (NFS)*. Este é um protocolo mais simples do que o *SMB*, e sua especificação sempre esteve disponível.

As opções para implementação do acesso dual pertencem a duas amplas categorias: adicionar suporte de protocolo dual a servidores, ou adicionar suporte de protocolo dual a clientes. Além de tudo que é igual, é normalmente mais fácil mudar servidores do que clientes, quase sempre mais fácil ajustar sistemas software livre do que sistemas proprietários. As opções estão resumidas na tabela a seguir.

	Servidores Windows	Servidores de Software livre e Unix
Clientes Windows	Acesso a arquivos SMB é padrão	Servidores suportam SMB usando o pacote Samba. É um software maduro com um desempenho excelente.
Clientes de Software livre	Clientes GNU/Linux podem acessar cotas SMB. Isto fica bastante fácil quando as máquinas cliente só tem um usuário por vez, mas torna-se mais envolvente com máquinas compartilhadas. Variantes comerciais Unix não tem normalmente capacidade de cliente SMB. É possível adicionar serviço NFS a servidores Windows, mas isso pode se tornar muito dispendioso.	Acesso a arquivos NFS é padrão. Se assim o desejarem, os clientes GNU/Linux podem usar SMB como parte de um plano de migração, mas é menos eficiente.

12.5. Ferramentas

Esta seção discute alguns dos componentes-chave do software livre, os quais serão usados na migração de sistemas proprietários.

12.5.1. Samba

Samba é um pacote de arquivo e servidor de impressão dos sistemas software livre. Ele implementa o protocolo SMB da *Microsoft* e, em muitos casos, pode substituir completamente as funções de um servidor Windows. O Samba também pode atuar como um Windows NT Domain Controller e é capaz de armazenar dados de gerência de domínio em um diretório acessado através de LDAP.

O Samba fornece ferramentas em seu pacote cliente que são muito úteis para diagnosticar problemas com redes SMB, apóiam na migração de arquivos do legado e servidores Windows, e ainda, são apropriadas para criptografar.

O Samba é mantido por um grupo central de mais ou menos 30 voluntários muito ativos por todo o mundo. Mais informações podem ser encontradas em <http://www.samba.org/>.

12.5.2. OpenLDAP

O OpenLDAP é uma implementação do *Lightweight Directory Access Protocol* (LDAP). Inclui um servidor de diretório, um conjunto de acesso a dados e ferramentas de gerenciamento, e um conjunto de bibliotecas para dar suporte ao LDAP em outros aplicativos.

O OpenLDAP é mantido por um pequeno grupo central mais um grande número de contribuintes. Uma pessoa do grupo central trabalha no projeto todo o tempo.

12.5.3. NSS e PAM

O NSS é o Name Service Switch: uma tecnologia usada pelo GNU/Linux e por algumas variantes do Unix para permitir que diferentes serviços de pesquisa de nomes sejam usados na procura de hostnames, nomes de usuário, nomes de grupo etc. Existem diversos módulos disponíveis, dos quais, os mais relevantes são:

1. Arquivos : é possível permitir consultas simples de hostnames, usuários, grupos de usuários baseadas em arquivos de texto locais
2. DNS: Consultas de hostname baseadas em Domain Name System (DNS)
3. LDAP: Consultas baseadas em LDAP – em sua maior parte nomes de usuário e grupos de usuários, mas também útil para muitos outros propósitos.
4. SMB: Consultas usando protocolo Windows SMB (veja 12.5.5 – Winbind)
5. NIS: Consultas de usuários e grupos de usuários em bases de dados Network Information System (NIS), muito utilizadas em ambientes Unix.

O arquivo de configuração do NSS normalmente é feita através do arquivo “/etc/nsswitch.conf”.

PAM (Pluggable Authentication Module) é um sistema livre de autenticação modular utilizado pelo GNU/Linux e por diversos outros sistemas Unix-Like. Por ser um sistema modular de autenticação ele permite grande flexibilidade na configuração do processo de autenticação e autorização dos usuários. Alguns módulos interessantes:

1. LDAP: Usa operações LDAP para verificar as credenciais do usuário. (O Active Directory é um protocolo baseado no LDAP, logo é possível autenticar usuários em domínios Active Directory).
2. SMB: Permite verificar as credenciais de um usuário em um domínio WinNT.
3. Access: Acesso restrito aos serviços da rede.
4. Cracklib: torna obrigatória a checagem de qualidade das senhas dos usuários. (inibe senhas do tipo lala123, 123456 etc..)
5. Smartcard: permite fazer a autenticação de um usuário com um smartcard.

Um howto inicial pode ser encontrado em <http://www.faqs.org/docs/Linux-HOWTO/User-Authentication-HOWTO.html>, documentação online pode ser encontrada em <http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/>, alguns módulos podem ser encontrados em <http://www.kernel.org/pub/linux/libs/pam/modules.html>

12.5.4. Acesso a arquivo GNU/Linux SMBFS

O Samba permite que um sistema software livre forneça serviço de arquivo a clientes Windows. O SMBFS trabalha ao contrário: permite que um sistema software livre acesse arquivos mantidos em servidores Windows. O SMBFS é disponibilizado com as mais recentes distribuições GNU/Linux, porém não é encontrado normalmente em sistemas Unix comerciais.

O modelo acesso-controle usado pelos sistemas de arquivos Windows é diferente do que é usado por GNU/Linux e outros sistemas software livre, portanto, há algumas limitações com relação ao que se pode conseguir com o SMBFS.

12.5.5. Winbind

Outro produto do time Samba, o Winbind permite que máquinas GNU/Linux individuais sejam anexadas ao domínio Windows NT. Mantém um mapeamento entre autenticadores Windows NT (SIDs) e UIDs e GIDs Unix-style. O Winbind pode fazer muitas outras coisas que reduzem a carga nos administradores de sistemas, como montar ambientes Unix-style para as pessoas que estiverem conectando-se (*login*) pela primeira vez.

A desvantagem do Winbind em grandes redes é que cada cliente de computador constrói seu próprio mapeamento entre os autenticadores Windows e os Unix. Isso pode causar problemas em estágios posteriores da migração, quando os servidores de arquivos software livre forem introduzidos.

Ao usar o Winbind, nomes de usuário e nomes de grupo usados por GNU/Linux são formados concatenando o nome do domínio Windows NT com o nome de usuário Windows NT, para formar uma única série. Isso pode levar a uma certa confusão, já que muitos utilitários Unix-style só fornecem espaço na sua produção para nomes de usuário de oito caracteres. O maior nome gerado pelo Winbid aparece truncado na tela.

12.6. Migrando o ambiente do sistema operacional

12.6.1. Acrescentar servidores GNU/Linux individuais a um domínio Windows NT existente

A configuração é extremamente simples:

1. Instale um servidor GNU/Linux, dando-lhe um endereço IP fixo.
2. Assegure-se de que os pacotes *Samba* estejam instalados. Serão necessários o *samba* original, o *common-samba* e o *client-samba*. Eles estão normalmente incluídos em uma instalação “servidor”.
3. Edite */etc/samba/smb.conf*, configure **domínio** modo segurança, e defina o nome do domínio (grupo de trabalho). Liste o PDC e quaisquer BDCs como servidores de senha. Defina as partes a serem servidas pela máquina.
4. Crie quaisquer diretórios que devam ser compartilhados, e configure as propriedades e permissões associadas.

5. Junte a máquina ao domínio Windows NT existente, usando a senha do Domínio Administrador (ou qualquer outro nome de usuário e senha que tenha o poder de fazer isso):
`smbpasswd -j DOMAINNAME -r PDCNAME -U Administrator`

6. Inicie o samba e configure para que ele reinicie quando o computador for religado:
`/etc/init.d/smb start`
`chkconfig smb on`

O servidor agora aparecerá em listas de *navegadores* e poderá ser usado como um servidor Windows NT.

12.6.2. Executar estações de trabalho GNU/Linux em domínios Windows NT

12.6.2.1. Configuração simples para pequeno número de máquinas

Nos primeiros estágios de teste das ferramentas software livre, é muito útil operar máquinas individuais GNU/Linux com configurações bem simples. Elas podem receber acesso a arquivos no servidor Windows para compatibilidade e testes de migração usando o comando **smbmount**.

Montar é o termo Unix/software livre *para* fazer de um disco ou sistema de arquivo remoto parte da hierarquia de arquivo da máquina local. O processo é normalmente feito de forma automática na hora do *boot*, sob o controle do arquivo

`/etc/fstab`, porém também pode ser feito de forma interativa. Por exemplo, o comando para trazer um CD-ROM ISO-standard para dentro do sistema de arquivo, sob o controle do sistema de arquivo `/mnt/cdrom` seria:

```
mount /dev/cdrom /mnt/cdrom.
```

O comando **mount** tem seu uso normalmente restrito ao usuário *raiz* por razões de segurança. Este não é um problema onde a máquina esteja sendo usada por um administrador do sistema, mas pode ser um desastre quando está envolvido um usuário não-técnico. O GNU/Linux oferece algumas saídas para este problema:

1. Use uma entrada especial em `/etc/fstab` que permita a usuários comuns montar certos objetos pré-definidos. Esse é o caminho usual para possibilitar que cdrom e disquetes sejam montados quando solicitados. Os arquivos montados normalmente aparecem como propriedade de quem quer que tenha montado o dispositivo.

2. Use um programa **setuid-root** para a operação privilegiada, checando antes se é segura. É a forma mais fácil de lidar com a montagem de partes remotas de Windows.

3. Use um **automounter** para montar sistemas de arquivos quando forem acessados pela primeira vez e para desmontá-los quando não estejam mais em uso. O *automounter* opera como um *daemon* e é normalmente dirigido por dados de configuração de rede. Isso demanda um esforço maior de configuração do que os outros métodos, porém é extremamente útil em grandes redes. Neste esquema, usaremos os comandos **smbmount** e **smbumount** para fazer uma parte de um Windows existente parecer parte do sistema de arquivo GNU/Linux. Nos sistemas Red Hat Linux eles são parte do pacote *samba-client*, portanto, assegure-se de que você instalou ambos os pacotes, *samba-common* e *samba-client*. Esses programas são desenhados de tal forma que as partes críticas podem receber alguns privilégios de *raiz*, embora não sejam instalados dessa forma por *default*,

Guia Livre – Referência de Migração para Software Livre do Governo Federal

então alguns poucos comandos devem ser operados por *raiz* antes de serem usados pela primeira vez:

```
chmod u+s /usr/bin/smbmnt /usr/bin/smbumount
```

Note que o comando muda **smbmnt** ao invés de **smbmount**. Isso é importante porque **smbmnt** encapsula somente as funções de **smbmount** que requerem privilégios de raiz. Isto feito, qualquer usuário pode usar o **smbmount** e o **sbumount** e eles funcionarão com os privilégios de raiz.

Agora qualquer usuário pode disponibilizar o Windows SMB como parte do sistema de arquivo GNU/Linux, montando-o em um diretório que já possui. Quaisquer arquivos que já estejam no diretório serão invisíveis enquanto o SMB estiver montado em cima.

Como exemplo, suponha que o usuário do GNU/Linux *Fred* deseja acessar arquivos em um servidor Windows NT chamado NT4SERVER no domínio THESTATE, que são compartilhados sob o nome FREDERICK e propriedade do usuário de Windows FREDERICK. *fred* começa fazendo um novo diretório para montar a parte Windows em:

```
mkdir ~/ntfiles
```

(A notação “~/” significa “no meu diretório pessoal”). Isso só precisa ser feito uma vez. Agora, para montar a parte remota:

```
smbmount //nt4server/frederick ~/ntfiles \  
-o username=frederick -o workgroup=thestate
```

O comando deve ser digitado em uma linha, ou partido com caracteres de continuação de linha “\” como mostrado aqui. Ele estará pronto para a senha de FREDERICK no servidor, e então monte a parte do Windows no alto do diretório *ntfiles* no diretório pessoal de *fred*. Para evitar tudo isso de novo a cada *login*, pode ser colocado em um arquivo *script* ou mesmo fazer parte do processo de *login* de *fred*.

Agora a parte montada comporta-se como se fosse parte do disco local. Pode-se criar, apagar e editar arquivos. Porém, há alguns riscos. Particularmente, não há tentativas de organizar o controle de acesso ao estilo Unix e os comandos Windows NT ACL para mudar a propriedade, ou modos de arquivos e diretórios na parte montada não surtirão efeito.

Antes de sair, é interessante desmontar a partilha:

```
sbumount ~/ntfiles
```

Novamente, isto pode tornar-se parte automática do processo de *logout*, caso desejado. O processo descrito nesta seção não cria qualquer ligação permanente entre contas no GNU/Linux e contas nos servidores Windows NT existentes, portanto usuários e senhas devem ser mantidos separadamente em cada máquina. O esforço de gerenciamento envolvido pode tornar-se excessivo rapidamente à proporção em que o número de máquinas vá crescendo, portanto este esquema só é realmente apropriado para ambientes de teste pequenos.

12.6.2.2. Configuração mais inteligente para distribuições mais amplas

Onde for necessária uma distribuição piloto mais ampla dos sistemas de estações de trabalho software livre, pode ser conveniente manter ainda serviços de arquivos e autenticação nos servidores Windows NT existentes. O *daemon* do *Winbind* do *Samba* fornece um modo fácil de ligar os dois ambientes. Tanto o Samba como o Winbind são partes padrão da distribuição Red Hat Linux, porém podem não estar instalados por *default* em configurações de estações de trabalho. Para usar o Winbind, devem ser instalados os seguintes pacotes: *samba*, *samba-common* e *samba-client*.

O arquivo `/etc/samba/smb.conf` deve ser editado para mostrar o nome correto do domínio Windows NT na linha do **workgroup**, e para colocar o sistema no modo de segurança **domain**. Os dados de segurança do Winbind também estão na seção global deste arquivo, por exemplo:

```
# separate domain and username with '+', like DOMAIN+username
winbind separator = +
# use uids from 10000 to 20000 for domain users
winbind uid = 10000-20000
# use gids from 10000 to 20000 for domain groups
winbind gid = 10000-20000
# allow enumeration of winbind users and groups
winbind enum users = yes
winbind enum groups = yes
# give winbind users a home directory location
template homedir = /home/winnt/%D/%U
# and a shell
template shell = /bin/bash
```

Para que o Winbind trabalhe, certos serviços devem estar funcionando. Para iniciá-los e para garantir que iniciem a cada *reboot*, os comandos são:

```
chkconfig smb on
chkconfig winbind on
/etc/init.d/smb start
/etc/init.d/winbind start
```

A máquina agora deve juntar-se ao domínio Windows NT. Isto requer usuário e senha Windows NT, com as permissões apropriadas (usualmente *Administrador*):

```
smbpasswd -j DOMAINNAME -r PDCNAME -U Administrator
```

Agora deverá ser possível conseguir listas de usuários Windows e grupo com o comando *wbinfo*:

```
wbinfo -u
wbinfo -g
```

Para que os dados do Winbind estejam disponibilizados no sistema, é necessário editar arquivos de configuração PAM e NSS. Isso deve ser feito com muito cuidado, porque você pode ficar bloqueado fora do sistema se esses arquivos forem avariados.

Em `/etc/nsswitch.conf` acrescente a palavra **winbind** à **passwd** e às linhas de **group**.

Em `/etc/pam.d/system-auth` acrescente uma linha da fórmula:

```
auth sufficient /lib/security/pam_winbind.so use_first_pass
```

logo após à linha **auth** equivalente que usa **pam_unix**, e uma da fórmula:

```
password sufficient /lib/security/pam_winbind.so use_first_pass
```

logo após a linha **password** equivalente que usa **pam_unix**.

Será necessário reiniciar o *Name Service Cache Daemon* neste estágio:

```
/etc/init.d/nscd restart
```

A tradução de usuários e grupos Windows para formato de arquivo de senha Unix-style, pode ser vista agora com:

```
getent passwd
```

```
getent group
```

Para automatizar a criação de diretórios pessoais de usuário no primeiro *login*, adicione isto à parte **session** de `/etc/pam.d/system-auth`:

```
session required /lib/security/pam_mkhomedir.so skel=/etc/skel/  
umask=0022
```

(Garanta que seja inserido como uma única linha ao invés de duas linhas, como o que está apresentado aqui). Note que isto irá criar um diretório pessoal Unix separado para o usuário em cada estação de trabalho que usarem. Também pode ser útil para inserir um *script* no diretório `/etc/skel` para que cada usuário tenha que montar seus arquivos Windows NT em um local padrão na hora do *login*.

12.6.3. Executar estações de trabalho GNU/Linux em domínios Active Directory

Em princípio, máquinas estações de trabalho GNU/Linux podem juntar-se ao domínio AD (*Active Directory*) quase que da mesma forma que se juntam ao domínio Windows NT. Na verdade, se o domínio AD estiver funcionando em modo compatibilidade –NT, então exatamente o mesmo processo pode ser usado.

O domínio AD também oferece a possibilidade de usar o LDAP para autenticação e consulta de dados. Este é o mesmo esquema proposto para redes maiores de sistemas software livre puro, e vale bem a pena considerar. Ao estender o esquema AD para incluir dados Unix, será possível gerenciar os usuários e servidores de estações de trabalho software livre, com ferramentas de administração AD. Para o esquema Winbind usado com o Windows NT, é preferível armazenar os dados de forma centralizada, já que ele mantém a organização entre as IDs Windows NT e as Ids Unix consistentes entre todas as máquinas.

12.6.4. Substituir o Windows NT PDC/BDC por Samba+LDAP

O *Samba* pode cumprir o papel de *Primary Domain Controller*, permitindo assim, que todos os servidores Windows sejam eliminados, até mesmo se ainda forem necessários clientes Windows. Observe que não é possível substituir somente o PDC ou somente um BDC em um domínio: todos os controladores de domínios devem estar operando o mesmo sistema, seja ele Windows ou Samba. Isso é porque, em parte, o protocolo de réplica do PDC não foi submetido a uma engenharia reversa. Além disso, os *Samba Domain Controllers* tem uma abordagem diferente para resiliência: eles o delegam para os servidores LDAP, onde os dados são realmente armazenados.

Instalar um *Samba+LDAP Domain Controller* é um trabalho muito grande para ser descrito aqui, mas pode ser feito em um dia mais ou menos, por uma pessoa experiente. A tarefa maior é planejar a migração dos nomes de usuário e nomes de grupo de um domínio existente. Uma parte do trabalho é coberta pelo *Samba-LDAP-HOWTO* do IDEALX (veja referências na Seção 12.12 abaixo). A mesma fonte fornece um conjunto de estruturas de ferramentas de migração, que podem ser uma boa base para se construir.

Em resumo, o processo é:

1. Instale o(s) servidor(es) software livre com *Samba e OpenLDAP*. Pode ser necessário construir o Samba a partir da fonte, por exemplo, o Red Hat Linux 7.3 não incluía a versão com LDAP disponibilizada.
2. Adicione as definições do esquema Samba ao servidor LDAP.
3. Instale o servidor LDAP com uma base apropriada Distinguished Name (DN) e uma estrutura de árvore de diretório (possivelmente usando as ferramentas do IDEALX para popular a árvore com entradas *boilerplate*).
4. Inicie o Samba e teste a função Domain Controller.
5. Use o **pwdump** no PDC para listar todas as entradas de usuários no SAM. Transfira o resultado como arquivo texto para o servidor software livre.
6. Configure o ferramenta IDEALX **smbldap-migrate-accounts.pl** para igualar com o ambiente em construção. Isso não é uma coisa simples, pois há várias opções a considerar.
7. Faça operar o **smbldap-migrate-accounts.pl** nos dados transferidos do PDC. Isso irá criar entradas no LDAP para todos os usuários do domínio. Também irá instalar suas senhas SMB para igualarem-se às senhas usadas no Windows NT (porém isso não permitirá *logins* Unix ou GNU/Linux, já que as senhas do Windows NT são embaralhadas e o esquema dos sistemas software livre é diferente). A ferramenta pode criar diretórios pessoais ao mesmo tempo, caso desejado.
8. Copie arquivos de usuários e perfis *roving* dos servidores Windows para os servidores software livre, ou religue os servidores Windows existentes aos domínios agora servidos por Samba Domain Controllers.

Grandes redes provavelmente necessitarão múltiplos servidores LDAP com réplica de dados para resiliência. Se um Samba Domain Controller estiver associado a casa servidor LDAP, pode ser realizado um esquema muito parecido com a instalação do Windows PDC/BDC.

Há muitas outras questões a serem consideradas, tais como:

1. Escolha das ferramentas para gerenciamento do usuário.
2. Como os grupos Windows NT e ACL serão organizados nos grupos Unix-style e ACL.
3. Usar ou não um novo nome de domínio para o serviço baseado em software livre.
4. Como criar misturas de senhas utilizáveis pelos sistemas software livre (ou continuar a usar as misturas Windows NT ou as LANMAN, até em um ambiente puramente software livre).

12.6.5. Substituir o Active Directory Windows 2000 por LDAP

O volume de dados mantido em um Active Directory fica em um depósito acessível LDAP.

À primeira vista, isso deveria facilitar a substituição dos servidores AD por equivalentes software livre. Infelizmente este não é o caso: os sistemas Windows 2000 não usam LDAP puro para todos os acessos a dados, e sim uma variante não padronizada do Kerberos para autenticação.

Muitas equipes de trabalho software livre estão trabalhando para consertar esse problema, porém, até o momento em que este trabalho foi escrito, a única forma possível de dar suporte aos clientes Windows 2000 e Windows XP, é fazê-los operar no domínio Windows NT, como descrito acima.

12.6.6. Executar uma infraestrutura GNU/Linux paralela e migrar usuários em grupos.

12.6.6.1. Substituindo todos os clientes Windows por GNU/Linux

Este é o mais simples de todos os esquemas de migração possíveis. A interação entre os sistemas Windows e software livre é limitada à transferência exclusiva de arquivos de usuários. Em linhas gerais, o processo é:

1. Construa o núcleo do ambiente software livre. Isto inclui servidores LDAP para manter a configuração e os dados do nome de usuário, servidores de instalação *master*, um ou mais servidores de arquivo e de impressão, e estações de trabalho suficientes para o pessoal de gerenciamento de sistemas.
1. Construa uma instalação para desenvolvimento e treinamento, com estações de trabalho suficientes para permitir o treinamento de grupos de pessoas de tamanhos apropriados. A tarefa inicial dessa instalação é validar e sintonizar a configuração antes da primeira apresentação do trabalho.

Nesse estágio, o processo de construção da estação de trabalho deve terminar de forma que as máquinas estejam configuradas com mínimo esforço humano. É muito importante que todas as

Guia Livre – Referência de Migração para Software Livre do Governo Federal

máquinas estejam configuradas *exatamente* da mesma forma durante a primeira fase de apresentação do trabalho, portanto, isso deve ser testado cuidadosamente.

2. Use a instalação de desenvolvimento e treinamento consultando os representantes da base de usuários para gerar entusiasmo pelo projeto e para reunir *feedback* sobre a interface do usuário. Faça mudanças de acordo com as necessidades, de forma a chegar à imagem difundida. Chegue a um acordo sobre os requisitos para o treinamento e a agenda.
3. Construa um conjunto de novas estações de trabalho suficiente para substituir o equipamento que está sendo usado pelo primeiro grupo a migrar para sistemas software livre.
4. Registre o primeiro grupo de usuários no sistema novo.
5. Treine o primeiro grupo de usuários no sistema novo.
6. Caso necessário, reconfigure quaisquer configurações alteradas durante o treinamento de forma que todos comecem com um ambiente novo.
7. Substitua o primeiro grupo de estações de trabalho *PC* pelos sistemas software livre pré-construídos. Ao mesmo tempo, copie os arquivos dos grupos para os novos servidores de arquivos e configure a cópia original para “somente para leitura”.
8. Ofereça suporte ativo ao primeiro grupo enquanto eles se acostumam a usar o sistema software livre.
9. Faça a atualização dos PCs removidos do primeiro grupo de acordo com a necessidade, e instale a imagem de estação de trabalho padrão.
10. Repita o procedimento a partir do item 5 com o próximo grupo de usuários.
11. Quando todos os usuários tiverem migrado para os sistemas software livre, faça cópias de todos os arquivos dos velhos servidores e tire-os de circulação.

12.6.6.2. Mantendo alguns clientes Windows

Onde for necessário manter alguns clientes Windows (por exemplo, para dar suporte a algumas funções cuja migração não é economicamente indicada devido a *softwares* não transferíveis), há duas opções principais:

1. Retenha um pequeno domínio Windows usando um ou mais servidores Windows.
2. Dê suporte aos clientes Windows de servidores baseados em software livre usando o Samba.

O caminho escolhido dependerá do motivo pelo qual os clientes estão sendo retidos, e de sua distribuição geográfica.

Em qualquer caso, é provável que o Samba será necessário em um ou mais dos novos servidores, para fornecer compartilhamento de arquivos entre clientes Windows e clientes com base software livre.

12.7. Migrando aplicativos tipo servidor

12.7.1. Servidores da Web: mudando do IIS para o Apache

O servidor da *web* usual do Windows é o IIS (Internet Information Server), que fornece serviços HTTP, FTP, e Gopher em um pacote. O IIS é conhecido por apresentar problemas com segurança e estabilidade, o que fez muitas organizações substituí-lo por um servidor alternativo. Na verdade, após a explosão em 2001 de vários defeitos particularmente sérios, analistas de Gartner publicaram um estudo com fortes recomendações a seus clientes sugerindo que o IIS não fosse utilizado para funções críticas, até que fosse completamente reescrito pela *Microsoft*.

Há vários servidores de rede à escolha para substituir o IIS. Muitos são softwares livres ou possuem condições de licença muito liberais. Alguns dos servidores usados mais amplamente estão comentados na Seção 11.4.2 acima.

Ao migrar *sítios* do IIS, a opção usual é o Apache, freqüentemente com PHP ou módulos Perl para criptografar. O Apache trabalha em GNU/Linux, FreeBSD, quase todas as outras variantes Unix, e também no Windows. Isso oferece uma ampla gama de opções de migração.

12.7.1.1. Questões de Migração

1. Nomes de Arquivos e URLs

Ao mudar um simples *sítio* do IIS no Windows para o Apache no GNU/Linux ou Unix, a principal questão à qual deve-se estar atento é que o sistema de arquivo do Windows ignora “caixa alta/caixa baixa” em nomes de arquivos, porém a maior parte dos sistemas de arquivos GNU/Linux ou Unix são sensíveis às “caixa alta/caixa baixa”. Como a hierarquia das páginas da *web* é normalmente representada diretamente no sistema de arquivo, significa que URLs tornam-se sensíveis no ambiente Unix ou GNU/Linux. (Isso não seria uma questão se o Apache fosse usado em um servidor Windows).

Uma questão menos comum é que parece que o IIS aceita tanto “\” quanto “/” como componentes separadores. Traduzido para o sistema de arquivo Windows, “/” e “\”, mas parece permitir que “\” trabalhe originalmente. Então, pode-se referir a um arquivo em URL tanto como `mydir\thisfile.html` quanto `mydir/thisfile.html`.

Nenhuma questão afeta um *sítio* escrito corretamente e consistente. Infelizmente, *sítios* escritos com *software* Windows freqüentemente têm uso inconsistente de caixa alta e caixa baixa, e algumas vezes, tem o caractere “\” em URLs onde a estrutura de arquivo do *sítio* contém um subdiretório. Na verdade, o *sítio* exemplo distribuído com as versões mais recentes do IIS apresenta ambas as questões. Há soluções paliativas fáceis para os dois problemas no Apache, as quais são demonstradas em exemplo no decorrer do capítulo. Porém, como regra geral, é melhor corrigir tais problemas nos dados do *sítio*.

2. Mapas de imagem tipo servidor

Alguns *sítios* mais antigos usavam um mapeamento tipo servidor de coordenadas x, y em uma imagem destinada a URLs. Isso agora é condenado porque é ineficiente e não trabalha bem com *navegadores* não-GUI, mas alguns *sítios* ainda podem usá-lo. Os mapas tipo servidor no IIS tomam a forma de arquivos com extensão “map”, e seu formato não é compatível com os arquivos Apache equivalentes.

A melhor abordagem é converter qualquer mapa tipo servidor em mapas tipo cliente, já que isso fornece uma experiência de *browsing* melhor para o usuário. Se isso não é possível, pode-se usar um *script Perl* simples para editar arquivos para uma forma usável pelo Apache.

3. Scripts e conexões de bancos de dados

Sítios mais complexos tendem a ter páginas dinâmicas baseadas em *scripting* e acesso a banco de dados. A maior parte dos *sítios* IIS usam ASP (Active Server Pages) como estrutura de *scripting* e podem usar o Access ou o SQL Server para banco de dados, dependendo do tamanho do aplicativo.

Há muitas formas de lidar com a migração de *scripts* ASP. Alguns dos mais populares são:

1. Pacote Chili!Soft ASP para Unix (agora chamado Sun ONE Active Server Pages)
2. ASP2PHP
3. O Apache do Apache::módulo ASP
4. Conversão manual para uma nova linguagem

O Chili!Soft ASP é um produto proprietário, mas em alguns casos, pode oferecer uma rota de migração muito econômica

O ASP2PHP é um conversor de *script* independente que converte arquivos de texto escritos em ASP e VBScript em arquivos de texto escritos em PHP. Está sendo desenvolvido um suporte para arquivos ASP usando JScript. O PHP é uma estrutura de *web-scripting* muito popular com características similares ao ASP, e pode ser, portanto, uma forma de transição bem fácil para os desenvolvedores. Para projetos maiores, é sempre bom fazer uma separação maior entre desenho de página e lógica do *script* do que os modelos ASP ou PHP permitem. Nesses casos, uma conversão manual usando um sistema baseado em modelo pode ser uma opção melhor.

O Apache::ASP fornece características ASP diretamente através da estrutura Apache, junto com *scripting* em Perl. O VBScript e o JScript não tem suporte.

Em alguns casos, pode ser melhor considerar uma conversão manual de ASP para uma nova estrutura. Isso garante maior flexibilidade e *sítios* complexos podem beneficiar-se bastante na mudança para um sistema baseado em modelo como o Template Toolkit (<http://www.tt2.org/>).

Todos os sistemas de *scripting* Apache possuem facilidades de acesso a bancos de dados para uma ampla faixa de tipos de bancos de dados (*SQL*, *flat file*, *indexed*, *LDAP*, *NIS* etc), portanto, podem ser construídos *sítios* dinâmicos de qualquer complexidade, dirigidos para bancos de dados.

4. Extensões FrontPage

O pacote de *web-design* do FrontPage introduziu um conjunto de extensões que permitem gerenciamento remoto de conteúdo da *web*. Tem sido usado desde então por outros pacotes de *web-design*.

As extensões FrontPage são disponíveis para sistemas Unix, porém não universalmente populares com administradores Apache por várias razões, incluindo questões de segurança e o grande número de mudanças introduzidas na área de armazenamento da página *web* padrão. Há disponível agora uma substituição baseada em padrões no formato protocolo *WebDAV* (RFC2518).

Recebe suporte da maior parte dos servidores de *web* (inclusive Apache, usando o módulo **mod_dav**) e é o protocolo de gerenciamento de *sítio* preferido atualmente. Alguns fornecedores de software proprietários fornecem suporte ao WebDAV. Portanto, um servidor Linux/Unix/Apache pode dar suporte tanto a clientes software livre quanto a clientes proprietários usando o mesmo mecanismo.

12.7.1.2. Migrando um *sítio* Internet estático

Este exemplo mostra um processo completo para migrar um *sítio* simples e estático do IIS no Windows NT para o Apache no GNU/Linux.

1. Prepare o servidor GNU/Linux, conecte-o à rede e teste o Apache. A maior parte das distribuições GNU/Linux fornecem pacotes Apache pré-configurados, portanto, isso é normalmente automático. Um servidor visível na Internet vai precisar, sem dúvida, de fortalecer sua segurança antes de conectar-se.

2. Localize os dados do *sítio* no servidor IIS (usualmente em **C:\InetPub**) e faça uma cópia pronta para transferência, por exemplo, usando um pacote de arquivo Zip.

3. Copie o arquivo Zip para a máquina GNU/Linux (usando por exemplo, FTP) e desempacote-o no local escolhido para os dados do *sítio*. Isto é configurado como

DocumentRoot em arquivo **httpd.conf** Apache, e normalmente está em algum lugar como **/var/www/html**.

4. Edite **httpd.conf** e acrescente **default.htm** à cláusula **DirectoryIndex**. (Por convenção, o Apache é configurado para procurar por *default/homepages* chamadas **index.html**, enquanto o IIS usa **default.htm**. Isto permite o uso dos dois nomes).

5. Nesse estágio, o *sítio* deve começar a funcionar, embora deva ser acessado pelo nome do novo servidor em vez do próprio URL. Também podem aparecer problemas onde os dados do *sítio* fazem uso inconsistente das caixas alta e baixa nos nomes de arquivos e URLs, e onde “\” tenha sido usado em URLs.

6. Se possível, teste o *sítio* neste estágio e corrija quaisquer problemas através da edição dos dados do *sítio*. Isto possibilitará melhor desempenho. Há ferramentas de checagem automática disponíveis, os quais percorrerão o *sítio* e dirão se alguns pontos de ligação apontam para locais não disponíveis. Você também pode fazer nesse estágio uma lista das páginas não alcançáveis, e passar todas por um testador HTML.

7. Caso não seja possível consertar os dados do *sítio*, acrescente estas linhas de configuração ao arquivo **httpd.conf**:

```
LoadModule spelling_module modules/mod_spelling.so
AddModule mod_spelling.c
CheckSpelling on
```

Note que isso produz uma varredura no diretório e um redirecionamento do HTTP para cada parte mal soletrada/com erros de caixas alta e baixa de um URL, portanto, preste atenção nas questões de desempenho.

8. Páginas que usam “\” incorretamente em URLs podem ser manejadas usando **mod_rewrite**, através da adição das linhas abaixo ao **httpd.conf**:

```
RewriteEngine on
```

```
RewriteRule ^(.*)\.(.*)$ $1/$2 [N]
```

Isto substitui a primeira \ por / no URL e depois repete, caso haja mais de uma \.

9. Verifique os mapas de imagem tipo servidor usando um comando do modo:

```
find /var/www/html -name '*.map' -print
```

Edite à mão, se houver somente um ou dois, ou use um *script* para consertá-los, se houver muitos.

10. Nesse estágio, todo o *sítio* deve estar trabalhando corretamente. Você pode configurar FTP, Samba ou *WebDAV* para prover acesso para a atualização das páginas.

11. Para que o *sítio* entre em funcionamento, desconecte o velho servidor e mude o endereço IP da nova máquina para substituí-lo, ou mude a entrada DNS do *sítio* para que esteja apontada para o novo servidor.

12.7.1.3. Uma configuração WebDAV simples

O *WebDAV* pode ser usado para gerenciar o conteúdo de alguns ou todos os seus *sítios*. Neste exemplo, ele é usado para todo o *sítio*, portanto não deve ser permitido qualquer outro acesso. (Outros sistemas de gerenciamento como FTP ou acesso direto ao arquivo irá confundir os clientes *WebDAV* pois eles não usam o mesmo esquema de trava).

1. Faça um diretório para as travas *WebDAV*. Deve pertencer ao mesmo usuário e grupo no qual o Apache funciona (veja opções de configuração de **User** e **Group** em **httpd.conf**). Uma boa escolha seria **/var/httpd/webdavlocks**.

2. Acrescente estas linhas à parte principal de **httpd.conf**:

```
Loadmodule dav_module libexec/libdav.so
Addmodule mod_dav.c
DAVLockDB /var/httpd/webdavlocks
```

3. Encontre o **Directory** ou **Location** associados à *sítio default*, e acrescente estas linhas:

```
DAV On
AllowOverride None
Options Indexes
AuthType Basic
AuthName "Sítio Managers Only"
AuthUserFile /var/httpd/htpasswd
<LimitExcept GET HEAD OPTIONS>
require valid-user
</LimitExcept>
```

4. Assegure-se de que os arquivos e diretórios associados sejam propriedade do mesmo usuário e grupo onde o Apache funciona, usando um comando do modo:

```
chown -R apache:apache /var/www/html
```

5. Crie o arquivo senha:

```
touch /var/httpd/htpasswd  
chown root:apache /var/httpd/htpasswd  
chmod 640 /var/httpd/htpasswd
```

6. Crie uma senha para um usuário chamado *webadmin* (ou qualquer outro nome que você escolher):

```
htpasswd -m /var/httpd/htpasswd webadmin
```

7. Reinicie o Apache ou faça com que releia seus arquivos de configuração, por exemplo:

```
/etc/init.d/httpd reload
```

8. Você pode gerenciar todo o *sítio* usando o protocolo *WebDAV*. O Windows 2000 e clientes posteriores podem acessá-lo como um “Network Place” no Windows Explorer, e os aplicativos do Office podem salvar os dados diretamente no *sítio*. O GNU/Linux fornece funções similares via **davfs**.

9. Note que o esquema aqui descrito fornece apenas segurança limitada. Você deve ler o manual do Apache para maiores detalhes sobre autenticação do usuário e escolher um esquema apropriado para suas necessidades. Pode ser necessário usar o SSL para proteger as transações; isto pode ser feito com o **mod_ssl** do Apache.

12.7.2. Bancos de Dados: mudando do Access e do Servidor SQL para o MySQL ou PostgreSQL

Muitos pequenos projetos de bancos de dados usam o Access. Este é um produto atraente para muitas pessoas porque é muito simples para iniciar, e tem uma interface familiar com o usuário. No entanto, o Access tem severas limitações; não foi desenhado para trabalho multiusuário pesado, e não consegue lidar com grandes conjuntos de dados.

Bancos de dados maiores usam SQL Server, Oracle, Sybase, DB2, etc. No caso desses sistemas maiores, a solução livre seria o PostgreSQL ou o Firebird, migrando também os aplicativos do cliente para plataformas software livre. Em alguns casos, particularmente onde a Administração tenha profunda habilidade com o banco de dados existente e esteja utilizando muitos recursos proprietários, pode ser feita inicialmente apenas a migração dos aplicativos do cliente, deixando a migração do banco de dados para uma outra etapa. Há formas padrão de conectar-se a bancos de dados relacionais pela rede, portanto a escolha da plataforma pode ser diferente para o banco de dados e os aplicativos do cliente. Além disso, a maioria dos bancos de dados proprietários são disponíveis em plataformas GNU/Linux e Unix, portanto, é possível mudar o sistema operacional sem ter que aprender um banco de dados completamente novo.

Por outro lado, bancos de dados proprietários podem ser itens muito caros, portanto é importante considerar a migração do banco de dados para Software livre, mesmo que em uma segunda etapa.

Os três mais conhecidos bancos de dados software livre são MySQL, o Firebird e PostgreSQL. São produtos maduros com grandes bases instaladas e equipes de desenvolvimento ativas. Ambos possuem bom suporte para o SQL padrão, e são capazes de desempenho muito bom.

Também vale a pena lembrar que bancos de dados não têm que ser relacionais. Algumas tarefas adaptam-se melhor com outros modelos, e o uso direto de um produto software livre como Sleepycat’s Berkeley DB pode ser extremamente eficiente. De forma similar, o modelo LDAP de

bancos de dados em rede de forma hierárquica é bastante apropriado para alguns tipos de aplicativos distribuídos.

12.7.2.1. Migrando Bancos de Dados Access

O Access só é disponibilizado em plataformas proprietárias, portanto, caso esteja planejado um ambiente completamente *livre*, seria interessante a migração da base de dados para MySQL e utilizar PHP em um ambiente intranet, em substituição dos formulários e relatório, porém demandaria um tempo maior para o desenvolvimento da interface do usuário.

Um outro cenário seria a utilização de algumas ferramentas livres que podem substituir completamente o Access, como Knoda (<http://knoda.sourceforge.net>) e o Haccess (<http://haccess.sourceforge.net>), estando este último ainda em desenvolvimento.

1. Exportação/importação à mão

Há várias formas de migrar dados do Access para outros bancos de dados. Para conjuntos de dados simples, talvez a forma mais fácil seja a de exportar as tabelas do Access como arquivos CSV (Comma Separated Values), e depois importa-las para o novo servidor. Este método requer que as tabelas sejam criadas à mão no novo servidor primeiramente, mas não necessita de qualquer *software* especial.

Como exemplo, aqui estão os comandos para criar um banco de dados com uma simples tabela e para importar um arquivo CSV para o MySQL. Primeiro entre em um *Shell prompt*:

```
mysql --user=myusername -p
```

Depois entre com o seguinte:

```
create database mydb;
use mydb;
create table mytable (
  firstname char(30),
  surname char(30),
  postcode char(10)
);
load data local infile 'exportfile.csv'
into table mytable
fields terminated by ',' enclosed by '"'
lines terminated by '\r\n';
```

2. Exportação/importação *scripted*

Existem vários *scripts* e programas que exportam um banco de dados Access completo com toda a informação necessária para recriar as tabelas em um outro DBM. Alguns deles produzem arquivos para serem copiados para a nova plataforma, enquanto outros se conectam diretamente através da rede e fazem as alterações imediatamente. Um exemplo dos *scripts* escritores de arquivos é o **exportsql2.txt** disponibilizado em <http://www.cynergi.net/exportsql>. Ele produz arquivos com instruções DROP TABLE, CREATE TABLE, e INSERT, que vão replicar o banco de dados Access em MySQL.

Vários outros ferramentas de migração são descritos no documento de Paul Dubois “*Migrating from Microsoft Access to MySQL*” (<http://www.kitebird.com/articles/access-migrate.html>).

Uma vez migrados os dados, é possível continuar usando o Access como interface do usuário, deletando-se as tabelas localmente e ligando-se às recém criadas tabelas no servidor MySQL.

12.7.2.2. Migrando bancos de dados do Servidor SQL

O processo aqui é muito parecido com o descrito acima; para bancos de dados simples é normalmente suficiente exportar os dados para um formato comum (usualmente CSV) e depois importa-los para o novo banco de dados. Bancos de dados mais completos, que incluem procedimentos de armazenagem e *triggers*, precisarão de mais esforço, e nesses casos, vale a pena procurar entre a gama de ferramentas disponíveis para auxiliar no processo de migração. Alguns deles são softwares livres, outros são comerciais. Alguns exemplos:

1. O PGAdmin é um *software* gratuito para administração de bancos de dados PostgreSQL databases. Há utilidades *plugin* que conduzem a migração de dados de outras máquinas de bancos de dados. Mais informação disponibilizada em <http://www.pgadmin.org/>.

2. O SQLPorter da Realssoftstudio - um produto comercial disponibilizado em muitas variantes, dependendo da fonte e da máquina de banco de dados alvo. Para maiores informações, veja <http://www.realssoftstudio.com/overview.php>.

3. O SQLWays da Ispirer - é um produto comercial que dá suporte a uma gama de máquinas de bancos de dados. Veja <http://www.ispirer.com/products>.

4. O SQLyog é uma outra ferramenta comercial. Ele gerencia o MySQL e também conduz a migração de dados de outros bancos de dados submetidos a ODBC: para detalhes, veja <http://www.webyog.com/sqlyog>.

5. O *sítio* do MySQL lista uma vasta gama de outras ferramentas de migração: veja a lista disponibilizada em <http://www.mysql.com/portal/software/convertors/index.html>.

12.7.2.3. Questões de migração de bancos de dados

A migração dos dados é, às vezes, a parte mais fácil do trabalho, contudo se os dados devem ser acessados pela rede como tabelas no estilo direto SQL, então não há mais muito que fazer.

É muito mais provável que os problemas venham a aparecer a partir dos utilitários auxiliares e das linguagens *scripting* que cercam qualquer banco de dados prático. O SQL é padronizado, no entanto, quase todos os vendedores de bancos de dados os estendem e encorajam as pessoas a usarem suas extensões não padronizadas.

Além disso, há várias formas diferentes para alcançar um dado resultado em SQL, e a escolha do mais eficiente pode variar de um banco de dados para outro.

Muitos aplicativos de bancos de dados são construídos com geradores de aplicativos ou construtores de formatos. Eles podem não funcionar com qualquer outro banco de dados que não aqueles com os quais foram vendidos.

Tanto o MySQL como o PostgreSQL desenvolveram-se muito nos últimos anos, por isso é importante que você leia revisões recentes ao considerar qual produto usará e se iniciará a migração.

12.7.3. Groupware: mudando do *Exchange*

O Exchange oferece serviços de correio, calendário e livros de endereços. É normalmente utilizado com o cliente Outlook no Windows, embora algumas instalações também usem o Outlook Web Access (OWA) para fornecer funções básicas através de uma interface *web*.

Todas as funções do Exchange podem ser substituídas por pacotes software livre, freqüentemente de forma muito eficiente. Os problemas aparecem quando se tenta oferecê-los sem alterações para os clientes Outlook, já que o mecanismo de comunicação entre o Exchange e o Outlook é proprietário. O Outlook é capaz de acessar alguns serviços baseados em padrões abertos, embora em alguns casos a experiência do usuário seja diferente do que a encontrada ao usar o protocolo proprietário. Como resultado, vale a pena decidir no início proceder a migração para um pacote cliente software livre ao mesmo tempo em que a migração do servidor está sendo feita, já que a população de usuários verá algumas diferenças, mesmo que tenham aderido ao Outlook. O cliente para substituição mais óbvio é o Ximian da Evolution.

12.7.3.1. Questões Gerais

Todos os usuários do Exchange terão nomes de usuário e senhas armazenados no sistema. As versões recentes do Exchange usam o *Active Directory* para isso, portanto as notas sobre migração de dados de registro do usuário que estejam em qualquer lugar do documento, também se aplicam ao *Exchange*. Resumindo, servidores com base em software livre podem acessar dados de registro via LDAP, de forma que os novos servidores possam usar o Active Directory existente ou os dados podem ser migrados para um armazém de dados com base em software livre, como o OpenLDAP.

12.7.3.2. Questões de Correio

Usuários podem ter um volume considerável de correspondência armazenada, tanto pessoal quanto compartilhada com membros de outros grupos. Deve haver um requisito legal ou de procedimento para manter um registro de toda a correspondência enviada e recebida, e nesse caso, o armazenamento e o acesso a esses dados deve ser considerado. As pessoas com computadores portáteis podem transferir toda a sua correspondência para o *laptop*, ou optar por manter uma cópia sincronizada com o documento principal no depósito central.

Ao planejar uma migração para os serviços de correio baseados no software livre, é importante localizar todos os dados armazenados e assegurar-se de que ainda estará acessível após a transição.

O Exchange pode usar grupos Windows como listas de distribuição – são os mesmos grupos que o próprio Windows utiliza para controlar o acesso. Esta não é a maneira usual de manter listas de distribuição em um ambiente software livre, porém ela pode contar com suporte, caso desejado.

Caso o Outlook seja retido como cliente correio, deve ser reconfigurado para usar IMAP em vez de acesso “original” às caixas de correio.

O Exchange não possui recurso de exportação, portanto a migração dos dados deve ser feita através de uma conexão de cliente.

Para maiores detalhes sobre sistemas de correio software livre, verifique a Seção 11.2 e o Apêndice C.

12.7.3.3. Questões de Catálogo de Endereços

Usuários do Outlook constroem um catálogo de endereços pessoal automaticamente, ao enviar e receber mensagens. Também possuem acesso a um ou mais livros de endereços compartilhados, caso usem o servidor Exchange. O conteúdo desses catálogos de endereços deve ser migrado para um formulário legível pelo software livre. Livros de endereços pessoais podem ser exportados em formulário vCard, o qual é entendido por muitos clientes correio e pode ser subdividido por *scripts* para conversão em outros formatos, caso necessário. De forma similar, livros de endereços compartilhados podem ser exportados e depois carregados em um armazém LDAP.

É provável que os principais problemas venham do fato de que o Outlook e o Exchange tendem a não usar endereços de correio padrão RFC822 internamente, e portanto os dados de livros de endereços podem não incluir endereços usáveis quando exportados. Nesse caso, será necessário algum processamento de correio, usando um *script* com acesso ao depósito do Active Directory para traduzir os endereços do “formato interno” para os endereços RFC822 padrão. Esta tradução será provavelmente necessária mesmo que o Outlook estiver retido como cliente correio, já que não poderá usar os endereços “formato interno” ao enviar correio em protocolos baseados em padrões como o SMTP.

12.7.3.4. Questões de Calendários

Algumas Administrações fazem uso considerável dos recursos do calendário do Outlook para agendar reuniões e gerenciar reservas de salas. Esses recursos podem ser usados sem o Exchange, porém há algumas limitações.

Se houver uma migração para clientes software livre planejada concomitantemente, os calendários deverão ser exportados em formato vCal e movidos para a nova plataforma de gerenciamento de calendário.

12.8. Migrando aplicativos estação de trabalho para software livre

12.8.1. Office

12.8.1.1. Conversão de documento

O OpenOffice.org é capaz de ler e escrever formatos proprietários extraordinariamente bem, portanto não é necessário converter documentos durante o processo de migração. Caso seja desejada a conversão do documento, isso pode ser automatizado com o recurso Autopilot selecionado no menu File do OpenOffice.org. Este recurso fornece uma forma de converter documentos em massa. A decisão de converter depende do uso futuro do documento. O Capítulo 5 fala sobre formatos de documentos e conversão em linhas gerais. Se os documentos vierem a ser editados repetidamente, o formato deverá ser o utilizado pela maioria dos editores.

12.8.1.2. Conversão de modelo

O OpenOffice.org pode usar modelos diretamente em formato Word 97, mas na prática, é melhor convertê-los para modelos originais e armazená-los em uma área modelo compartilhada apropriada. Esse procedimento oferece a oportunidade de testar cada modelo e corrigir cada erro de conversão. O OpenOffice.org faz, por si só, a maior parte do trabalho de conversão, e o processo

pode ser automatizado para grandes coleções de modelos, usando a função Autopilot Document Conversion, que se encontra no menu File.

Modelos de outros processadores de texto provavelmente precisarão de recriação à mão.

12.8.1.3. Conversão de Macro

O OpenOffice.org usa uma macro linguagem do tipo BASIC. É muito similar, estruturalmente, às linguagens usadas pelo Word e pelas versões posteriores do WordPerfect. No entanto, os nomes dos objetos sobre os quais trabalha são diferentes, portanto todas as macros necessitarão de algum esforço de conversão manual.

As macros são um grave risco de segurança em documentos e não são necessárias para a maioria das tarefas do dia-a-dia; assim vale muito a pena verificar se podem ser dispensadas. A maioria das tarefas de formatação são mais bem manuseadas com o uso de modelos e estilos, e a manipulação de dados simples pode ser feita usando-se formulários.

As versões OpenOffice.org a partir da 1.1 incluem um gravador de macro, tornando mais fácil a criação de macros simples, caso se julgue serem essenciais.

Não há, atualmente, qualquer meio automático de conversão de macros, embora algum trabalho sobre isso esteja sendo feito.

12.8.1.4. Processamento de texto

Há muitos pacotes de processadores de texto em uso nos sistemas Windows. Organizações bem gerenciadas tendem a tê-los padronizados em um pacote, ou talvez em transição de um para outro. Os pacotes mais comuns são:

- . *Microsoft Word e Microsoft Works*
- . *Wordperfect*
- . *Lotus AmiPro, Lotus WordPro e Lotus Notes*
- . *IBM Display Write*

Arquivos em formatos proprietários não são legíveis diretamente pelo OpenOffice.org, portanto, precisarão de conversão. É frequentemente possível exportar os arquivos do respectivo aplicativo em algum formato comum aceitável; a conversão pode, no entanto, requerer uma terceira ferramenta.

Arquivos WordPerfect ainda não são diretamente legíveis, mas há um projeto em andamento para incluir esse formato no OpenOffice.org. Há um programa de conversão baseado em *script* disponível que pode ser usado para conversão formato *bulk*.

O *sítio* <http://www.raycomm.com/techwhirl/magazine/technical/openofficewriter.html> contém uma comparação extremamente útil das funções disponíveis no Word e no OpenOffice.org. A interface usuário é suficientemente similar à do Word, para as pessoas poderem trocar de um para o outro com pouca dificuldade (embora ainda fosse melhor promover treinamento para introduzir o novo pacote de forma efetiva).

12.8.1.5. *Publishing*

A produção de documentos além da capacidade dos processadores de texto é normalmente feita com *Pacotes Desktop Publishing (DTP)*. Os comuns incluem:

- . *Framemaker*
- . *Pagemaker*
- . *QuarkXPress*

O produto software livre Scribus encontrado em <http://www.scribus.net/> pretende substituir esses pacotes e deve ser avaliado.

O OpenOffice.org tem uma capacidade muito maior que os processadores de texto comumente tinham quando os pacotes DTP foram produzidos pela primeira vez. Recursos avançados como os do *Master Documents* possibilitam que se trabalhe com grandes projetos como produção de livros, e recursos de *layout* são muito flexíveis agora.

Abordagens alternativas incluem o uso de pacotes de *post-processing*, onde o texto é marcado em uma linguagem similar ao HTML e depois convertido para seu layout imprimível final através da aplicação de folhas de estilo. Esses sistemas não-GUI podem ser muito úteis para a produção de documentos que mudam rapidamente e para impressão *on-demand* de material baseado em banco de dados.

12.8.1.6. Planilhas

Planilhas comuns baseadas em ambiente proprietário incluem:

- *Microsoft Excel*
- *Lótus 123 e derivados*

Na maioria dos casos, uma migração do Excel ou do Lótus 123 para o OpenOffice.org ou Gnumeric apresenta poucos problemas, a não ser que a planilha contenha controles ou outros mecanismos que requeiram macros. Neste caso, esses controles e macros devem ser reescritos.

12.8.1.7. Gráficos de apresentação

Em um ambiente proprietário, as apresentações são normalmente criadas usando o *Microsoft PowerPoint* ou o *Corel Draw*. O PowerPoint com seu formato de arquivo *.ppt é o mais comum.

Uma das melhores opções do software livre é o OpenOffice.org. Ele pode ler *slides* e modelos PowerPoint com muito poucos erros, e pode ser configurado para escrever arquivos *.ppt, caso desejado. Como mencionado na Seção 12.8.1.2 acima, valeria a pena traduzir em *bulk* modelos importantes para o formato OpenOffice.org original.

Os usuários devem poder trocar do PowerPoint para o OpenOffice.org facilmente, já que o conceito e o layout da tela são muito similares.

12.8.1.8. Manipulação de gráficos e imagens

Pacotes gráficos dividem-se em três principais categorias:

- Gráficos de apresentação, tratados na Seção 12.8.1.7 acima.
- Gráficos de vetores (linha), caracterizados por programas MCAD simples e pacotes como o *Microsoft Visio*.
- Gráficos Bitmap, incluindo programas *paintbrush* e pacotes de manipulação fotográfica, como o Adobe Photoshop.

1. Aplicativos para gráficos de vetores

O OpenOffice.org inclui um recurso de desenho.

O *Dia* (<http://www.lysator.liu.se/~alla/dia>) é um pacote software livre similar ao Visio. É muito utilizado para gerar diagramas de documentação, e possui filtros que lêem arquivos de outras versões do Visio (não a versão 2002). Há bibliotecas de símbolos para uma série de aplicativos. O Kivio faz um trabalho parecido e é desenhado para integrar-se bem com o ambiente KDE, porém parece mais focado em fluxogramas. O Sodipodi (<http://sodipodi.sourceforge.net/>) trabalha bem com o SVG (Scalable Vector Graphics).

Arquivos originados no Visio e em pacotes similares podem ser legíveis em software livre, porem isso deve ser testado para cada caso individual antes de planejar a migração.

2. Gráficos Bitmap

Esta categoria alcança desde os programas simples como Paint, até manipuladores de imagem avançados como o Adobe Photoshop. O mundo software livre gerou, no mínimo, tantos programas gráficos quanto o setor proprietário, e com recursos e qualidade igualmente variáveis. No entanto, um pacote se sobressai aos outros, o Gimp (<http://www.gimp.org/>).

O Gimp é capaz de ler quase todos os formatos de arquivos gráficos *bitmap* conhecidos (inclusive o formato do próprio Photoshop), e pode gerar a maioria deles também. Ele fornece todos os recursos de um bom programa *paintbrush* junto com camadas, canais e outras ferramentas avançadas familiares aos usuários do Photoshop. O Gimp é amplamente utilizado na geração e no aprimoramento de imagens para a *web* e para publicação. O único recurso importante no qual é falho é no processo completo de gerenciamento da cor, e portanto pode não ser apropriado para trabalhos de pré-impressão de alta qualidade.

12.8.1.9. Geração de PDF

Gerar um arquivo PDF em software livre é mais fácil do que no Windows, onde algo como o Adobe Acrobat precisa ser comprado. Há várias ferramentas Postscript e PDF disponibilizados em distribuições padrão para isto. Adicionalmente, o OpenOffice.org oferece uma forma de produzir diretamente o produto do PDF. A produção de um PDF pode ser configurada como um serviço de impressão, fornecendo, desse modo, um método de continuidade para usuários de ambiente proprietário. O utilitário Scribus, citado no item 12.8.1.5 é capaz até mesmo de gerar formulários PDF.

12.8.2. Correio

Há uma enorme gama de interfaces usuário para correio eletrônico, tanto para ambientes proprietários quanto software livre. Por isso, este documento pode somente delinear uma visão geral do processo de migração e suas questões. O Correio também está discutido na Seção 11.2 e no Apêndice C.

As principais questões do tipo cliente são:

- Escolha do novo *Mail User Agent (MUA)*, e conseqüentemente de sua interface usuário.
- Migração da correspondência pessoal armazenada.
- Migração dos registros no catálogo de endereços.

Independentemente do MUA escolhido, será necessário migrar a correspondência armazenada e os registros do catálogo de endereços.

Caso o MUA antigo tenha sido configurado para armazenar todas as pastas de correio em um servidor IMAP, será preciso muito pouco trabalho e o novo MUA pode ser simplesmente configurado para acessá-las em seu local. Onde os arquivos locais tenham sido usados como pastas, será necessário rastreá-los e convertê-los. Por *default*, o *Outlook* armazena correspondência em arquivos com extensão **.pst** em **C:\Documents and Settings\<username>\Local Settings\Application Data\Microsoft\Outlook**.

Ferramentas de migração úteis incluem:

- <http://outport.sourceforge.net/> - exporta dados do *Outlook* para *Evolution* etc. Está em desenvolvimento ativo e, em meados de 2003, ainda apresentava importantes limitações, incluindo a incapacidade para exportar anexos à correspondência.
- A própria ferramenta de exportação do Outlook, provavelmente escrevendo em formato CSV ou Excel. Também apresenta o problema de incapacidade de exportar anexos a esses formatos.
- <http://sourceforge.net/projects/ol2mbox> - Ferramenta do software livre para converter arquivos *Outlook* *.pst para um formato usável em correios software livre. Dá suporte aos anexos.
- *Kmailcvt* – Ferramenta software livre para converter alguns formatos proprietários para uso com Kmail.

12.8.3. Calendários e Groupware

Os calendários, junto com o gerenciamento de contatos e o correio, são usualmente agrupados em um título geral *Personal Information Management (PIM)*. Alguns pacotes integrados, como o *Outlook* da *Microsoft*, oferecem as três funções em uma interface, porém outros como o ACTI, concentram-se em gerenciamento de contatos e poderiam ser considerados próximos aos sistemas *CRM (Customer Relationship Management)*.

O alvo do software livre é o *Evolution*, que integra funções de uma forma muito próxima à que o *Outlook* faz. O *Mozilla* também poderia ser considerado assim; ele inclui um cliente correio eletrônico eficiente e existe agora um módulo calendário disponibilizado em <http://www.mozilla.org/projects/calendar>. Ele é baseado no padrão iCalendar aberto e os usuários podem publicar e partilhar calendários usando o protocolo *WebDAV*.

12.8.3.1. Calendários

Alguns dos mais bem desenvolvidos recursos de calendários software livre estão em conjuntos *groupware* baseados na *Web*, e vale a pena examiná-los como possíveis serviços amplos de organização.

Os padrões iCalendar (previamente vCalendar) definem um formato de troca para os registros de calendário. Detalhes podem ser encontrados em <http://www.imc.org/pdi> e em RFC2445, RFC2446, e RFC2447. A maioria dos jogos de calendários software livre podem lidar com dados nesse formato, portanto, é a rota de migração preferida, bem como os meios normais de gerenciamento dia-a-dia.

Algumas das ferramentas de migração mencionados na Seção 12.8.2 acima também são capazes de extrair informação sobre calendários dos arquivos de dados do Outlook, para o formato iCalendar.

12.8.3.2. Gestão de contatos

Quase todos os pacotes de correio eletrônico que já existiram definiram seus próprios formatos para o armazenamento de dados de livros de endereços. Alguns se restringem a armazenar somente endereços de correio eletrônico, porém formatos mais recentes tendem a incluir todos os tipos de informações de contatos. Essa diversidade de formatos torna a migração mais difícil, ao contrário do que poderia ser.

Felizmente, a maioria dos aplicativos de correio eletrônico, tanto no mundo proprietário quanto no mundo do software livre, tenderam a implementar os formatos de permuta iCard (anteriormente vCard) nos últimos anos. A especificação desse formato é aberta, e pode ser obtida em <http://www.imc.org/pdi> e também em RFC2425 e RFC2426. Se os detalhes do contato devem ser transferidos de um aplicativo proprietário para um aplicativo software livre, este é o formato preferido.

Uma outra forma de lidar com informação sobre contatos, é consolidá-la em um diretório que abrange toda a organização e depois acessa-la via LDAP. Certamente isso deve ser feito com dados amplamente utilizados, tais como o livro de telefones internos e a lista de correio eletrônica mantidos por muitas organizações. Contudo, não é uma substituição completa por livros de endereços pessoais: um catálogo de endereços deve ser pequeno e focado nas necessidades da pessoa que o usa, enquanto um diretório deve ser abrangente e (provavelmente) e muito grande para folhear efetivamente.

Algumas das ferramentas mencionados na Seção 12.8.2 acima também são capazes de extrair informação dos arquivos de dados do Outlook para o formato iCard.

12.8.4. Navegação Internet

Usuários de sistemas proprietários podem usar alguma versão do *Microsoft Internet Explorer* para navegação. Também é possível que alguns usem Netscape, Mozilla ou Opera.

Migrar de um *web navegador* para outro é bastante fácil para usuários, já que todos possuem recursos iguais e interfaces usuários (à parte de *navegadores* modo texto como o Lynx, por razões óbvias). A questão, para usuários individuais, estará centrada na conversão da lista de favoritos: a maioria dos *navegadores* software livre pode importar favoritos do IE e do Netscape, se instalados na mesma plataforma, porém se o OS também estiver sendo migrado, então pode ser necessário exportar primeiro favoritos no formato de um arquivo HTML.

Guia Livre – Referência de Migração para Software Livre do Governo Federal

Qualquer organização que use páginas da *web* intranet deve checar se o HTML está em conformidade com os padrões W3C, de forma que se apresente apropriadamente em todos os *navegadores*. Há ferramentas para ajudar neste assunto em <http://www.w3c.org/>.

Quaisquer páginas que dependam de JavaScript, necessitarão ser testadas com particular cuidado, já que o dialeto varia de um *navegador* para outro e o uso de extensões não padronizadas causará problemas.

Quaisquer páginas que dependam dos controles ActiveX, deverão ser redesenhadas para trabalhar de alguma outra forma, já que os *navegadores* software livre não dão suporte a essa tecnologia proprietária. O ActiveX tem um modelo de segurança muito pobre, assim, desabilita-lo é um passo valioso, em qualquer caso.

Formatos de extensão da *web* comuns, como Java, PDF, Flash, e RealPlayer recebem um bom suporte dos *navegadores* software livre (embora usem freqüentemente *plugin* sem ser de software livre), o que algumas vezes pode ser difícil de encontrar no *sítio* do fornecedor.

Outros formatos como *Shockwave Director*, precisarão usar o *plugin* CodeWeavers' CrossOver.

12.8.5. Bancos de Dados Pessoais

Pessoas com volume de dados muito grande ou complexo para uma planilha, porém não grande o suficiente que justifique um banco de dados comercial completo, freqüentemente usam o *Microsoft Access*. Este pacote oferece um depósito de dados relacionais simples, junto com *scripting* e ferramentas para construção de formulário.

A migração de dados para bancos de dados software livre está coberta na Seção 12.7.2.

Há vantagens em armazenar bancos de dados em servidores bem gerenciados, mesmo se a função central IT não esteja gerenciando os dados ou fornecendo suporte aos aplicativos. Uma migração software livre fornece a oportunidade de oferecer tal serviço de armazenamento de dados, através da configuração de um servidor onde indivíduos possam construir seus próprios aplicativos de banco de dados. Há vários pacotes baseados na *web* que poderiam ser escolhidos como base para isso, tal como o PHPmyAdmin: o *sítio* <http://www.phpmyadmin.net/documentation> oferece os detalhes.

Ferramentas com GUIs mais convencionais incluem:

- *Kexi* (<http://www.koffice.org/kexi>) - um banco de dados com a interface usuário do projeto KDE, voltado para um mercado similar ao do *Access*.
- *DBDesigner* (<http://www.fabforce.net/dbdesigner4>) - um ferramenta para usuários mais avançados, que se integra tanto ao Gnome quanto ao *KDE*.
- *Knoda* (<http://www.knoda.org/>) - outro com interface usuário simples para o *KDE*.

Nenhuma dessas ferramentas tenta ler arquivos *Access*.

12.9. Migrando serviços de impressão para software livre

Em ambientes de pequenos escritórios, é comum as impressoras estarem diretamente ligadas a estações de trabalho. Escritórios maiores, e os que requerem um alto volume de impressão, tendem a usar impressoras ligadas à rede - elas podem estar diretamente ligadas à rede ou dirigidas por um servidor de impressão.

Os ambientes software livre oferecem suporte a ambos os modelos, embora seja mais comum encontrar servidores de impressão e um pequeno número de impressoras de grande capacidade.

12.9.1. O modelo de impressão no ambiente proprietário

A impressão no ambiente proprietário é quase sempre feita a partir de um item no menu de um aplicativo GUI. É muito raro que a linha de comando seja utilizada. Os aplicativos geram produtos impressos usando um processo muito similar ao que usam para colocar informação na tela. Um conjunto de *drivers* específicos para impressão é usado pelo sistema operacional, para gerar o fluxo de dados correntes para a impressora. Esses *drivers* são normalmente fornecidos pelo produtor da impressora, e devem ser instalados no local ou no servidor de impressão antes de qualquer impressão. Em um ambiente de rede, é normalmente melhor instalar e configurar os *drivers* no servidor de impressão, de forma que os clientes não necessitem ser configurados à mão. (Ainda é necessário conectar as impressoras ao cliente: isso pode ser feito à mão ou através de um *script login*).

12.9.2. O modelo de impressão Unix e GNU/Linux

O GNU/Linux herdou seu modelo de impressão do BSD Unix. Os aplicativos geram arquivos ou fluxos de dados para impressão que são repassados para um “programa de impressão ao fundo”, que se responsabiliza pelo trabalho de impressão. Os trabalhos podem ser colocados em uma fila e passados transparentemente para outras máquinas na rede. Os primeiros sistemas Unix não possuíam uma interface independente de impressão para gerar dados de impressão, dessa forma, cada aplicativo tinha que incluir um código para cada tipo de impressora que conduzisse. No tempo em que a impressão era feita com um único tipo de caractere, isso não era problema, porem, quando os fabricantes começaram a adicionar recursos gráficos, cada um criou uma nova e diferente linguagem de impressão.

Os sistemas de impressão BSD sempre tiveram capacidade de alimentar trabalhos de impressão através de um conjunto de filtros, então as pessoas começaram a desenhar filtros que convertessem de uma linguagem de impressão para outra, para aumentar a variedade de impressoras com suporte. Muitas das melhores impressoras usadas em laboratórios de pesquisa possuíam intérpretes PostScript, portanto, o PostScript passou a ser usado como a linguagem de impressão independente comum.

A maioria dos distribuidores GNU/Linux estão substituindo agora o sistema de impressão BSD por um novo pacote chamado *CUPS (Common Unix Printing System)*, que dá suporte ao *Internet Printing Protocol (IPP)* além do tradicional protocolo **lpr**. Isso completa a transição para o novo modelo de impressão:

- Os aplicativos geram trabalhos de impressão no Postscript.
- Quando os trabalhos tiverem passado para o sistema de impressão, o aplicativo pode requisitar qualquer recurso especial que a impressora suporte (impressão dupla, dobrada, costurada, perfurada, ligada etc). As requisições têm um formato padrão, mas obviamente só serão bem sucedidas se a impressora possuir o *hardware* necessário. Há um modo padrão para que os aplicativos possam saber quais os recursos suportados por uma dada impressora.

- Os trabalhos podem ser colocados em fila local na estação de trabalho ou passados imediatamente para um servidor de impressão. O usuário não precisa saber que método está sendo utilizado.

- O sistema de impressão pode distribuir os trabalhos entre um conjunto de várias impressoras similares.
- O servidor de impressão dirige o trabalho através de um canal de filtros, que o converte em fases, para qualquer formato necessário à impressora corrente, e para controlar a comunicação com a impressora.

Há, no momento, mais de 600 modelos de impressoras conhecidas por trabalharem com esse modelo (i.e. aplicativos GNU/Linux podem acessar todas as funções disponibilizadas usando os *drivers* para ambiente proprietário fornecidos pelos fabricantes, e com resultados equivalentes ou melhores).

Embora o PostScript seja o formato intermediário mais usado, o CUPS pode ser configurado para dar suporte a quase todos os formatos de arquivos com os quais os filtros podem trabalhar. Particularmente, é comum que se permita que PDF, JPEG e alguns outros formatos sejam impressos diretamente, e alguns locais adicionam filtros que fazem impressão bonita e automática de correio eletrônico etc.

O CUPS fornece interfaces compatíveis com o conjunto **lpr** BSD e também com o **lp** do System V. Portanto, é possível substituir inteiramente os sistemas antigos por máquinas existentes (FreeBSD, OpenBSD, e a maioria das variantes Unix).

O CUPS fornece uma vasta gama de características e recursos, incluindo identificação automática de servidores de impressão, contabilidade de páginas, cotas etc. Para maiores detalhes veja o *sítio* CUPS referido abaixo.

12.9.3. Configurando um serviço de impressão baseado em software livre

Para distribuições muito pequenas, é simples configurar impressoras diretamente ligadas à cada estação de trabalho de cada cliente. Elas podem ser compartilhadas através da rede, se desejado, e o CUPS oferece suporte muito facilmente.

O uso de servidores de impressão é recomendado para todos os casos onde haja mais do que um punhado de clientes, ou quando haja um volume substancial de impressão. Devem ser instaladas uma ou mais máquinas servidoras de impressão, e elas devem receber nomes lógicos no DNS além de seus *hostnames*. Isso permite configurações para nomes de referência como **printserver.example.org** em vez de **pc35.example.org**, tornando então mais fácil reorganizar o serviço mais tarde. Todas as máquinas de clientes devem ser configuradas para dirigirem-se a um dos servidores de impressão para todas as necessidades de impressão: isso evita ter que reconfigurar quaisquer clientes quando impressoras forem adicionadas ou removidas.

12.9.4. Imprimindo a partir de clientes proprietários para impressoras GNU/Linux anexas

Há muitas formas de configurar servidores de impressão baseados em GNU/Linux para dar suporte à máquina estação de trabalho. Isso varia de acordo com o volume de esforço inicial e com o volume de esforço requerido por cliente.

12.9.4.1. Usando o protocolo lpr

Esse método é apropriado onde um número muito pequeno de clientes proprietários precisa receber suporte.

O **lpr** é um protocolo muito comum, para passar trabalhos de impressão entre máquinas Unix. Como mencionado acima, ele está sendo substituído gradualmente pelo IPP, mas é largamente implementado e pode ser utilizado a partir de muitas versões de ambiente proprietário.

1. Assegure-se de que a máquina GNU/Linux esteja configurada para aceitar trabalhos usando o **lpr**.

2. Obtenha um jogo apropriado de *drivers* de ambiente proprietário para a impressora. Em termos ideais, deveria ser o *driver* CUPS genérico, que gera PostScript portátil, porém é possível usar *drivers* específicos para impressão, caso o CUPS esteja configurado para permitir impressões simples.

3. Conecte-se à máquina Windows como Administrador.

4. Abra o programa utilitário da Rede no Painel de Controle, selecione a tecla Serviços e assegure-se de que a "impressão *Microsoft* TCP/IP da *Microsoft*" esteja listada. Adicione, se necessário (isso requer o CD de distribuição e a reiniciação do computador).

5. Configure a impressora no cliente Windows como se fosse uma impressora local (não conectada à rede). Ao selecionar uma porta para a impressora, crie uma nova "porta LPR" e configure-a para enviar os trabalhos para o servidor GNU/Linux.

Agora o cliente Windows poderá enviar trabalhos de impressão para a máquina GNU/Linux, porém as ferramentas Windows podem não conseguir ver e manipular trabalhos em fila. O CUPS fornece suporte ao gerenciamento baseado na *web*, portanto os usuários devem ser avisados para usarem esse recurso, quando necessário.

12.9.4.2. Usando cotas de impressoras

Esse método também é apropriado para um pequeno número de clientes proprietários. Ele trabalha com Windows 95/98/ME, bem como com Windows NT/2000.

1. Instale e configure o Samba no servidor GNU/Linux. Siga as instruções para criar cotas de impressoras: é fácil fazer o Samba criar automaticamente uma cota para cada impressora que o servidor reconheça.

2. Em cada cliente Windows, use o *Add Printer Wizard* para acrescentar uma impressora conectada à rede. Você deve ser capaz de folhear uma lista de servidores para encontrar o servidor que você quer usar. Você vai precisar instalar *drivers* de impressoras localmente, na máquina do cliente.

Como recurso para refinar o esquema, o Administrador pode usar o *Add Printer Wizard* para carregar *drivers* de impressoras no servidor Samba, de forma que não seja necessário instalá-los individualmente nos clientes.

12.9.4.3. Usando configuração *Point and Print*

Este método é apropriado para instalações maiores e para aquelas em que máquinas de clientes novos devem ser configuradas por equipes menos qualificadas. Requer um pouco mais de esforço para configurar num primeiro momento, porém é mais fácil de usar depois de pronto. O processo é bastante complicado, portanto, consulte o *HOWTO* do Samba para maiores detalhes.

1. Instale e configure o Samba no servidor GNU/Linux. Para suporte *Point and Print* deve ser a versão 3.0 ou acima, embora muitas funções tenham suporte em 2.2.4. Assegure-se de que o Samba esteja construído com suporte CUPS.

2. Configure o CUPS para fornecer suporte às impressoras Windows, acrescentando o pacote de *driver* CUPS.

3. Use `cupsaddsmb` para instalar os *drivers* Windows do CUPS para o Samba.

4. Conecte-se a partir de um cliente Windows usando uma identidade que permita modificar configurações de impressão no servidor, e configure as características da impressora *default* apropriadamente (tamanho do papel etc). Isso é mais complicado do que parece, pois o Windows fornece duas janelas idênticas em partes diferentes da configuração GUI e somente uma delas afeta as configurações *default*. (Veja o *HOWTO* do Samba para detalhes).

5. Em cada cliente Windows, folheie a vizinhança da rede para encontrar o servidor. Clique na impressora que você deseja e “Conecte-a”. A impressora agora aparece na lista de impressoras locais e pode ser utilizada facilmente.

Grandes instalações tenderão a usar um *script* de conexão para o item 5, em vez de fazê-lo à mão.

12.9.5. Imprimindo esquemas de migração

Para organizações pequenas, com algumas estações de trabalho e impressoras, é simples configurar um servidor de impressão baseado em GNU/Linux e simplesmente reconfigurar cada estação de trabalho cliente à mão. Caso haja diversas impressoras compartilhadas ligadas às máquinas estação de trabalho, pode valer a pena aproveitar a oportunidade para consolidá-las em um servidor de impressão. Isso pode tornar-se mais fácil ajustando placas *ethernet* às impressoras, onde elas suportem o recurso (isso também pode oferecer uma melhora substancial do desempenho sobre interfaces seriais ou paralelas). Impressoras paralelas só podem ser ligadas à rede usando caixas de impressão de rede.

Locais maiores certamente irão beneficiar-se do uso de um ou mais servidores de impressão. Essas máquinas podem desempenhar outras tarefas também, porém, se há um volume substancial de impressão, deve-se lembrar que converter o PostScript para outros formatos gera um trabalho intensivo para a CPU e a capacidade das máquinas deve ser avaliada apropriadamente. Vale a pena fazer a configuração *Point and Print* completa caso tenha que ser fornecido suporte aos clientes proprietários, já que a migração de máquinas clientes dos servidores de impressão do ambiente proprietário para a plataforma livre pode ser feita com um simples *script* de conexão.

12.9.6. Problemas Potenciais

Há alguns problemas comuns que podem ser evitados através de um planejamento cuidadoso:

1. Assegure-se de que cada impressora seja controlada apenas por um servidor. Faça com que todas as outras estações de trabalho e servidores enviem trabalhos para a impressora através do servidor que os controla. Isto é particularmente importante com impressoras ligadas à rede. Caso isso não seja feito, a impressora poderá receber dois ou mais trabalhos ao mesmo tempo, com possível corrupção do produto.

2. Se possível, tente fazer com que somente um jogo de *drivers* formate a produção de uma impressora. Seria provavelmente melhor fazê-lo no servidor controlador, porém não necessariamente – dependerá, até certo ponto, do servidor que possui o melhor *driver* para a impressora. Outras máquinas processando a impressão do produto, devem tratá-lo como dados brutos. Caso isso não seja feito, um *driver* poderá tentar formatar um fluxo de produtos já formatado, corrompendo a produção. Isso só tende a tornar-se um problema quando o produto formatado contém dados binários.

12.9.7. Informações adicionais sobre impressão

Uma boa parte da informação está disponibilizada na *Web*. Estes *sítios* são pontos de partida úteis:

<http://www.cups.org/> - CUPS, o Common Unix Printing System.

<http://www.linuxprinting.org/> - o *sítio* Linux Printing, com uma vasta gama de informação útil.

<http://www.linuxprinting.org/kpfeifle/SambaPrintHOWTO> - o HOWTO do Samba Printing. Note que este é o *sítio* de distribuição do autor e que o documento é um trabalho em curso. Procure pela última minuta.

12.10. Aplicativos legados

Os aplicativos que não possuem os seus correspondentes em software livre e que não possam ser recompilados para trabalhar como software livre, terão que trabalhar em uma máquina rodando o sistema operacional que possa suportar este aplicativo legado, ou em um emulador de *hardware* ou *software*. As técnicas discutidas no capítulo são aplicáveis.

12.11. Proteção antivírus

Um pacote antivírus atualizado é essencial em qualquer ambiente. Até mesmo organizações com pouco interesse em segurança ignoram esta proteção a seu próprio risco.

Em contraste, há muito poucos vírus viáveis que afetam os sistemas software livre. Como consequência, a proteção contra vírus, nos ambientes de software livre, limita-se normalmente à varredura de correio eletrônico para evitar a passagem de vírus para usuários de ambiente proprietário.

Houve ataques automatizados a sistemas de software livre no passado. Um forte foco em segurança nos anos posteriores ao evento reduziu os riscos consideravelmente, porém ainda é possível que um vírus efetivo possa um dia ser liberado. Boas práticas de gerenciamento de sistemas e educação contínua do usuário são atualmente uma defesa maior que um *software* antivírus.

Guia Livre – Referência de Migração para Software Livre do Governo Federal

Há atualmente dois projetos anti-vírus software livre conhecidos, o Open Anti Virus (<http://www.openantivirus.org/>) e o ClamAV (que aparentemente desapareceu da rede). Ambos encontram-se em estágios prematuros de desenvolvimento e nenhum dos dois é recomendado para o serviço ainda. Muitos dos produtos antivírus comerciais possuem versões que funcionam em plataformas software livre. Essas versões não são inteiramente equivalentes às suas contrapartes no ambiente proprietário – no momento elas estão direcionadas para funções como varredura de correspondência em vez de detecção imediata em código executável de vírus, como é comum nos sistemas proprietários. No entanto, como explicado acima, detecção imediata é, na maior parte, desnecessária nos sistemas software livre; a varredura da correspondência é geralmente suficiente.

12.12. Referências

<http://www.samba.org/> The Samba SMB file/print/domain server
<http://www.openldap.org/> The OpenLDAP directory server
<http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html>
The Linux-PAM system administrator's guide
<http://samba.mirror.ac.uk/samba/docs/man/Samba-HOWTO-Collection.html>
The main Samba HOWTO collection
http://www.csn.ul.ie/~airlied/pam_smb_pam_smb_PAM_module_to_authenticate_GNU/Linux_users_with_SMB
<http://samba.idealx.org/index.en.html> IDEALX tools and HOWTOs relating to Samba (algumas em Francês).

13. Cenário 2 - Unix

A Administração possui servidores Unix (“big Unix” - Solaris, HP/UX, AIX, OSF/1, etc). Usa PCs com aplicativos cliente-servidor. Algumas estações de trabalho Unix e Terminals X.

A migração das estações de trabalhos PC será similar à do Cenário 1 acima. É provável que as estações de trabalho e os Terminais X estejam funcionando com aplicativos baseados em X, os quais devem funcionar sem qualquer problema nas novas estações de trabalhos software livre. O principal problema aqui é migrar os servidores.

Migrar do Unix para o GNU/Linux é similar a transferir de uma versão do Unix para outra. Tendo em vista que o termo Unix inclui AT&T, BSD e os códigos base OSF/1, que são implementações diferentes do padrão POSIX – como é o GNU/Linux. As diferenças aparecem quando um programa usa recursos que estão fora do POSIX, coisas típicas como administração de sistema e recursos para melhorar o desempenho. Programas escritos de forma pobre, com pouca atenção ao POSIX, também apresentarão problemas – escrever programas portáteis é uma arte, “programas cultivados em casa” tendem a demandar algum trabalho (eliminação de todos os avisos do compilador produzidos no mais alto nível de advertência é um bom começo). Contudo, é provável que os problemas sejam detalhes e não arquiteturais, fundamentais. Os Unices usam protocolos abertos, tais como TCP/IP e serviços comuns como DNS and DHCP.

É provável que também a configuração seja diferente. No entanto, não é provável que os dados do sistema sejam mantidos em um formato proprietário, conseqüentemente, manipula-los para estarem em conformidade com os requisitos GNU/Linux deve ser bastante fácil. Isso inclui nomes de usuário e senhas, embora diferenças sutis possam significar que a simples transferência não é possível.

Se o código de fonte estiver disponibilizado, a recompilação deve permitir que o código seja transferido. Contudo há algumas questões que precisam ser comentadas:

1. Não há padrão para a localização dos arquivos e eles podem estar bem criptografados nos programas (como `/usr/bin`, `/usr/local/bin` ou `/opt/bin`).

2. Pode haver valores diferentes para as constantes do sistema, por exemplo, o número máximo de arquivos que podem estar abertos ao mesmo tempo.

3. Diferenças sutis na linguagem de programação, por exemplo ksh e pdksh. Compiladores C diferentes são mais ou menos exatos na checagem da sintaxe, portanto, o código pode ser permitido em uma máquina e apresentar erro em outra.

O código pode não ser portátil por alguns motivos, por exemplo:

- Uso não portátil das constantes, como usar número em vez de SIGPIPE (algo definido em um arquivo cabeçalho C). Isso é um exemplo de um programador programando para o sistema operacional em vez do padrão POSIX.
- Suposições sobre o comprimento da palavra ou a ordenação do *byte*.

O gcc, compilador no GNU/Linux, tem opções bastante flexíveis nessas circunstâncias.

4. Cada Unix pode ter arquivos cabeçalhos e bibliotecas diferentes. Também podem estar em locais diferentes. Os locais e nomes podem ser alterados automaticamente, uma vez

Guia Livre – Referência de Migração para Software Livre do Governo Federal

encontrados. No entanto, se a biblioteca ou o arquivo cabeçalho apresentam comportamentos diferentes, será preciso intervenção manual.

Por exemplo:

- A semântica de algumas chamadas de bibliotecas difere:
 - Como *threads*;
 - Exec (ignorado o bit setuid em *scripts*);
 - Asynchronous I/O;
 - Ioctl para controle tty.
- Valores diferentes para *errno*.

O código original pode usar aplicativos proprietários ou bibliotecas não disponibilizadas em GNU/Linux. Pode precisar ser reescrito para usar o que estiver disponível em GNU/Linux. Pode ser o caso se forem requisitadas interfaces de *hardware* especiais, por exemplo, uma placa de fax. Há, psoftware livreiramente, muito trabalho a fazer nessas circunstâncias.

5. Os *makefiles* que ajudam na construção de aplicativos podem precisar de atualização para refletir as diferenças mencionadas acima.

6. Os aplicativos podem fazer suposições sobre subsistemas específicos como os de impressão e bancos de dados. Isso significa, por exemplo, que o código SQL pode ter que ser reescrito.

7. Transferir qualquer código para um novo *hardware*, compilador ou sistema operacional pode apresentar erros no programa, os quais estiveram sempre lá, porém nunca apareceram, por exemplo, por estar a memória disposta de forma diferente, números inteiros tem tamanhos diferentes ou os bytes são ordenados de forma diferente.

As seguintes referências dão mais detalhes:

<http://www.linuxhq.com/guides/LPG/node136.html>

http://www1.ibm.com/servers/esdd/articles/porting_linux/index.html?t=gr,l=335,p=PortSolaris2

Linux

<http://www-106.ibm.com/developerworks/linux/library/l-solar/?open&t=gr,l=921,p=Sol-Lx>

<http://www-1.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gm130115.pdf>

<http://www.unixporting.com/porting-guides.html>

14. Cenário 3 – Mainframe

A Administração é baseada em uma mainframe (pode funcionar com MVS, VM/CMS, AS/400, ou até Unix). A maioria dos usuários possui acesso à tela verde. Há poucos PCs, a maior parte sendo usados como emuladores de terminais, mas com um ou dois aplicativos.

Este cenário é similar ao do *Thin Client* no que diz respeito a estação de trabalho, particularmente se a arquitetura for mantida.

Não há dados de migração de servidores disponíveis.

15. Cenário 4 – Cliente Leve

A Administração usa estações de trabalho tipo Cliente Leve com acesso via Citrix ou similar a uma mescla de Windows e aplicativos Unix.

O uso do ABR não é assumido aqui porque as razões originais de utilizar um modelo *thin client* possam ainda ser consideradas consistentes. Contudo, se for contemplada uma mudança para o ABR, muitos dos mesmos problemas do Cenário 1 vão aparecer. A migração neste Cenário, sob esta pressuposição é, portanto, muito simples, já que a arquitetura não vai mudar.

Pelo fato de o cliente ser muito *thin*, tudo que é necessário é um *software leitor de fonte aberta* para cada protocolo requerido. O sistema de janelas não necessita de muita funcionalidade, portanto será suficiente um gerenciador de peso leve como o *twm*.

Os seguintes protocolos podem obter suporte (entre outros):

1. HTTP. Qualquer *navegador* software livre será suficiente. A capacidade de funcionar com Javascript e Java terá que ser investigada. Adicionalmente, quaisquer extensões requisitadas terão que receber suporte diretamente, através de um substituto usando o pacote *plugger* ou através do pacote proprietário *CodeWeavers CrossOver Plugin*.

2. ICA. Este é o protocolo Citrix proprietário. O Citrix fornece um leitor ICA “preço zero”, porém não software livre, que trabalha no GNU/Linux.

3. RDP. Este é o protocolo utilizado pelo Servidor de Terminal Windows. Há disponível um leitor software livre para RDP em estação de trabalho.

4. VT220, VT100 etc. Estes protocolos DEC recebem suporte de *xterm* usando o conjunto de variáveis ambientais *TERM* apropriadas. A conexão com o *host* é feita via *telnet*.

O *xterm* pode emular muitos tipos de terminais diferentes através da mudança do valor da variável *TERM*. Por exemplo, a configuração de **TERM=prism9** vai emular o protocolo usado pelas máquinas *PRIME*. Todas as emulações assumem a conectividade *telnet* ou similar e um protocolo baseado em caractere em vez de baseado em página.

5. 3270. O programa *x3270* fornece o suporte apropriado. A conexão com o *host* é feita via *telnet*.

6. X. Este é o protocolo de exibição original BGNU/Linux e, portanto, não deve apresentar problemas.

Há produtos proprietários para alguns dos protocolos mais herméticos.

O *Linux Terminal Server Project (LTSP)* <http://www.ltsp.org/> fornece um número de *kits* para construir dispositivos *thin client* baseados em GNU/Linux. Este é um projeto extremamente ativo e a qualidade do *software* parece ser muito boa.

As mudanças requisitadas nos servidores são similares às considerações discutidas nos outros Cenários.

Apêndice A – Estudos de Casos

Os estudos de caso disponíveis até esta versão são os seguintes, listados nas próximas páginas:

- Ministério do Desenvolvimento Agrário
- Ministério das Comunicações
- RADIOBRÁS
- Marinha do Brasil

Ministério do Desenvolvimento Agrário

Plano de Migração para Software livre

Paulo Ricardo Carvalho de Oliveira
Coordenador Geral de Modernização e Informática
Ministério do Desenvolvimento Agrário

Os Motivos

1. A oportunidade de realizar a autonomia da REDE MDA, concebendo uma gestão completa dos recursos de rede, dos sistemas de controle e de comunicação entre as unidades do MDA.
2. A possibilidade de realização de uma customização dos sistemas e serviços, através de uma completa integração dos processos que são os principais desafios da Administração Pública:
 - ✓ Desburocratização;
 - ✓ Aplicação dos princípios da qualidade total;
 - ✓ Comunicação multimídia;
 - ✓ Prestação de serviços;
 - ✓ Transparência total;
 - ✓ Reengenharia tecnológica;

Plano de Ação

O projeto de migração previu a contratação de uma consultoria especializada, criação da sala da REDE MDA, aquisição de novos computadores servidores com Software livre, aquisição de novas estações de trabalho com a suíte Openoffice.org instalada, curso de introdução ao Openoffice.org, desenvolvimento de sistemas de controle interno, desenvolvimento da Intranet e desenvolvimento do portal do MDA.

Definimos como foco de atuação a migração de 95% dos servidores de rede e também a migração dos aplicativos de automação de escritório de 30% dos usuários no período de curto prazo. Esta estratégia foi a escolhida por ser considerada a menos impactante para os usuários da REDE MDA.

Aspectos Culturais

A utilização do Software livre deve ser considerada como uma conquista participativa, pois envolve cada profissional, individualmente, em todas as etapas e depende do seu interesse em aceitar o desafio da mudança.

Os fatores culturais são os mais complexos para lidar com a mudança nas organizações. Existe a necessidade de motivação constante nas equipes para ampliar a flexibilidade, dotando-as de capacidade para enfrentar os desafios que a modernidade impõe às organizações.

As tomadas de decisão exigem o apoio da administração superior, que deve conhecer as potencialidades e dificuldades da utilização do SL na condução de todos os profissionais, para a construção dessa nova cultura organizacional. O apoio político é imprescindível para que haja possibilidade de implementações que alterem as rotinas das instituições.

No MDA, o Ministro Miguel Rosseto sempre foi entusiasta do SL e incentivou as primeiras edições do Fórum Internacional do Software livre (FISL) em Porto Alegre/RS, no cargo de vice governador do estado, durante o governo Olívio Dutra.

Capacitação dos Usuários e Equipe Técnica

O Software livre é um sistema em desenvolvimento e possui canais que proporcionam aos usuários participarem deste desenvolvimento, bastando que estes estejam capacitados para interagir com os sistemas e compreender o seu funcionamento, desde o nível mais básico, como utilização de softwares clientes de correio, editores de texto ou acesso à internet, até o nível avançado, como desenvolvimento de sistemas integrados online.

A migração impõe a necessidade de capacitação de todos os usuários e principalmente dos profissionais da área do desenvolvimento e de rede. A capacitação deve ser realizada constantemente e possuir canais de comunicação ágeis. É necessário um monitoramento da evolução deste conhecimento em todas as áreas.

Para enfrentar o nosso desafio, foi realizada uma pesquisa para identificar o perfil do usuário da REDE MDA, onde foi constatado um alto índice (85%) dos profissionais cursando o 3º grau ou com 3º grau completo, fator considerado como positivo para implementação das mudanças previstas. Havia um processo de aquisição de 100 estações de trabalho, onde foi definido que os aplicativos de automação de escritório seriam todos da suíte Openoffice.org. Para estes usuários foi contratada uma empresa que realizou o curso de Introdução, divididas em turmas de 8 horas de aula, distribuindo apostila e CD com softwares livres e gratuitos. Um total de 90 profissionais foram treinados. Numa pesquisa de satisfação, respondida por 50% dos alunos, 85% consideraram o curso como bom ou ótimo.

Para os profissionais de rede e de suporte, serão realizados cursos de Linux em empresas tercerizadas.

Os Serviços de REDE e Correio Eletrônico

Na área de rede, 90% dos serviços funcionam atualmente na plataforma de SL e existem ferramentas de monitoramento do hardware e dos serviços através de gráficos on-line. Contamos com 7 servidores espelhados, através do serviço de Alta Disponibilidade, garantindo a estabilidade dos serviços. No serviço de correio eletrônico, possuímos barreira de vírus e de spam. Estão sendo desenvolvidos sistemas de gestão e de integração dos serviços, no qual se destaca um Sistema on-line de Gerenciamento do Postfix (Servidor de Correio), através de uma interface gráfica, com ferramentas de importação e exportação de usuários, gerenciamento de aliases, gerenciamento de informações dos usuários e outras funcionalidades. Está previsto também um Sistema on-line de Gerenciamento do Samba (Servidor de Rede), integrado com o Postfix e semelhante nas funcionalidades.

Segue tabela dos serviços e dos Softwares Livres utilizados:

Tipo de Serviço	Software livre Utilizado
Sistema Operacional	RedHat 9.0
Servidor SMTP	Postfix
Servidor POP3	ipop3d
Webmail	Squirrelmail
Anti Vírus	Clamav
Servidor Backup	Amanda
Servidor de Arquivos	Samba
Servidor de Logon	Samba
Servidor DNS	Bind
Servidor Firewall	Ipchains
Servidor HTTP	Apache
Servidor Base de Dados	PostgreSQL e MySQL
Servidor Gráficos Monitores	LRRD

Customização dos Sistemas

Na área de desenvolvimento de sistemas foi possível a alteração imediata do Portal estático do MDA para um sistema de gerenciamento de conteúdo, o NAWEB, que possibilita gerenciamento on-line, mecanismo de busca, edição de notícias e outras funcionalidades. O NAWEB possibilitou a criação do projeto "SACI LIVRE", SACI é uma sigla de Sistema de Administração de Conteúdo Institucional na Internet, desenvolvido em software livre que possibilitará a administração de vários portais institucionais. Através de ferramentas desenvolvidas em módulos, o sistema permitirá a colaboração da comunidade do Software livre e proporcionará uma nova concepção de administração de portais institucionais corporativos.

Utilizando a linguagem PHP e o Banco de Dados PostgreSQL, ferramentas bastante populares e robustas, padronizamos os Sistemas Online e começamos a desenvolver uma série de sistemas de controle interno, totalmente integrados com os sistemas de controle de rede, correio eletrônico e com a Intranet.

A equipe de desenvolvedores é o pilar da migração do MDA, pois é dela a tarefa de construir as ferramentas que substituirão as aplicações proprietárias para aplicações em SL. Outra tarefa da equipe de desenvolvimento é a de definir e sustentar os padrões que orientarão o desenvolvimento dos novos sistemas nas unidades do MDA.

Quando os sistemas são customizados, as tarefas diárias viram processos digitais que são aprimorados com a ferramenta *on-line*. As aplicações mais modernas possibilitam uma série de funcionalidades:

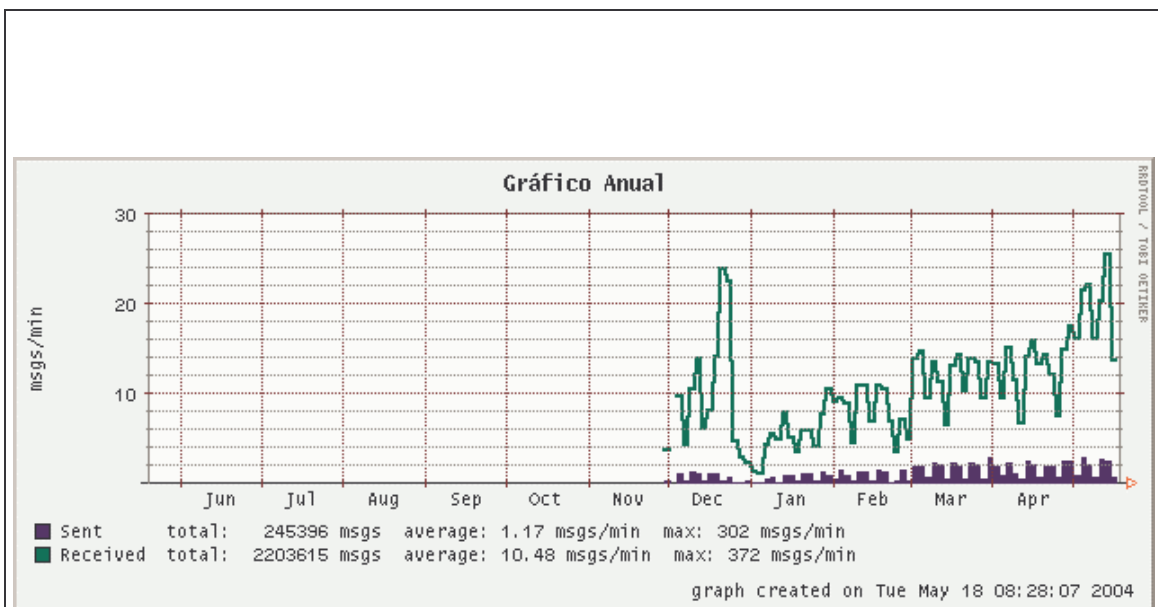
- ✓ Controle total do fluxo de informações;
- ✓ Unicidade da informação;
- ✓ Comunicação digital e multimídia;
- ✓ Pesquisa e *garimpagem* de dados;
- ✓ Documentação e registros;
- ✓ Interação em tempo real;
- ✓ Interface
- ✓ Universalidade de Acesso

Todos os projetos estão concebidos para serem integrados em módulos, isso torna mais simples a construção dos primeiros pilares e possibilita desenvolver mais de um sistema por vez. Já estão prontos os módulos de controle de acesso e segurança - CONTRA e o módulo de Gerenciamento de usuários de correio - POSTMAN. Inauguramos recentemente nossa nova Intranet e lançamos o novo Sistema de Atendimento ao Usuário - SISAU, que possibilita um gerenciamento total das solicitações de serviços e de tarefas.

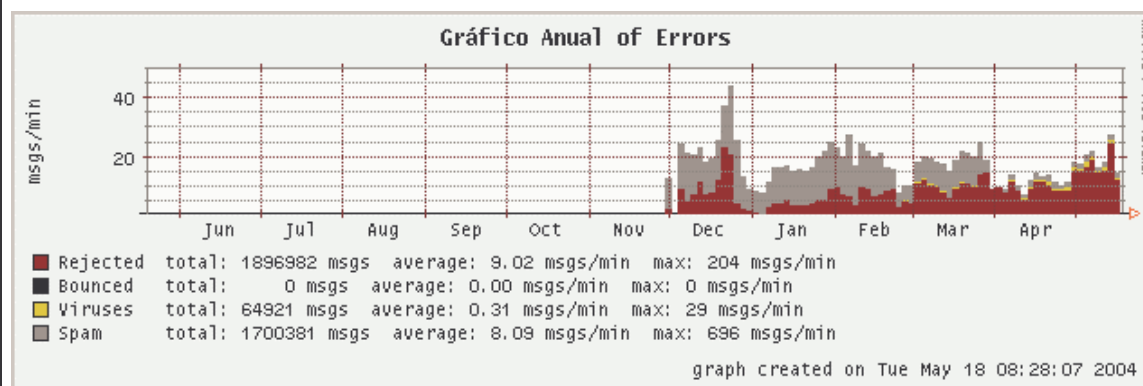
Os Desafios Enfrentados

Para a migração do serviço de Correio Eletrônico, os softwares clientes de correio de 100% dos usuários foram alterados. Essa tarefa foi certamente a mais trabalhosa, pois apesar de criarmos um *passo-a-passo* para auxiliar na utilização da nova ferramenta, recebemos algumas críticas, o que não deixa de ser um fato normal.

No servidor de correio eletrônico, conseguimos uma grande redução de paradas dos serviços e uma ampliação da utilização dos serviços. Atualmente estamos enviando em média 3500 mensagens e recebemos em média 38.000 mensagens em dias úteis. O gráfico abaixo representa essa evolução:



Outros serviços relevantes são as ferramentas de bloqueio de vírus e de spam, que possibilitaram maior segurança e estabilidade para a REDE MDA. O gráfico abaixo ilustra a evolução desse mecanismo de controle:



Para a migração do serviço de REDE, a maior parte dos serviços foram implementados com amplo sucesso e total transparência para os usuários da rede.

Na implementação do Openoffice.org o desafio maior foi a adaptação dos usuários, que apresentaram uma pequena dificuldade na formatação dos documentos migrados dos aplicativos proprietários. Os usuários avançados tiveram problemas no suporte, pois o curso realizado contemplava apenas informações básicas de comparação dos aplicativos. As operações avançadas necessitaram de suporte e nossa equipe não estava plenamente capacitada para solucionar todos os problemas.

Em termos gerais, o maior desafio para uma instituição pública no processo de migração é a capacidade de planejamento e aquisição dos equipamentos necessários. No caso do MDA, o tempo de implementação poderia ser reduzido pela metade, se esse processo burocrático de licitação fosse mais ágil.

Experiência Adquirida

Procuramos pontuar os itens que consideramos relevantes para tornar o nosso processo de migração mais confiável e sustentável:

Válvula de Escape - Sempre desenvolver o plano com alternativas de retorno à situação anterior, caso a implementação de um serviço não obtenha sucesso. Isso evita o desgaste de parada dos serviços, que pode ser alvo de críticas, gerar transtornos e trabalho dobrado.

Comunicação - Comunicar sempre as paradas de serviços e as atividades, com a devida antecedência para que todos acompanhem a evolução do processo de migração.

Pesquisa Contínua - Criar canais de comunicação na Intranet e na Internet, com os documentos *passo a passo* de cada novo software disponível. Ampliar o acervo da biblioteca sobre os softwares escolhidos.

Horários Alternativos - Realizar todas as mudanças possíveis fora do horário do serviço dos usuários, para minimizar os impactos das atividades para os usuários.

Resultados Positivos

A maior vantagem é a flexibilidade que o desenvolvimento de sistemas em Software livre proporciona, fortalecendo a melhoria contínua na área de TI, dotando as unidades executoras de ferramentas de suporte para maximizar a utilização dos recursos na execução das ações do MDA ao seu público beneficiário e na transparência dos resultados ao restante da sociedade.

Ministério das Comunicações

Migração para Software Livre

Introdução

Em junho de 2003 iniciamos o estudo de viabilidade do projeto de migração para software livre do Ministério das Comunicações, e desde então estamos trabalhando para concretizar o mais rapidamente possível essa solução.

Escopo

Migração da plataforma operacional dos servidores do CPD, substituição dos sistemas operacionais, suítes de escritório e outros aplicativos das estações de trabalho e substituição das ferramentas de administração de rede e desenvolvimento de sistemas.

Planejamento e Execução

Realizamos um planejamento estratégico com toda a equipe de técnicos da Coordenação de Informática do MC, durante um dia, fora do local de trabalho para que pudessemos aproveitar o máximo possível o tempo sem que houvesse qualquer interferência de trabalho. Além disso, disponibilizamos de um profissional facilitador do processo de planejamento para ajudar-nos na tarefa proposta.

Conseqüentemente, várias ações foram definidas e estão sendo seguidas como plano de trabalho deste Ministério. Abaixo explicitamos as ações e seus desdobramentos.

- a) Patrocinador do projeto – é fundamental que o projeto seja apoiado pela alta administração do órgão. Sem esse apoio não podemos implementar o projeto, pois ele não estaria na política estratégica da organização. O apoio deverá ser renovado sempre que houver uma substituição da pessoa do patrocinador.
- b) Estudo de viabilidade – buscamos comparar o gasto anual com licenciamento de software com o gasto em substituições (consultorias e treinamentos), além da compatibilidade técnica entre os programas que deveriam ser substituídos. Desse levantamento, mapeamos todos os possíveis problemas de adequação dos novos programas.
- c) Decisão sobre a distribuição que seria utilizada – inicialmente decidimos utilizar a distribuição Debian nos servidores, tendo como premissa, que essa distribuição não poderia ser adquirida por nenhuma empresa de tecnologia da informação.

Assim, iniciamos concomitantemente, os testes com a distribuição Kurumin em ambiente corporativo. Ela mostrou-se ser uma ferramenta de rápida configuração, tendo permitido diminuir o tempo de instalação da estação para o usuário em relação com as máquinas que usam programas proprietários.

Contudo, a distribuição ainda não está completa para execução no ambiente corporativo. Nossa equipe teve que construir alguns scripts para que o ambiente corporativo fosse incorporado e pudesse funcionar adequadamente. Entretanto, é uma ferramenta que pode facilitar a introdução do software livre nas estações, sem custo e com uma excelente performance.

- d) Capacitação da equipe técnica – investir na capacitação de seus técnicos para que eles

possam atuar sobre esse novo paradigma é fator crítico para o sucesso do seu projeto e deve ser a primeira ação efetuada. Sem o comprometimento da equipe técnica e sua habilidade em contornar determinadas situações, o projeto ficará sem rumo e fadado ao fracasso. No nosso caso, definimos os treinamentos técnicos em sistema operacional GNU-Linux de acordo com o perfil do profissional, podendo dessa forma adequar o conteúdo programático de cada treinamento.

Caso possua uma equipe terceirizada, de acordo com cada contrato de serviço, pode-se negociar com a empresa o treinamento dos profissionais. No caso do nosso projeto, definimos junto à administração do MC o treinamento para alguns servidores do quadro e para os colaboradores terceirizados negociamos com a empresa a sua capacitação.

- e) Migração dos serviços básicos de rede (não críticos) – A equipe técnica do MC, após a sua capacitação, iniciou a substituição de alguns serviços não críticos de rede que estavam em software proprietário, para o seu correlativo em software livre. Essa mudança foi totalmente transparente para os usuários do Ministério. Hoje um terço dos serviços de rede já estão migrados e sem que nenhum apresente qualquer alteração ou falha.
- f) Sensibilização dos usuários – a mudança de paradigma é grande para o usuário das estações de trabalho. Essa resistência pode ser diminuída se eles forem preparados para a mudança, buscando mostrar a importância dessa alteração para o nosso país.

Nós realizamos três dias de palestras sobre software livre, com a participação do Sr. Djalma Valois, mostrando para os usuários a necessidade de mudar e que eles não deixaria de cumprir suas tarefas, pois todas as suas principais ferramentas já possuíam substitutas em software livre.

- g) Consultoria técnica especializada – Foi definido como crítica a migração dos serviços fundamentais de comunicação das estações com o ambiente externo e a configuração de alta disponibilidade dos serviços. Para solucionar esse problema, preparamos uma licitação para auxiliarmos nessa tarefa crítica que esperamos finaliza-la em dezembro/2004.
- h) Capacitação dos usuários – Configuramos um projeto de capacitação dos usuários para uso das estações com a preocupação de não deixa-los despreparados para a tarefa.
Nossa intenção é construir uma cultura de treinamento em informática no órgão, capacitando os usuários para as mudanças desejadas, sem contudo, traumatiza-los com essas mudanças.
Nosso projeto prevê a mudança da plataforma operacional da estação de trabalho, simultaneamente com o término do treinamento do usuário.

RADIOBRÁS

Migração na RADIOBRÁS

Para não haver solução de continuidade nos trabalhos desempenhados na empresa e possibilitar a implantação do Linux nas estações de trabalho. O procedimento a ser adotado é:

- Nas estações que só utilizarem a ferramenta Office (principalmente Word e excell) e acesso à internet, será instalado o sistema operacional Linux e o Openoffice (ambos open source e free).
- Nos microcomputadores que utilizarem programas proprietários (como por exemplo o sistema de recursos humanos, o sistema de publicidade legal, etc) será necessário a utilização de um software para acessar tal recurso. Este programa é o Rdesktop.
-

Tendo estas diretrizes como norte, foram eleitos equipamentos em diversas áreas, dando preferência a aqueles que não necessitassem utilizar o Rdesktop, por um problema de infraestrutura e de licenças.

Em julho de 2003 diante de iminente migração, foi iniciado o treinamento em sistema operacional linux para a equipe de informática. Foi ministrado um curso com carga horária de 60 horas. Foram treinados tanto o pessoal de suporte, bem como os desenvolvedores.

Em setembro/2003 com o apoio do ITI, PRODABEL e UFMG foi efetuada a migração de 14 (quatorze) estações de trabalho, em 3 áreas da administração. A distribuição utilizada foi o Libertas 2.0 da PRODABEL e com o apoio da UFMG foi iniciada a autenticação destas estações no LDAP.

Como havíamos feito a migração e o pessoal da informática continuava extremamente inseguro no tocante principalmente ao suporte aos usuários e serviços. Enfrentamos diversos problemas com impressão e acesso aos arquivos do legado.

Com as dificuldades enfrentadas no dia-a-dia, resolvemos continuar testando outras distribuições, e optamos por fazer uma mudança do Libertas 2.0 para o Red Hat 9, com esta pequena mudança conseguimos melhorar o fluxo dos serviços e as demandas dos usuários diminuíram muito.

Ainda assim, havia um grande problema, o acesso ao SERPRO, que nesta época era feito via HOD 6, e o mesmo não tinha suporte ao Java Virtual Machine 2 ou superiores, sendo que os navegadores usados no linux já os utilizavam. Isto além de ter causado transtornos aos usuários que tiveram seus equipamentos migrados e impossibilitando migrar áreas onde o uso dos sistemas baseados no SERPRO é fundamental.

Com muita luta e apoio do ITI, o SERPRO atualizou sua versão de HOD e possibilitou este acesso, que abre novas possibilidades de migração.

Hoje contamos com 37 estações utilizando linux/openoffice e 7 servidores (Postfix, Squid, MySQL, APACHE, LDAP, DNS, DHCP e Firewall).

Mostrando nossa disposição quanto a migração, foi autorizado pelo Diretor da Área Administrativa a participação do maior número de técnicos da área de informática na semana de capacitação promovida pelo ITI. Dos 17 funcionários do Departamento 15 participaram de pelo menos um curso.

A equipe de desenvolvimento está se preparando para programar em PHP, uma equipe que conta com programadores em linguagens Delphi, VB, JAVA e ASP, e já está acertado que os novos desenvolvimentos de sistemas serão utilizando esta linguagem.

Principais problemas enfrentados:

- a equipe técnica não estava preparada para a migração. Havia sido dado só um treinamento de 60 horas, mas não havia ainda pessoas habilitadas para efetuar o suporte;
- a instalação foi efetuada em um final de semana, sem ter sido feito um plano B e sem ter havido treinamento para os usuários, e nem mesmo uma palestra de esclarecimento;
- a distro selecionada para implantação nas máquinas foi o Libertas 2.0 GNU/LINUX da PRODABEL, por falta de conhecimento por parte da equipe técnica da RADIOBRAS. Esta distro apresentou vários problemas por causa de nossa realidade de equipamentos (eram bem mais novos que os usados com a distro em Minas Gerais), o que ocasionou problemas de configuração de equipamentos, principalmente impressoras;
- migração do cliente de e-mail (Xmian – evolution) sem no entanto ter havido a migração do servidor (Exchange), os usuários ficaram sem listas de endereço e contatos, entre outros;
- a versão de openoffice utilizada, ainda apresentava muitas incompatibilidades com os documentos do legado, produzidos com o Microsoft Word;
- não havia a opção de utilizar o RDESKTOP para acesso aos sistemas legados, pois não possuíamos licenças de uso para terminal service nos nossos servidores Windows;
- perda de acesso aos sistemas do SERPRO, pois com a utilização do mozilla que utiliza o Java2 ficamos impossibilitados comunicar com o sistema HOD do SERPRO que só trabalhava com Java1;

Principais avanços:

- a equipe técnica ficou mais coesa com o novo desafio, e aprofundou seus conhecimentos de linux. Aprendeu a compartilhar mais o conhecimento e a pesquisar as soluções aos problemas apresentados;
- a equipe de desenvolvimento está padronizando o a linguagem de novos desenvolvimentos facilitando a manutenção dos sistemas, bem como o desenvolvimento colaborativo;

Marinha do Brasil

Experiência na utilização de Software livre em "mainframe"

No período de um ano, foram iniciados estudos, realizados testes e implantado ambiente Linux no IBM S/390 da Marinha do Brasil, demonstrando a possibilidade de preservação de investimentos já instalados.

Numa 1ª fase, estão sendo mantidos em paralelo, em áreas virtuais distintas, S.O. proprietário e S.O. Linux.

Vantagens:

- compartilhamento de recursos;
- múltiplos ambientes operacionais;
- uso compartilhado e flexibilidade;
- flexibilidade de software e hardware;
- partições lógicas distintas;
- consolidação de servidores para: contingência, desenvolvimento, treinamento etc;

Com isso, o "mainframe" pode ser empregado como servidor corporativo (web, Intranet, e-mail, EAD).

Resultados esperados:

- uso mais racional dos recursos de hardware;
- redução de custos (flexibilidade de configuração);
- interoperabilidade;
- portabilidade das aplicações;
- aumento dos requisitos de segurança com a customização.

Apêndice B - Wine

Wine significa “*Wine Is Not an Emulator*”, e pode-se encontrar detalhes completos em <http://www.winehq.com/>.

B.1. História

O desenvolvimento do Wine começou em torno de 1993, com Bob Armstad, que operava GNU/Linux e Windows na mesma máquina. O *software* GNU/Linux atingiu o ponto de estar capacitado a satisfazer a maior parte de suas necessidades, mas ele *psoftware* livreuía alguns programas de jogos dos quais gostava muito e que só estavam disponíveis em Windows.

Cansado de recarregar o sistema só para executar jogos, começou a trabalhar em um meio de interceptar as chamadas do sistema aos jogos usados e mapeá-los para o ambiente GNU/Linux X. Outras pessoas ouviram falar a respeito do trabalho e começaram a ajudar, até que o Wine tornou-se capaz de executar os jogos que eles queriam jogar.

Em torno de 1995, as pessoas tentavam operar outros programas, inclusive o Quicken e o conjunto Office, e então esses aplicativos tornaram-se a principal área de interesse. Um grupo separou-se, continuando a fornecer suporte às complexas técnicas gráficas usadas nos jogos, mas que não são usadas em geral em aplicativos de estilo escritório. O projeto agora adotou uma abordagem mais formal, com uma equipe com um núcleo de *desenvolvedores* e um líder de projeto. Desde 2000, o projeto estabeleceu-se em bases mais sistemáticas, com um líder de projeto e uma equipe de suporte com base nos Estados Unidos, duas pequenas equipes de desenvolvimento no Canadá e *desenvolvedores* na maioria dos países europeus. Os maiores vendedores também estão contribuindo, por exemplo, a IBM.

Foram criadas técnicas para identificar as chamadas do sistema operacional que o programa fazia. Na maior parte dos casos, o desenvolvimento de um pequeno número de códigos habilitaria um aplicativo a ser operado. Descobriu-se que os programas freqüentemente preparam-se para chamar uma interface particular, porém não fazem a chamada de fato. Então o código foi escrito para permitir aos programas continuarem a fazer essas chamadas preparatórias, sem que ocorram erros imediatos, bem como códigos para dar suporte às chamadas verdadeiras.

O primeiro uso comercial do Wine foi pelo Corel, que trabalhou bastante no suporte do Wine, e usou-o para produzir uma versão nativa GNU/Linux do Wordperfect 8. Outras companhias desde então passaram a usar o Wine para produzir versões GNU/Linux de seus produtos com o mínimo esforço e mudanças *psoftware* livreíveis, sendo que uma das últimas é o Xilinx, que produz pacotes CAD eletrônicos especialistas. É provável que o projeto Ximian Mono use o Wine para permitir que aplicativos .NET escritos para Windows trabalhem sem serem re-escritos. Veja <http://appde.winehq.com/> para detalhes sobre o nível de suporte para uma gama de aplicativos. Recentemente, uma equipe de *desenvolvedores* de aplicativos Windows experiente começou a produzir um conjunto de programas-teste para checar sistematicamente as 12.000+ chamadas de sistema correntes na biblioteca do Windows.

Atualmente, o Wine abrange cerca de 750.000 linhas de código “C” realizando cerca de torno de 90% das chamadas em especificações populares do Windows, como ECMA-234 e Open32. As chamadas não documentadas publicamente são mais difíceis de implementar, mas estão sendo feitos progressos.

Algumas companhias que trabalham com o Wine, desenvolvem códigos para funções particulares, que inicialmente são proprietárias. Elas o fazem para financiarem a si enquanto realizam seu trabalho no projeto principal. E movem seu código para o projeto principal quando obtém a uma fonte de receita alternativa conveniente.

B.2. O que o Wine faz

O Wine intercepta todas as chamadas aos sistemas Windows e DOS junto com as interrupções BIOS, e tenta mapeá-las no ambiente GNU/Linux X. São executadas instruções de processador original como se estivessem no ambiente Windows, portanto, o Wine não é um emulador completo.

O Wine não está amarrado à arquitetura Intel x86 – por exemplo, existe uma versão para o DEC/Compaq Alpha, mas o uso e a demanda séria existem somente no x86. Ele não possibilita a programas Windows x86 serem executados em outra arquitetura, como PowerPC ou SPARC, embora o Wine faça compilações e opere em ambos.

Nem todas as interfaces no ambiente Windows podem ser mapeadas em uma interface nos ambientes GNU/Linux e X. Algumas interfaces Windows simplesmente não possuem software livre ou um equivalente. Isto significa que, em algumas instâncias, há que se escrever uma quantidade significativa de códigos para dar suporte ao mapeamento. Por exemplo, há problemas com os cursores mais complexos usados por alguns programas Windows. O *X Window System* não consegue lidar com mais de duas cores em um cursor, o que significa que o Wine precisa assumir quais cores usar, ocasionalmente com resultados inúteis.

O Wine é atualmente dois produtos, o Wine propriamente dito, que permite operar programas Windows pré-compilados, e o Winelib, que pode ser usado para compilar um programa Windows para produzir um programa GNU/Linux original (isso é o que o Corel usava para produzir a versão GNU/Linux do Wordperfect).

O Winelib pode ser usado para operar programas em outros *hardwares* além do x86, caso o código fonte estiver disponível, embora ainda possam permanecer alguns outros problemas específicos de arquitetura (por exemplo, questões endian).

B.3. Em que o Wine é bom

Há suporte disponível para programas Windows 3.x/95/98/ME/NT (embora o suporte Windows NT seja menos completo). Muitos programas dirigidos ao **Windows 2000** vão operar, a não ser que usem novas interfaces especializadas introduzidas com o **Windows 2000**. Trabalhou-se pouco até agora no suporte específico a programas Windows XP, mas existem poucos deles ainda.

O Wine fornece suporte à maioria das interfaces Windows documentadas publicamente, no entanto, o suporte não é sempre tão completo quanto se gostaria. Veja <http://www.winehq.com/?page=status> para detalhes sobre a situação atual de suporte no Wine.

Programas que operam isolados ou que usam apenas interfaces de comunicação externa, trabalharão normalmente. Cada programa deve ser checado individualmente porque as interfaces precisas e os parâmetros usados podem interagir causando problemas.

Há relatórios de pessoas que operaram compiladores e ambientes de desenvolvimento com algum sucesso.

B.4. Em que o Wine não é bom

Algumas áreas específicas não estão completas, por exemplo a *Dynamic Data Exchange (DDE)*. No entanto, muitos programas fazem chamadas DDE sem usa-las de fato, e portanto funcionarão bastante bem. O OpenGL e outras áreas gráficas especializadas de alta velocidade também apresentam questões. Existe, em parte, a implementação de *Access Control Lists* (como no **Windows NT**), mas ainda não foram integradas com ACLs na base O/S.

O dispositivo VxD de tecnologia de *driver*, introduzido com o **Windows 98**, é uma área difícil. Ele precisa de acesso ao interior do *hardware* e do *kernel* de uma forma que qualquer sistema multiuso sério não permitiria. Há técnicas disponíveis para produzir equivalentes, porém envolvem bastante trabalho, sem garantia de que irão funcionar. Em alguns casos, o vendedor pode ser persuadido a produzir uma versão original GNU/Linux, que usa interface GNU/Linux normal. Como essa arquitetura está sendo abandonada pela *Microsoft* (o tipo de arquitetura do Windows NT não permite tal acesso), isso deixará de ser um problema aos poucos.

Alguns programas Windows tentam manipular dispositivos diretamente (especialmente portas seriais). Isso não é permitido no GNU/Linux, ou qualquer outra variante Unix. Isso normalmente só é aplicável a pacotes de comunicações como o Procomm, e programas derivados de produtos DOS onde tais coisas tinham que ser feitas.

O desenho de algumas imagens gráficas ainda não é satisfatório, especialmente o retoque da fonte True Type. Contudo, trabalha-se ativamente com esse objetivo.

A outra área difícil é o *software* desenvolvido pela própria *Microsoft*. Isso porque esses produtos tendem a usar interfaces não documentadas. Embora seja possível descobrir o que acontece, os *desenvolvedores* devem ser cautelosos, já que as leis relativas à engenharia reversa são muito severas em alguns países. Nos EUA, por exemplo, é proibida a engenharia reversa para qualquer propósito, e a maior parte dos outros países ocidentais permitem somente para o estabelecimento de compatibilidade. Portanto, o trabalho nessa área será sempre um pouco lento.

A operação de instaladores de aplicativos tem sido particularmente problemática, mas trabalhos recentes resolveram a maior parte das dificuldades e esse trabalho continua. Algumas dessas dificuldades são causadas por *desenvolvedores* que não utilizam as técnicas recomendadas. O acesso ao registro é um exemplo disso. O formato do Wine tem um formato diferente do Windows, para facilitar a recuperação. Enquanto as interfaces documentadas forem usadas para acessar o registro, não há problema, porém, às vezes, os desenvolvedores acessam o registro diretamente, sob o risco de corromperem um registro Windows verdadeiro, e isso resulta em que o programa não pode mais trabalhar no Wine.

O Wine é criticado algumas vezes por baixo desempenho, porém isso é freqüentemente devido a seu extenso código de *debugging*. É possível compilar o Wine sem isso, porém deve-se fazê-lo com cuidado, já que significa que os problemas não podem ser diagnosticados sem recompilação adicional.

B.5. Wine – alternativas comerciais

Como mencionado anteriormente, versões prolongadas do Wine estão disponibilizadas como produtos comerciais para dar suporte à corrente principal do Wine. As duas companhias que estão fazendo isso são a Transgaming e a CodeWeavers.

A Transgaming trabalha principalmente no aperfeiçoamento de gráficos e interfaces de som e seu produto visa o mercado de jogos. A CodeWeavers está trabalhando em aplicativos *office* de

tendência dominante, e tem um produto, o Crsoftware livreOver Office, que dá suporte, por exemplo, ao Office e ao Lótus Notes.

B.6. Wine e Visual Basic

Foi reportado que o Visual Basic 3 opera no Wine, porém não há detalhes disponíveis.

O Visual Basic 6 atualmente não é instalável. Está sendo feito um trabalho para resolver esse problema, porém é muito cedo para dizer se será completamente bem sucedido ou não.

Não há informações de tentativas com outras versões.

B.7. Migração de Aplicativo para o Wine

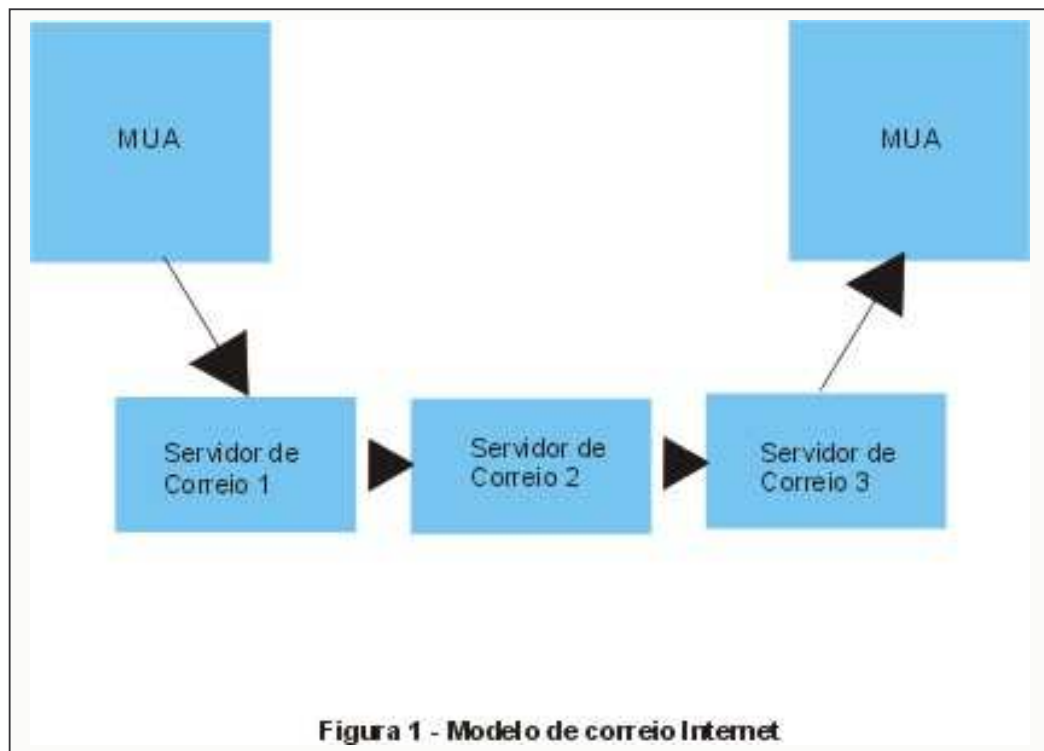
Esta é uma lista de diretrizes gerais para gerir o processo de migração de aplicativos para GNU/Linux no Wine:

1. Checar as condições da licença. Algumas companhias publicaram licenças que proíbem a execução de seu aplicativo, exceto no sistema operacional alvo. Por exemplo, a Oracle costumava fazer isso e a *Microsoft* começou a fazer isso para componentes que podem ser livremente *downloaded*; a situação é fluida. Remova qualquer programa em tais condições da lista teste e faça uma lista deles separada.
2. Obtenha cópias de todos os aplicativos para serem migrados (podem ser todos). Licenças de *sítios* podem não permitir cópias para testes.
3. Configure uma máquina com o último instantâneo do Wine.
4. Teste cada um dos programas da lista teste. Anote todos os problemas encontrados, anote também se eles estão na fase de instalação, inicialização ou execução. Além disso, avalie se eles afetam, na verdade, o que os usuários precisam fazer através de testes, com uma seleção representativa de usuários finais. Anote também o baixo desempenho. O produto disso serão avisos indicando onde as chamadas do sistema ainda não foram implementadas ou estão implementadas de forma incompleta.
5. Para cada programa da lista de problemas, verifique primeiro se já existe uma implementação GNU/Linux. Se houver, não deve haver problemas, mas teste até onde puder. Se não houver implementação GNU/Linux, será necessário contatar o vendedor e sugerir criar uma através do uso do Winelib. Novamente, podem estar faltando DLLs. Há que se lidar com cada vendedor separadamente.
6. Onde os vendedores recusarem-se a cooperar, terão que ser encontrados aplicativos alternativos, ou o projeto deverá ser abandonado.
7. Uma vez disponibilizadas a lista de DLLs extras e de chamadas de biblioteca requeridas, será possível obter um preço pela implementação.
8. Cada programa precisará ser retestado com novos instantâneos de Wine/Winelib até que os problemas todos desapareçam. Remendos às vezes causam problemas com programas que anteriormente estavam operando corretamente, e isso precisa ser testado.
9. O Wine é normalmente compilado com rastreamento *debugging*, e isso atinge o desempenho de forma ruim, especialmente em interações de telas. Quaisquer programas que operem corretamente, mas tenham problemas de desempenho, devem ser re-executados em contraposição a uma cópia do Wine compilado without debug tracing. Se o desempenho ainda for insatisfatório, será necessário um trabalho de desenvolvimento.

Apêndice C – Sistemas de Correio

Este apêndice detalha sistemas de correio em geral, porque a abrangência dos produtos de correio software livre pode, às vezes, ser confusa e a terminologia usada nem sempre é clara.

O Modelo de Correio da Internet é baseado em vários componentes lógicos, cada um dos quais tem um trabalho específico para fazer, e comunica-se com os outros através do uso de protocolos abertos. Este é o modelo usado pelos sistemas software livre. O modelo pode ser melhor descrito com a ajuda de alguns diagramas.



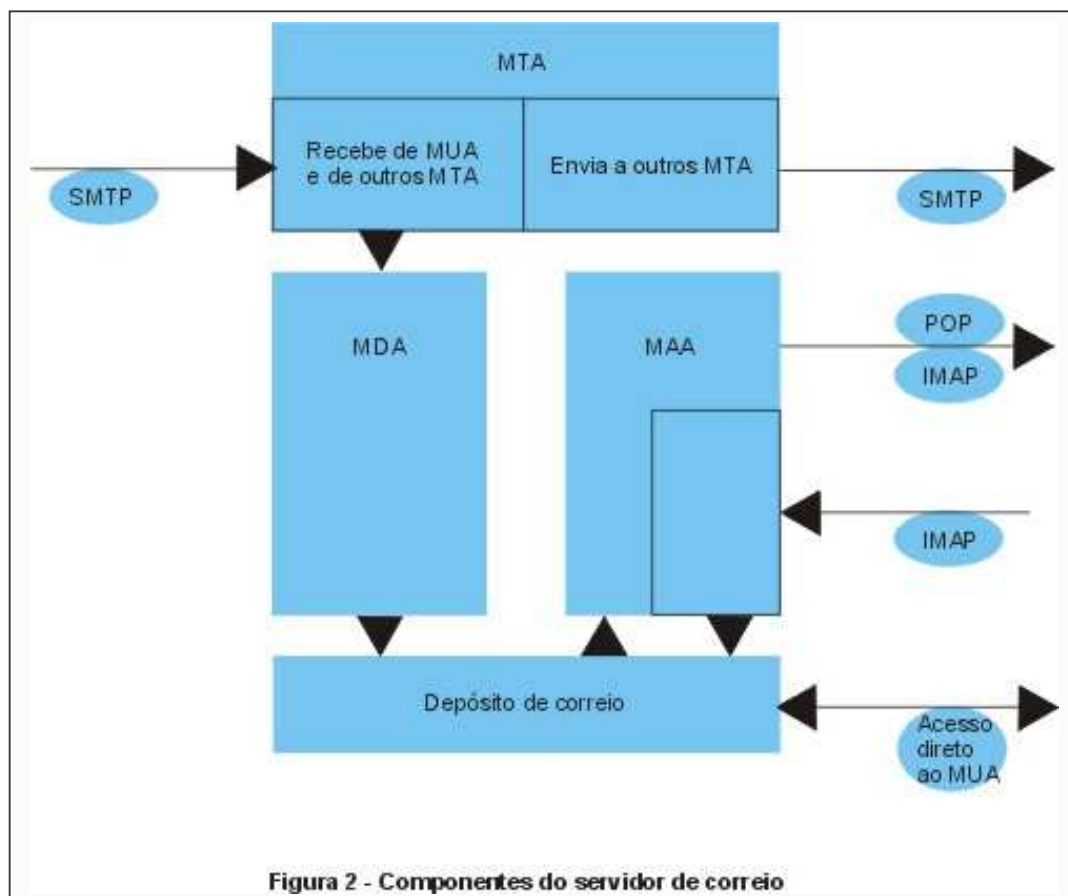
Este diagrama mostra o caminho para a entrega de uma correspondência simples. A correspondência é gerada por um Agente Usuário de Correio (*Mail User Agent/ MUA*). Depois é passado para um servidor de correio, que tem que resolver se pode entregar a correspondência localmente ou se a correspondência deve ser passada para outro servidor. A correspondência é passada de servidor em servidor, até que um deles decida que pode entregar a correspondência localmente, e o faz então. Quando essa entrega estiver completa, a correspondência estará disponível para que um MUA a leia. O MUA final tem a responsabilidade de recuperar a correspondência, bem como de passá-la para uma Interface Usuário de Correio (MUI), para exibi-la para o usuário.

De que forma cada servidor de correio decide entregar a correspondência localmente ou não, poderia ser assunto para um outro capítulo. Em síntese, cada servidor consulta um ou vários arquivos de configuração local, junto com informação de servidores DNS (principalmente os registros MX). Tudo isso é usado então para decidir o que é considerado local. Para a correspondência não-local, o servidor usa a informação para determinar o endereço do próximo servidor de correio ao qual mandar a correspondência.

Cada servidor de correio tem, em geral, a estrutura mostrada na figura 2.

O Agente de Transporte de Correio (*Mail Transport Agent/ MTA*) aceita conexões de outros servidores de correio e MUAs via Protocolo de Transporte de Correio Simples (*Simple Mail Transport Protocol/ SMTP*). Caso a correspondência não seja para entrega local, é então enviada a outro servidor pelo MTA. Se a correspondência for para entrega local, ela é passada para um Agente de Entrega de Correio (*Mail Delivery Agent/ MDA*). O MDA é responsável por armazenar a correspondência na caixa de correio do usuário. A caixa de correio é simplesmente uma forma de armazenar dados, por exemplo, um arquivo, uma série de arquivos separados ou até um banco de dados SQL. A estrutura de armazenamento preciso é definida por aquilo que o MDA suporta. Quando um usuário deseja ver sua correspondência, ele usa um MUA que recupera a correspondência diretamente ou entra em contato com um componente do servidor, que recupera a correspondência da caixa de correio e passa-a para o MUA. Tais componentes do servidor não se encaixam no modelo tradicional MTA/MDA/MUA e nós os chamaremos de Agentes de Acesso ao Correio (*Mail Access Agents/ MAA*). Este termo, no entanto, não é de uso corrente.

O MUA comunica-se com um MAA usando um protocolo aberto, que usualmente é o Protocolo de Correio (*Post Office Protocol/ POP*), ou o Protocolo de Acesso ao Correio Internet (*Internet Mail Access Protocol/IMAP*). O protocolo POP normalmente deleta correspondências da caixa de correio quando estas são passadas ao cliente e o IMAP normalmente as deixa lá. O protocolo IMAP também permite que o MUA altere a caixa de correio, por exemplo, deletando correspondência ou movendo-a de um diretório para outro.



Guia Livre – Referência de Migração para Software Livre do Governo Federal

O MUA pode armazenar correio localmente, na máquina onde está trabalhando. Isso ocorre normalmente quando o POP é usado. Esse armazenamento local então permite que o acesso futuro seja independente do servidor, que é particularmente útil para máquinas que não são permanentemente conectadas à rede. Por outro lado, o IMAP normalmente opera sem cópias locais, mas também pode operar no que chamamos de modo desconectado, que mantém uma cópia local, permitindo ao correio ser manipulado sem uma conexão de rede. Nesse modo, as caixas de correio local e do servidor estão sincronizadas quando é feita uma conexão de rede. Infelizmente, nem todos os MUAs suportam completamente o IMAP desconectado.

Algumas vezes, um programa diferente de um MUA recupera a correspondência e armazena-a localmente para um MUA, sem ter que se conectar ao servidor. Tais programas puxam correio para suas máquinas, em contraste com um MTA padrão, para o qual o correio é empurrado por outros MTAs. Isso pode ser útil se os usuários não desejarem permitir conexões da Internet com suas máquinas, ou estejam operando por trás de uma firewall. Um exemplo de tal programa é o *fetchmail*.

A dificuldade com esse modelo é que os aplicativos disponíveis não se organizam diretamente para ele. Os aplicativos, muito freqüentemente, fazem mais do que uma das funções; por exemplo, o MTA pode incorporar a função MDA, e o *Sendmail* MTA popular pode até ser usado como um MUA em algumas circunstâncias.

Como a correspondência é passada do MUA origem através de vários servidores, até o MUA final, é acrescentada uma série de cabeçalhos, que grava os detalhes da viagem e também controla o processamento da correspondência pelos servidores intermediários e pelo MUA final. Alguns deles são cabeçalhos *Multi-purpose Internet Mail Extension (MIME)*, que são usados para uma série de objetivos de controle, inclusive suporte para conjuntos de caracteres *non-ASCII*, suporte para conteúdo embutido como imagens e suporte para anexos. Quando um MUA anexa um arquivo, ele grava seu tipo como um cabeçalho MIME e então, é responsabilidade do MUA final estar apto a decodificá-lo.

Abaixo discutimos partes desse modelo mais detalhadamente:

C.1. MTA

A maior parte dos MTAs permite ao administrador controlar a aceitação da correspondência em função do remetente. Isso é feito, freqüentemente, limitando-se o número de endereços IP, vindos de conexões SMTP, que o MTA vai aceitar. Isso é extremamente valioso na prevenção de *spammers* que usam o MTA como um *relay* e entopem a largura de banda da rede ao MTA.

Existe um conjunto de mais ou menos 20 extensões para o SMTP chamadas Extended SMTP ou ESMTP. Essas extensões permitem, entre outras coisas, transferência mais rápida de correspondência entre MTAs concordantes, usando a extensão de um canal de informação.

Uma outra extensão possibilita a codificação do *Transport Layer Security (TLS)* entre MTAs concordantes, e uma outra, a *SMTP-AUTH*, permite que os usuários sejam autenticados, usando uma série de técnicas. Ambas as extensões são úteis quando o MTA não permite que um cliente se conecte normalmente porque seu endereço IP está fora do espaço de endereçamento confiável. Isso pode acontecer, por exemplo, se um usuário de laptop digitar de um *sítio* qualquer da Internet. Veja C.4.2 abaixo.

O modelo original assumia que o proprietário de uma conta de correio tinha uma conta de *login* no servidor de correio. Isso significava que o MTA poderia examinar o arquivo de senha local para autenticar os usuários. Esse modelo é muito restritivo e os MTAs modernos precisam dar suporte

Guia Livre – Referência de Migração para Software Livre do Governo Federal

aos Usuários Virtuais onde os detalhes do proprietário da conta são mantidos em um banco de dados, freqüentemente de forma independente dos detalhes da conta de *login* normal. Isso significa que um usuário pode ter uma senha para correio e uma outra para *login*.

O banco de dados pode receber suporte de LDAP, um banco SQL ou um arquivo simples. O MySQL é o servidor de SQL preferido pois é eficiente e rápido no que diz respeito a um aplicativo basicamente “somente para leitura”. O PostgreSQL e o Oracle também podem ser usados.

Um banco de dados com suporte LDAP é recomendado pois oferece suporte melhor para distribuição. Implementações com *default* LDAP freqüentemente usam os produtos *Berkeley Database* da *Sleepycat Systems*.

Algumas vezes, uma máquina pode só conectar-se a um servidor de correio intermitentemente. Isso pode acontecer no caso dos que trabalham em casa ou usuários de laptop, por exemplo. Também pode acontecer em pequenos escritórios, onde o custo de uma conexão permanente não se justifica. Nessas circunstâncias, a central não pode enviar correspondência como faria normalmente, e precisa armazená-la até que seja feita uma conexão. Comentários similares são válidos para o MTA (se houver) na máquina cliente, ou, no caso de um escritório pequeno, o servidor de correio que é *gateway*. Esses MTAs precisam suportar tais situações e são normalmente chamados de *Smart Host* (Anfitriões Inteligentes) quando o fazem.

A distribuição de um *Smart Host* pode ser feita através de SMTP ou POP3.

A distribuição via SMTP é direta e a segurança da máquina receptora pode ser aperfeiçoada através da restrição da conexão para dentro somente a partir do *Smart Host*.

A distribuição via POP3 pode ser feita usando-se o MUA ou o aplicativo *fetchmail*. O *Fetchmail* fará o *download* da correspondência para uma caixa de correio local, como mencionado acima, ou entregará em um MTA local, caso seja requisitado, por exemplo, onde estão envolvidas múltiplas contas de correio.

Ambos os métodos trabalham bem, porém apresentam a desvantagem de não permitirem o uso de “listas negras” para prevenir *spam* de *relays* abertos e de outras fontes indesejáveis. Ferramentas como SpamAssassin podem eliminar a maior parte do spam, mas os custos de processamento são muito mais altos e largura de banda maior é utilizada para descarregar correio destinado ao exame.

C.2. MUA

O MUA e o MUI juntos formam o pacote que a maioria dos usuários consideram “o correio”. Esse é o *software* cliente que opera em um servidor da *web* ou diretamente em estação de trabalho para permitir às pessoas enviar e receber correspondência. É normalmente fornecido algum tipo de armazenamento, de forma que a correspondência possa ser inserida em “pastas” ou “caixas de correio locais” para referência futura.

O MUA lida com protocolos do tipo SMTP para submissão de correspondência e IMAP ou POP para recuperação de correspondência e arquivamento. Ele compreende o formato das mensagens de correio e pode decompor mensagens MIME em suas partes componentes. Onde houver uma requisição de forte segurança “ponta-a-ponta”, o MUA também é responsável pela codificação e assinatura das mensagens. Há dois padrões que competem nesse caso: o S/MIME, que é baseado em certificados X.509, e o PGP/GPG, que é baseado em um formato diferente de certificado com um modelo mais “*web-of-trust*” do que “*hierarchy-of-trust*”.

Guia Livre – Referência de Migração para Software Livre do Governo Federal

A maioria dos MUAS de software livre dá suporte a assinaturas digitais usando o *GNU Privacy Guard (GPG)*. Somente poucos dão suporte a assinaturas S/MIME. Corporações de negócios e Governos optaram pelo padrão S/MIME e seu uso deve, portanto, contar com suporte.

C.3. Armazenagem de Correio

Os sistemas Unix assumiram originalmente que o proprietário de uma conta de correio teria acesso à máquina recebendo o servidor de correio, e que poderia ler um arquivo contendo suas correspondências – ou, alternativamente, que a correspondência seria entregue à máquina normalmente usada pelo usuário para trabalhar. Isso funcionava bem para ambientes com um número pequeno de usuários que também precisavam de uma conta *login* real em uma máquina com um servidor de correio, porém não é prático ou seguro em termos gerais.

O formato original para armazenar correio era um arquivo único por usuário, com as correspondências novas sendo anexadas ao final. Esse arquivo poderia ficar muito grande e a leitura através dele de uma correspondência aleatória seria logo ineficiente. Esse formato é frequentemente conhecido como “*mbox*” e ainda é usado por alguns MUAs, em particular para correio armazenado localmente para o usuário. Foi feita uma alteração em que cada correspondência passou a ser mantida como um arquivo diferente na estrutura do diretório, que permite acesso aleatório mais eficiente. Uma variante dessa estrutura é chamada “*mh*” e uma outra particular com diretórios e procedimentos de acesso definidos é chamada “*maildir*”.

Algumas vezes, essas estruturas eram mantidas no servidor de correio e exportadas aos clientes usando, por exemplo, NFS. Isso permitia que o correio fosse mantido de forma centralizada, o que significava que poderia ser devolvido de forma apropriada, porém introduziu problemas de fechamento com a estrutura de arquivo único. No entanto, o uso do NFS não se tornou popular, talvez pela falta de bons clientes NFS no Windows.

Nem todos os MTAs dão suporte a esses métodos diferentes de acesso direto, por isso a necessidade de MAAs. Um MUA que não pode acessar o depósito de correio diretamente, terá que usar um componente MAA que usa POP ou IMAP.

Tanto o POP3 quanto o IMAP enviam senhas como texto simples por *default*. O IMAP pode usar senhas misturadas, se o MUA suportar. O uso de *links* TLS criptografados é possível, se o MAA e o MUA suportarem, é recomendável em redes locais, e deve ser obrigatório para acesso remoto.

Os MTAs algumas vezes comunicam-se com os MDAs usando o *Local Mail Transport Protocol* ou *LMTP*. A maior parte dos MTAs e MDAs dão suporte ao mesmo.

C.4. Usuários em Movimento

O problema com usuários em movimento é que eles podem conectar-se de endereços IP imprevisíveis, portanto os métodos normais usados pelos MTAs para aceitar correio chegando irá impedi-los de enviar correio via servidor de correio da Administração. Os MTAs devem restringir o acesso de clientes desconhecidos a eles mesmos, para impedir seu uso por *spammers* como *relay* de terceiros.

Há três técnicas gerais disponíveis para contornar esse problema:

C.4.1. Redes Privadas Virtuais (Virtual Private Networks - VPNs)

Guia Livre – Referência de Migração para Software Livre do Governo Federal

Num VPN, à máquina remota pode ser alocado um endereço que pode ser incluído no espaço confiado aos MTA. O problema é que todo o acesso para a rede interna ficará disponível a qualquer pessoa que acesse a máquina remota, um risco significativo com laptops, a não ser que as chaves de acesso sejam codificadas com uma senha registrada toda vez que a conexão for iniciada. Infelizmente, o problema é que os usuários algumas vezes configuram suas máquinas para lembrar as senhas.

C.4.2. SMTP-AUTH e TLS

A extensão SMTP-AUTH para o SMTP permite que um MTA seja configurado para requisitar uma senha para autenticar o usuário remoto. Os principais métodos de autenticação são PLAIN, LOGIN e CRAM-MD5.

O PLAIN requer que a senha seja mantida em aberto no cliente, mas possa ser codificada no servidor. Se a conexão SMTP não estiver codificada, então a senha será passada em aberto (embora em base-64) pela rede.

O LOGIN é menos eficiente do que o PLAIN, pois requer três interações de rede em vez de uma e, como no PLAIN, o nome de usuário e a senha viajam em aberto pela rede.

O CRAM-MD5 codifica o nome de usuário e a senha quando eles passam pela rede. No entanto, a senha deve ser mantida in texto simples no cliente e no servidor. Ele requer somente duas interações de rede.

Nem todos os MUA's suportam SMTP-AUTH e os que o fazem, podem suportar um número limitado de métodos. Por exemplo, o Outlook Express usa o LOGIN.

Comparando-se ao uso de um VPN, o único acesso habilitado é o de enviar correspondência, assim outros serviços não estarão comprometidos se a máquina remota for roubada.

O ESMTP também permite que uma sessão TLS seja negociada entre cliente e servidor. Essa conexão codifica dados na rede e também pode autenticar a máquina cliente. A autenticação requer um certificado de cliente que confira com aquele existente no servidor.

C.4.3. POP-before-SMTP.

Este método tira vantagem do fato de que os protocolos POP e o IMAP demandam autenticação da senha. Após uma conexão POP ou IMAP bem sucedida para ler o correio, o MAA mantém um *login* autenticado com o endereço IP do cliente, a data e a hora. Quando o cliente tenta enviar correspondência que não seja para o domínio local, o MTA verifica se o endereço IP do cliente está em seu espaço de endereçamento confiável. Se não estiver, ele verifica o banco de dados de *logins* autenticados para o endereço IP. Caso não haja *login* autenticado registrado do endereço IP do cliente, ou caso a última conexão autenticada não tenha acontecido recentemente, o MTA recusa a transmissão da mensagem. O período de tempo é configurável e o *default* típico é de 20 minutos. Esse método requer que o MAA e o MTA cooperem. Por essa razão, nem todas as combinações funcionam. Esse método tem a desvantagem de que os usuários precisam verificar se há correspondência chegando primeiro. Alguns usuários podem considerar isso difícil, a não ser que o MTA o faça automaticamente para eles.

C.5. Desempenho

Em geral, um MTA usa pouco poder de processamento; anfitriões que só operam isso são usualmente limitados pela largura de banda ou pelo desempenho do disco. Os servidores do IMAP e

Guia Livre – Referência de Migração para Software Livre do Governo Federal

do POP requerem maior poder de processamento e o IMAP requer um pouco mais de RAM do que o POP. Contudo, é provável que nenhum deles seja problema para o hardware atual.

Os *scanners* antivírus requerem bastante RAM e poder de processamento, especialmente se forem permitidos anexos MIME.

Mesmo assim, os limites de desempenho são normalmente estabelecidos pelo tráfego e não pelo número de contas.

Abaixo estão alguns exemplos de desempenho de servidores de correio. Foram incluídos para dar alguma idéia do que é requerido:

Sítio 1 - 2 x Pentium III Xeon 2.4G, 4 Gb RAM, 3 x 36GB SCSI Raid 5

Usuários virtuais com consulta em MySQL.

Postfix 2.0.6, Courier-IMAP 1.7, MySQL 4.12, RAV-Antivirus, Mailman 2.1, Red Hat

Linux 8.0, sem SSL.

Em torno de 4,800 usuários.

Sítio 2 - Athlon 1200, 1 Gb RAM, RAID5

Postfix + Courier-IMAP (anti-virus *scan* em outra máquina), sem SSL.

8,500 Usuários.

Sítio 3 - Pentium 133, 40 Mb RAM, IDE disk

Debian GNU/Linux, Courier-MTA + Courier-IMAP + SpamAssassin (o último somente para um usuário).

Típicamente usuários 18 POP3 e 7 usuários IMAP a qualquer tempo.

Processador em torno de 20% ocupado.

Sítio 4 - dual Pentium II 450 Xeon, 256 Mb RAM

MySQL, Courier-MTA, Courier-IMAP, sqwebmail, SSL.

50 usuários, principalmente POP3.

Sítio 5 - Pentium II 400 with 256M RAM

Courier-MTA + SpamAssassin, Red Hat Linux 8.0.

300 caixas de correio, em torno de 4,000 mensagens por dia.

Sítio 6 - Pentium III 677Mhz, 512Mb RAM, 2 x IDE disk

FreeBSD 4.7, Exim 4.05, OpenLDAP 2.1.5, Cyrus 2.1.11, Mailman 2.1, Apache 1.3.26.

A máquina é principalmente um *webserver* ocupado, porem ele também lida com muitos milhares de correspondências por dia sem qualquer carga adicional perceptível.

Apêndice D – Glossário

ACL	Access Control List. Uma Lista de Controle de Acesso é uma lista anexa a um objeto, tal como um arquivo. Consiste de expressões de controle, cada um das quais concede ou nega alguma capacidade a um usuário particular ou a um grupo de usuários.
Administration	Uma administração pública européia.
Administrators	A gerência de TI de uma Administração.
API	Application Programming Interface. O método específico recomendado por um sistema operacional de computador, aplicativo ou ferramenta de terceiros, pelo qual um programador escrevendo um aplicativo pode fazer requisições do sistema operacional. Também conhecido por Application Programmers Interface.
ASP	Active Server Pages. Uma página HTML que inclui um ou mais scripts (pequenos programas embutidos) que são processados em um servidor da <i>Web Microsoft</i> antes da página ser enviada para o usuário. Um ASP é, de alguma forma, similar à abrangência de um <i>serverside</i> ou um aplicativo de interface de <i>gateway</i> comum (Common Gateway Interface - CGI) em que tudo envolve programas que operam no servidor, normalmente talhando uma página para o usuário.
BDC	Backup Domain Controller. Papéis que podem ser designados a um servidor em uma rede de computadores que usa o sistema operacional <i>Windows NT</i> . O <i>Windows NT</i> usa a idéia de um domínio para gerir acesso a um conjunto de recursos de rede (aplicativos, impressoras, etc) para um grupo de usuários. O usuário precisa somente conectar-se ao domínio para ganhar acesso aos recursos, que podem estar localizados em vários servidores diferentes na rede. Um servidor, conhecido como controlador de domínio primário, gerencia o banco de dados master do usuário para o domínio. Um ou mais outros servidores são designados como controladores do domínio de <i>backup</i> . O controlador de domínio primário envia cópias do banco de dados, periodicamente, aos controladores de domínio do <i>backup</i> . Um controlador de domínio de <i>backup</i> pode intervir como controlador de domínio primário, caso o servidor de PDC falhe e também pode ajudar a equilibrar a carga de trabalho se a rede estiver muito ocupada.
Beta Code	Quando o <i>software</i> é escrito, ele passa por vários estágios diferentes antes de ser considerado suficientemente livre de erros e funcionalmente correto para ser usado como <i>software</i> de produção. O primeiro desses estágios é chamado de alfa e o segundo de beta. Beta Code é, portanto, um código substancialmente correto, porém ainda pode conter erros significativos. Deve, portanto, ser usado com cautela.
Binaries	O <i>software</i> é comumente escrito em uma linguagem facilmente compreensível pelas pessoas, que é chamada de Código Fonte. Esse código é convertido para uma fórmula diretamente entendida pelo processador do

computador. Esse código é chamado de Binário porque consiste de uma fileira de zeros e uns. Essa é a fórmula na qual a maior parte dos códigos proprietários são entregues e é muito difícil converter de volta para uma fórmula facilmente compreensível pelas pessoas. O código fonte permite que sejam feitas alterações no *software* e também permite que as pessoas entendam o que ele está fazendo.

**Concurrent User
Licence.**

Uma forma de licença que cobra em base ao maior número de usuários que podem acessar um aplicativo ao mesmo tempo.

Boilerplate Entries

Qualquer padrão ou partes indiferenciadas de dados, usualmente coisas que devem estar presentes, porém não são de grande interesse.

CIL

Common Intermediate Language. Código intermediário independente do compilador e da máquina, que vai operar por um Common Language Runtime or CLR. Esse código pode ser obtido de várias linguagens, inclusive C# e C. Ambos, CIL e CLR, são parte do CLI ou Common Language Infrastructure.

Daemon

Um programa ligado a um sistema ou processo, aguardando para desempenhar sua tarefa, até ser acionado por um outro processo ou evento.

DBMS

Database Management System. Um programa que permite que um ou mais usuários de computador acessem dados em um banco de dados. O DBMS gerencia requisições de usuários (e requisições de outros programas), de forma que usuários e outros programas fiquem livres de ter que entender onde os dados são mantidos fisicamente em mídia de armazenamento e, em um sistema multi-uso, quem mais pode estar acessando os dados.

DEC Protocol

The Digital Equipment Corporation or DEC created a set of protocols for controlling terminal devices. These protocols have become widely used and are now standards.

DHCP

Dynamic Host Configuration Protocol. Um protocolo de comunicação que permite aos administradores da rede gerenciar de forma central e automatizar a designação dos endereços do Internet Protocol (IP) na rede de uma organização.

Distribution

Para *softwares* de fonte aberta como o Linux, companhias como a Red Hat especializam-se em pacotes de componentes de muitas fontes juntos em um único pacote ou um conjunto de pacotes, que podem ser distribuídos convenientemente para usuários com um único download ou em um conjunto de CDs.

DNS

Domain Name Server. Usado para conversão entre o nome da máquina na Internet e seu endereço numérico.

**Domain
(Authentication)**

Um conjunto de identificadores de autorização (pessoas e processos) geridos por um servidor de autenticação.

Guia Livre – Referência de Migração para Software Livre do Governo Federal

Free Software Está definido em <http://www.gnu.org/philosophy/free-sw.html> .

FTP	File Transfer Protocol. Um meio independente do sistema de transferir arquivos entre sistemas conectados via TCP/IP. Garante que o arquivo foi transferido corretamente, mesmo que tenha havido erros durante a transmissão.
Gopher Services	Sistema antigo de recuperação de informação do tipo hipertexto.
GPL	General Public License do GNU.
Green Screen	Um terminal ou monitor que só é capaz de exibir caracteres de tamanho fixo e (possivelmente) gráficos de blocos simples. O nome vem do fato de que muitas telas de monitores <i>mainframe</i> dos anos 1970 e 1980 usavam fósforo verde.
GUI	Graphical User Interface.
Hashes	Um <i>Hash</i> é um identificador <i>short-form</i> único, uma “impressão digital” de algo mais complicado. <i>Hashes</i> são produzidos usando-se funções matemáticas <i>one-way</i> . São usados em sistemas de banco de dados e em sistemas de segurança e de codificação.
HTTP	Hypertext Transfer Protocol. Um conjunto de regras para troca de arquivos (texto, gráficos, imagens, som, vídeo, e outros arquivos multimídia) na <i>World Wide Web</i> . Com relação ao jogo de protocolos TCP/IP (que são a base para troca de informação na Internet), o HTTP é um protocolo de aplicativo.
Java Applet	Um mini-programa de <i>software</i> que um <i>navegador</i> habilitado por Java ou ActiveX usa e do qual faz o <i>download</i> automaticamente. Pode acrescentar suporte sofisticado a páginas da <i>Web</i> , muito além de programações como DHTML ou Javascript.
Java Servlet	um programa Java que opera como parte de um serviço de rede, tipicamente em um servidor HTTP e responde a necessidades de clientes. O uso mais comum de um <i>servlet</i> é de estender um servidor da <i>web</i> através da geração de conteúdo <i>web</i> dinamicamente. Por exemplo, um cliente pode precisar de informação de um banco de dados; pode ser escrito um <i>servlet</i> que receba o pedido, consiga e processe os dados da forma como o cliente necessita e retorne o resultado para o cliente.
JDBC	Java Database Connectivity. Uma especificação de interface de programa aplicativo (application program interface - API) para conectar programas escritos em Java aos dados em bancos de dados populares. A interface de programa aplicativo permite que se codifique declarações de requisição de acesso em <i>Structured Query Language (SQL)</i> , as quais são então passadas para o programa que gerencia o banco de dados. O resultado é retornado através de uma interface similar.
Kernel	O núcleo de um sistema operacional que lida com tarefas como alocação de memória, dispositivo <i>input</i> e <i>output</i> , alocação de processo, segurança e acesso ao usuário.

LDAP	Lightweight Directory Access Protocol. Um protocolo de <i>software</i> para habilitar qualquer pessoa a localizar organizações, indivíduos e outros recursos como arquivos e dispositivos em uma rede, seja em rede (Internet) pública ou em intranet interna. O LDAP é uma versão “leve” (com pouco volume de códigos) do Directory Access Protocol (DAP), que é parte do X.500, um padrão para serviços de diretório em uma rede.
LGPL	Lesser General Public License do GNU
Load Balancing	Equilibrar a carga é dividir o volume de trabalho que um computador tem para fazer entre dois ou mais processadores ou computadores, de forma que seja feito um volume maior de trabalho no mesmo tempo e, em geral, todos os usuários são mais bem servidos. O equilíbrio de carga pode ser implementado com hardware, <i>software</i> ou uma combinação dos dois. Tipicamente, o equilíbrio de carga é a principal razão agrupamento de servidores de computadores.
MAA	Mail Access Agent. Um termo usado neste relatório para descrever o componente de correio do servidor que gerencia o acesso ao depósito de correio por um MUA. São exemplos os servidores do POP e do IMAP. Veja Apêndice C página 108 acima para uma discussão completa.
MDA	Mail Delivery Agent. Um componente de correio que aceita correio de um MTA e o devolve ao depósito de correio.
Metadata	Uma definição ou descrição de dados.
MTA	Mail Transport Agent. Esse é o componente de correio que tem a responsabilidade de decidir se a correspondência entregue a ele é para uma conta local ou não. Ele passa a correspondência local para um MDA ou armazena-a direto no mailstore. O correio remoto é passado para um outro MTA.
MUA	Mail User Agent. O componente de correio do cliente, o qual recupera o correio do mailstore e apresenta-o para o usuário. Ele permite ao usuário criar novas correspondências e manda-las ao MTA para serem transmitidas. O MUA sera frequentemente associado a uma interface gráfica.
.NET	Conjunto de tecnologias de <i>software</i> da <i>Microsoft</i> para conectar informação, pessoas, sistemas e dispositivos. É baseado em serviços da <i>web</i> , que são pequenos aplicativos que podem conectar-se uns aos outros, bem como a outros aplicativos maiores na Internet. O projeto Mono software livre é uma implementação da estrutura de desenvolvimento .NET.
NFS	Network File Service. Um protocolo usado comumente pelo Unix como sistema para acessar arquivos mantidos em sistemas remotos, como se fossem locais.
ODBC	Open Database Connectivity. Uma interface de programação de aplicativo de padrão aberto (<i>application programming interface - API</i>) para acessar um banco de dados. Usando relatórios ODBC em um programa,

pode-se acessar arquivos em vários bancos de dados diferentes, inclusive Access, dBase, DB2, Excel, e Text. Além do *software* ODBC, é necessário um módulo ou *driver* separado para cada banco de dados a ser acessado.

Open Relay	Um <i>relay</i> aberto (algumas vezes chamado de um <i>relay</i> inseguro ou <i>third-party relay</i>) é um servidor de correio SMTP que permite a transmissão de mensagens de correio por terceiros. Através do processamento de correio que não é para usuário local, nem de usuário local, um <i>relay</i> aberto possibilita que um remetente inescrupuloso indique a rota de grandes volumes de <i>spam</i> . Na verdade, o proprietário do servidor – que, tipicamente, não tem conhecimento do problema – fornece rede e recursos de computador para os objetivos do remetente. Além dos custos financeiros incorridos quando um <i>spammer</i> seqüestra um servidor, uma organização pode também sofrer travamentos do sistema, danos a equipamentos e perda de negócios.
Open Source Software	Está definido em http://www.opensource.org/docs/definition_plain.html .
OSS	Veja <i>Open Source Software</i> .
PDA	Personal Digital Assistant. Um computador de mão eletrônico.
PDC	Primary Domain Controller. Veja <i>Backup Domain Controller (BDC)</i> .
PHP	PHP: Hypertext Preprocessor. Uma linguagem e um intérprete criptografados disponibilizados gratuitamente e usados primariamente em servidores Linux <i>Web</i> . O PHP é uma alternativa à tecnologia <i>Active Server Page (ASP)</i> da <i>Microsoft</i> . Como o ASP, o script do PHP é embutido dentro de página da <i>web</i> junto de seu HTML. Antes de a página ser enviada a um usuário que a tenha solicitado, o servidor da <i>web</i> chama o PHP para interpretar e desempenhar as operações solicitadas no script PHP.
PKI	Public Key Infrastructure. Uma PKI possibilita que usuários de uma rede pública insegura como a Internet, troquem dados e dinheiro de forma segura e privada, através do uso de um par de chaves criptográficas, que é obtido e partilhado através de uma autoridade confiável. A infraestrutura de chave pública fornece um certificado digital que pode identificar um indivíduo ou uma organização, e serviços de diretório que podem armazenar e, quando necessário, revogar os certificados.
Potential User Licence	Uma forma de licença que cobra com base no número máximo de usuários que tem capacidade de acessar um aplicativo.
Protocol	Um conjunto especial de regras usado em uma conexão de telecomunicação, entre pontos finais. Existem protocolos em vários níveis em uma conexão de telecomunicação. Há protocolos de <i>hardware</i> de telefonia. Há protocolos entre cada uma das várias camadas funcionais e cada camada correspondente do outro lado de uma comunicação. Ambos os pontos finais precisam reconhecer e cumprir os preceitos do protocolo. Os protocolos são freqüentemente descritos em uma indústria ou padrão internacional.

Proxy Server	Um servidor que atua como intermediário entre um usuário de uma estação de trabalho e a Internet, de forma que a empresa possa garantir segurança, controle administrativo e serviço de caching. Um servidor proxy é associado a um servidor de gateway ou a parte dele, que separa a rede da empresa da rede de fora e um servidor firewall, que protege a rede da empresa da intromissão de fora.
Scenario	Veja Seção 4 para uma definição.
Esquema	A organização ou estrutura para um banco de dados. A atividade de modelar dados conduz a um esquema.
Session Manager	Quando um usuário se liga a um computador, cria-se uma sessão que consiste de um ambiente completo de informação de controle pessoal para eles, uma série de processos. O <i>manager</i> permite que o usuário mude esse ambiente e pode também salva-lo de forma que o próximo usuário, ao ligar-se ao computador, volte à situação em que estava antes de se desligar pela última vez.
Smart Card	Um cartão plástico que contem um <i>chip</i> de computador. O cartão é usado para desempenhar operações que requerem os dados que estão armazenados no <i>chip</i> .
SMB	Server Message Block. Este é o protocolo usado na rede Windows para permitir que recursos como arquivos de uma máquina sejam compartilhados em outras máquinas como se fossem locais.
SMS	Short Message Service. Um serviço para enviar mensagens de até 160 caracteres (224 caracteres se estiver usando um modo 5-bit) para telefones móveis que usam comunicação Global System for Mobile (GSM).
Source Code	Veja Binaries.
SQL	Structured Query Language. Uma linguagem de programação e interativa padrão para obter informação de um banco de dados e para atualiza-lo. Embora o SQL seja padrão ANSI e ISO, muitos produtos de banco de dados suportam o SQL com extensões proprietárias para a linguagem padrão. As <i>queries</i> assumem o formato de uma linguagem comando que permite selecionar, inserir, atualizar e encontrar o local dos dados, e assim por diante. Também há uma interface de programação.
SSL	Secure Sockets Layers. Um protocolo comumente usado para gerenciar a segurança de uma transmissão de mensagem na Internet. O SSL foi sucedido recentemente pelo <i>Transport Layer Security (TLS)</i> , que é baseado no SSL. O SSL é incluído como parte dos <i>navegadores</i> da <i>Microsoft</i> e da <i>Netscape</i> e da maior parte dos produtos servidores da <i>Web</i> .
Stored Procedure	Um conjunto de instruções do <i>Structured Query Language (SQL)</i> com um nome designado que é armazenado no banco de dados de forma compilada de forma que possa ser compartilhado por vários programas .

Trigger	Um conjunto de instruções do <i>Structured Query Language (SQL)</i> que dispara automaticamente uma ação, quando uma operação específica ocorre, como a mudança de dados em uma tabela.
TLS	Transport Layer Security. Uma camada que provê serviços de criptografia e autenticação que podem ser negociados durante a fase inicial de muitos protocolos da Internet (e.g. SMTP, LDAP, IMAP, POP3). O TLS é derivado do SSL e usa os mesmos certificados, mas não requisita que cada serviço receba um novo número de porta; veja SSL.
VMS	Um sistema operacional desenvolvido pela Digital Equipment Corporation (DEC) para usar em seus minicomputadores VAX. Posteriormente transferido para o sistema Alpha 64-bit. Um dos principais designers do VMS, posteriormente desenhou o kernel do Windows NT.
WebDAV	World Wide Web Distributed Authoring and Versioning. O padrão <i>Internet Engineering Task Force (IETF)</i> para autoria colaborativa na <i>Web</i> : um conjunto de extensões para o <i>Hypertext Transfer Protocol (HTTP)</i> que facilita a edição colaborativa e a gestão de arquivos entre usuários localizados remotamente uns dos outros na Internet.
Window Manager	Em um ambiente gráfico moderno um usuário é apresentado a uma série de janelas onde operam processos. Isso significa que eles podem operar muitas coisas diferentes ao mesmo tempo e ter o resultado exibido na tela simultaneamente. O papel do <i>window manager</i> é gerir essas janelas. Ele tem que manter a pista da janela na qual o usuário está interessado no momento, permitir ao usuário trocar janelas e criar e destruir janelas. Ele também controla a forma em que as janelas aparecem, seu formato e características de controle.
XML	Extensible Markup Language. Uma forma flexível para formatos de informação e para partilhar o formato e os dados na <i>World Wide Web</i> , nas <i>intranets</i> , e em qualquer outro lugar. O XML é uma recomendação formal do <i>World Wide Web Consortium (W3C)</i> , similar à linguagem das páginas da <i>Web</i> atuais, o <i>Hypertext Markup Language (HTML)</i> .
X Session	Quando um usuário se liga a um computador e opera programas no protocolo X, ele cria uma <i>X session</i> .
X Terminal	Um terminal especialmente desenhado para operar um servidor X, que permite aos usuários exibir o produto de programas operando em outro computador, usando o protocolo X em uma rede.

Apêndice E – Licença CC-GPL



Licença Pública Geral do GNU (GPL) [General Public License]

Versão 2, Junho de 1991 Direitos Autorais Reservados © 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite [conjunto] 330, Boston, MA [Massachusetts] 02111-1307 USA [Estados Unidos da América]

É permitido a qualquer pessoa copiar e distribuir cópias sem alterações deste documento de licença, sendo vedada, entretanto, qualquer modificação.

Introdução

As licenças da maioria dos softwares são elaboradas para suprimir sua liberdade de compartilhá-los e modificá-los. A Licença Pública Geral do GNU, ao contrário, visa garantir sua liberdade de compartilhar e modificar softwares livres para assegurar que o software seja livre para todos os seus usuários. Esta Licença Pública Geral é aplicável à maioria dos softwares da [Free Software Foundation](#) [Fundação do Software livre] e a qualquer outro programa cujos autores se comprometerem a usá-la. (Em vez dela, alguns outros softwares da Free Software Foundation são cobertos pela Licença Pública Geral de Biblioteca do GNU). Você também poderá aplicá-la aos seus programas.

Quando falamos de software livre, estamos nos referindo à liberdade, não ao preço. Nossas Licenças Públicas Gerais visam garantir que você tenha a liberdade de distribuir cópias de software livre (e cobrar por isso se desejar), que receba código-fonte ou possa obtê-lo se desejar, que possa modificá-lo ou usar partes dele em novos programas livres; finalmente, que você tenha ciência de que pode fazer tudo isso.

Para proteger seus direitos, necessitamos fazer restrições que proíbem que alguém negue esses direitos a você ou que solicite que você renuncie a eles. Essas restrições se traduzem em determinadas responsabilidades que você deverá assumir, se for distribuir cópias do software ou modificá-lo.

Por exemplo, se você distribuir cópias de algum desses programas, tanto gratuitamente como mediante uma taxa, você terá de conceder aos receptores todos os direitos que você

Guia Livre – Referência de Migração para Software Livre do Governo Federal

possui. Você terá de garantir que, também eles, recebam ou possam obter o código-fonte. E você terá a obrigação de exibir a eles esses termos, para que eles conheçam seus direitos.

Protegemos seus direitos através de dois passos: (1) estabelecendo direitos autorais sobre o software e (2) concedendo a você esta licença, que dá permissão legal para copiar, distribuir e/ou modificar o software.

Além disso, para a proteção de cada autor e a nossa, queremos ter certeza de que todos entendam que não há nenhuma garantia para este software livre. Se o software for modificado por alguém e passado adiante, queremos que seus receptores saibam que o que receberam não é o original, de forma que quaisquer problemas introduzidos por terceiros não afetem as reputações dos autores originais.

Finalmente, qualquer programa livre é constantemente ameaçado por patentes de software. Queremos evitar o risco de que redistribuidores de um programa livre obtenham individualmente licenças sob uma patente, tornando o programa, com efeito, proprietário. Para impedir isso, deixamos claro que qualquer patente deve ser licenciada para o uso livre por parte de qualquer pessoa ou, então, simplesmente não deve ser licenciada.

Os exatos termos e condições para cópia, distribuição e modificação seguem abaixo.
TERMOS E CONDIÇÕES PARA CÓPIA, DISTRIBUIÇÃO E MODIFICAÇÃO.

1. Esta Licença se aplica a qualquer programa ou outra obra que contenha um aviso inserido pelo respectivo titular dos direitos autorais, informando que a referida obra pode ser distribuída em conformidade com os termos desta Licença Pública Geral. O termo "Programa", utilizado abaixo, refere-se a qualquer programa ou obra, e o termo "obras baseadas no Programa" significa tanto o Programa, como qualquer obra derivada nos termos da legislação de direitos autorais: isto é, uma obra contendo o Programa ou uma parte dele, tanto de forma idêntica como com modificações, e/ou traduzida para outra linguagem. (Doravante, o termo "modificação" inclui também, sem reservas, a tradução). Cada licenciado, doravante, será denominado "você".

Outras atividades que não a cópia, distribuição e modificação, não são cobertas por esta Licença; elas estão fora de seu escopo. O ato de executar o Programa não tem restrições e o resultado gerado a partir do Programa encontra-se coberto somente se seu conteúdo constituir uma obra baseada no Programa (independente de ter sido produzida pela execução do Programa). Na verdade, isto dependerá daquilo que o Programa faz.

2. Você poderá fazer cópias idênticas do código-fonte do Programa ao recebê-lo e distribuí-las, em qualquer mídia ou meio, desde que publique, de forma ostensiva e adequada, em cada cópia, um aviso de direitos autorais (ou copyright) apropriado e uma notificação sobre a exoneração de garantia; mantenha intactas as informações, avisos ou notificações referentes a esta Licença e à ausência de qualquer garantia; e forneça a quaisquer outros receptores do Programa uma cópia desta Licença junto

com o Programa.

Você poderá cobrar um valor pelo ato físico de transferir uma cópia, e você pode oferecer, se quiser, a proteção de uma garantia em troca de um valor.

3. Você poderá modificar sua cópia ou cópias do Programa ou qualquer parte dele, formando, dessa forma, uma obra baseada no Programa, bem como copiar e distribuir essas modificações ou obra, de acordo com os termos da Cláusula 1 acima, desde que você também atenda a todas as seguintes condições:
 - a. Você deve fazer com que os arquivos modificados contenham avisos, em destaque, informando que você modificou os arquivos, bem como a data de qualquer modificação.
 - b. Você deve fazer com que qualquer obra que você distribuir ou publicar, que no todo ou em parte contenha o Programa ou seja dele derivada, ou derivada de qualquer parte dele, seja licenciada como um todo sem qualquer custo para todos terceiros nos termos desta licença.
 - c. Se o programa modificado normalmente lê comandos interativamente quando executado, você deverá fazer com que ele, ao começar a ser executado para esse uso interativo em sua forma mais simples, imprima ou exiba um aviso incluindo o aviso de direitos autorais (ou copyright) apropriado, além de uma notificação de que não há garantia (ou, então, informando que você oferece garantia) e informando que os usuários poderão redistribuir o programa de acordo com essas condições, esclarecendo ao usuário como visualizar uma cópia desta Licença. (Exceção: se o Programa em si for interativo mas não imprimir normalmente avisos como esses, não é obrigatório que a sua obra baseada no Programa imprima um aviso).

Essas exigências se aplicam à obra modificada como um todo. Se partes identificáveis dessa obra não forem derivadas do Programa e puderem ser consideradas razoavelmente como obras independentes e separadas por si próprias, nesse caso, esta Licença e seus termos não se aplicarão a essas partes quando você distribuí-las como obras separadas. Todavia, quando você distribuí-las como parte de um todo que constitui uma obra baseada no Programa, a distribuição deste todo terá de ser realizada em conformidade com esta Licença, cujas permissões para outros licenciados se estenderão à obra por completo e, conseqüentemente, a toda e qualquer parte, independentemente de quem a escreveu.

Portanto, esta cláusula não tem a intenção de afirmar direitos ou contestar os seus direitos sobre uma obra escrita inteiramente por você; a intenção é, antes, de exercer o direito de controlar a distribuição de obras derivadas ou

obras coletivas baseadas no Programa.

Além do mais, a simples agregação de outra obra que não seja baseada no Programa a ele (ou a uma obra baseada no Programa) em um volume de mídia ou meio de armazenamento ou distribuição, não inclui esta outra obra no âmbito desta Licença.

4. Você poderá copiar e distribuir o Programa (ou uma obra baseada nele, de acordo com a Cláusula 2) em código-objeto ou formato executável de acordo com os termos das Cláusulas 1 e 2 acima, desde que você também tome uma das providências seguintes:
 - a. Incluir o código-fonte correspondente completo, passível de leitura pela máquina, o qual terá de ser distribuído de acordo com as Cláusulas 1 e 2 acima, em um meio ou mídia habitualmente usado para intercâmbio de software; ou,
 - b. Incluir uma oferta por escrito, válida por pelo menos três anos, para fornecer a qualquer terceiro, por um custo que não seja superior ao seu custo de fisicamente realizar a distribuição da fonte, uma cópia completa passível de leitura pela máquina, do código-fonte correspondente, a ser distribuído de acordo com as Cláusulas 1 e 2 acima, em um meio ou mídia habitualmente usado para intercâmbio de software; ou,
 - c. Incluir as informações recebidas por você, quanto à oferta para distribuir o código-fonte correspondente. (Esta alternativa é permitida somente para distribuição não-comercial e apenas se você tiver recebido o programa em código-objeto ou formato executável com essa oferta, de acordo com a letra b, acima).

O código-fonte de uma obra significa o formato preferencial da obra para que sejam feitas modificações na mesma. Para uma obra executável, o código-fonte completo significa o código-fonte inteiro de todos os módulos que ela contiver, mais quaisquer arquivos de definição de interface associados, além dos scripts usados para controlar a compilação e instalação do executável. Entretanto, como uma exceção especial, o código-fonte distribuído não precisa incluir nada que não seja normalmente distribuído (tanto no formato fonte como no binário) com os componentes principais (compilador, kernel e assim por diante) do sistema operacional no qual o executável é executado, a menos que este componente em si acompanhe o executável.

Se a distribuição do executável ou código-objeto for feita mediante a permissão de acesso para copiar, a partir de um local designado, então, a permissão de acesso equivalente para copiar o código-fonte a partir do

mesmo local será considerada como distribuição do código-fonte, mesmo que os terceiros não sejam levados a copiar a fonte junto com o código-objeto.

5. Você não poderá copiar, modificar, sublicenciar ou distribuir o Programa, exceto conforme expressamente estabelecido nesta Licença. Qualquer tentativa de, de outro modo, copiar, modificar, sublicenciar ou distribuir o Programa será inválida, e automaticamente rescindirá seus direitos sob esta Licença. Entretanto, terceiros que tiverem recebido cópias ou direitos de você de acordo esta Licença não terão suas licenças rescindidas, enquanto estes terceiros mantiverem o seu pleno cumprimento.
6. Você não é obrigado a aceitar esta Licença, uma vez que você não a assinou. Porém, nada mais concede a você permissão para modificar ou distribuir o Programa ou respectivas obras derivativas. Tais atos são proibidos por lei se você não aceitar esta Licença. Conseqüentemente, ao modificar ou distribuir o Programa (ou qualquer obra baseada no Programa), você estará manifestando sua aceitação desta Licença para fazê-lo, bem como de todos os seus termos e condições para copiar, distribuir ou modificar o Programa ou obras nele baseadas.
7. Cada vez que você redistribuir o Programa (ou obra baseada no Programa), o receptor receberá, automaticamente, uma licença do licenciante original, para copiar, distribuir ou modificar o Programa, sujeito a estes termos e condições. Você não poderá impor quaisquer restrições adicionais ao exercício, pelos receptores, dos direitos concedidos por este instrumento. Você não tem responsabilidade de promover o cumprimento por parte de terceiros desta licença.
8. Se, como resultado de uma sentença judicial ou alegação de violação de patente, ou por qualquer outro motivo (não restrito às questões de patentes), forem impostas a você condições (tanto através de mandado judicial, contrato ou qualquer outra forma) que contradigam as condições desta Licença, você não estará desobrigado quanto às condições desta Licença. Se você não puder atuar como distribuidor de modo a satisfazer simultaneamente suas obrigações sob esta licença e quaisquer outras obrigações pertinentes, então, como conseqüência, você não poderá distribuir o Programa de nenhuma forma. Por exemplo, se uma licença sob uma patente não permite a redistribuição por parte de todos aqueles que tiverem recebido cópias, direta ou indiretamente de você, sem o pagamento de royalties, então, a única forma de cumprir tanto com esta exigência quanto com esta licença será deixar de distribuir, por completo, o Programa.

Se qualquer parte desta Cláusula for considerada inválida ou não executável, sob qualquer circunstância específica, o restante da cláusula deverá continuar a ser aplicado e a cláusula, como um todo, deverá ser aplicada em outras circunstâncias.

Esta cláusula não tem a finalidade de induzir você a infringir quaisquer patentes ou direitos de propriedade, nem de contestar a validade de quaisquer reivindicações

deste tipo; a única finalidade desta cláusula é proteger a integridade do sistema de distribuição do software livre, o qual é implementado mediante práticas de licenças públicas. Muitas pessoas têm feito generosas contribuições à ampla gama de software distribuído através desse sistema, confiando na aplicação consistente deste sistema; cabe ao autor/doador decidir se deseja distribuir software através de qualquer outro sistema e um licenciado não pode impor esta escolha.

Esta cláusula visa deixar absolutamente claro o que se acredita ser uma consequência do restante desta Licença.

9. Se a distribuição e/ou uso do Programa for restrito em determinados países, tanto por patentes ou por interfaces protegidas por direito autoral, o titular original dos direitos autorais que colocar o Programa sob esta Licença poderá acrescentar uma limitação geográfica de distribuição explícita excluindo esses países, de modo que a distribuição seja permitida somente nos países ou entre os países que não foram excluídos dessa forma. Nesse caso, esta Licença passa a incorporar a limitação como se esta tivesse sido escrita no corpo desta Licença.
10. A Free Software Foundation poderá de tempos em tempos publicar novas versões e/ou versões revisadas da Licença Pública Geral. Essas novas versões serão semelhantes em espírito à presente versão, mas podem diferenciar-se, porém, em detalhe, para tratar de novos problemas ou preocupações.

Cada versão recebe um número de versão distinto. Se o Programa especificar um número de versão desta Licença que se aplique a ela e a "qualquer versão posterior", você terá a opção de seguir os termos e condições tanto daquela versão como de qualquer versão posterior publicada pela Free Software Foundation. Se o Programa não especificar um número de versão desta Licença, você poderá escolher qualquer versão já publicada pela Free Software Foundation.

11. Se você desejar incorporar partes do Programa em outros programas livres cujas condições de distribuição sejam diferentes, escreva ao autor solicitando a respectiva permissão. Para software cujos direitos autorais sejam da Free Software Foundation, escreva para ela; algumas vezes, abrimos exceções para isso. Nossa decisão será guiada pelos dois objetivos de preservar a condição livre de todos os derivados de nosso software livre e de promover o compartilhamento e reutilização de software, de modo geral.

EXCLUSÃO DE GARANTIA

11. COMO O PROGRAMA É LICENCIADO SEM CUSTO, NÃO HÁ NENHUMA GARANTIA PARA O PROGRAMA, NO LIMITE PERMITIDO PELA LEI APLICÁVEL. EXCETO QUANDO DE OUTRA FORMA ESTABELECIDO POR ESCRITO, OS TITULARES DOS DIREITOS AUTORAIS E/OU OUTRAS PARTES, FORNECEM O PROGRAMA "NO ESTADO EM QUE SE ENCONTRA", SEM NENHUMA GARANTIA DE QUALQUER TIPO, TANTO EXPRESSA COMO IMPLÍCITA, INCLUINDO, DENTRE OUTRAS, AS

GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E ADEQUAÇÃO A UMA FINALIDADE ESPECÍFICA. O RISCO INTEGRAL QUANTO À QUALIDADE E DESEMPENHO DO PROGRAMA É ASSUMIDO POR VOCÊ. CASO O PROGRAMA CONTENHA DEFEITOS, VOCÊ ARCARÁ COM OS CUSTOS DE TODOS OS SERVIÇOS, REPAROS OU CORREÇÕES NECESSÁRIAS.

12. EM NENHUMA CIRCUNSTÂNCIA, A MENOS QUE EXIGIDO PELA LEI APLICÁVEL OU ACORDADO POR ESCRITO, QUALQUER TITULAR DE DIREITOS AUTORAIS OU QUALQUER OUTRA PARTE QUE POSSA MODIFICAR E/OU REDISTRIBUIR O PROGRAMA, CONFORME PERMITIDO ACIMA, SERÁ RESPONSÁVEL PARA COM VOCÊ POR DANOS, INCLUINDO ENTRE OUTROS, QUAISQUER DANOS GERAIS, ESPECIAIS, FORTUITOS OU EMERGENTES, ADVINDOS DO USO OU IMPOSSIBILIDADE DE USO DO PROGRAMA (INCLUINDO, ENTRE OUTROS, PERDAS DE DADOS OU DADOS SENDO GERADOS DE FORMA IMPRECISA, PERDAS SOFRIDAS POR VOCÊ OU TERCEIROS OU A IMPOSSIBILIDADE DO PROGRAMA DE OPERAR COM QUAISQUER OUTROS PROGRAMAS), MESMO QUE ESSE TITULAR, OU OUTRA PARTE, TENHA SIDO ALERTADA SOBRE A POSSIBILIDADE DE OCORRÊNCIA DESSES DANOS.

FINAL DOS TERMOS E CONDIÇÕES

Como Aplicar Estes Termos para Seus Novos Programas

Se você desenvolver um programa novo e quiser que ele seja da maior utilidade possível para o público, o melhor caminho para obter isto é fazer dele um software livre, o qual qualquer pessoa pode redistribuir e modificar sob os presentes termos.

Para fazer isto, anexe as notificações seguintes ao programa. É mais seguro anexá-las ao começo de cada arquivo-fonte, de modo a transmitir do modo mais eficiente a exclusão de garantia; e cada arquivo deve ter ao menos a linha de "direitos autorais reservados" e uma indicação de onde a notificação completa se encontra.

<uma linha para informar o nome do programa e uma breve idéia do que ele faz.>

Direitos Autorais Reservados (c) <nome do autor>

Este programa é software livre; você pode redistribuí-lo e/ou modificá-lo sob os termos da Licença Pública Geral GNU conforme publicada pela Free Software Foundation; tanto a versão 2 da Licença, como (a seu critério) qualquer versão posterior.

Este programa é distribuído na expectativa de que seja útil, porém, SEM NENHUMA GARANTIA; nem mesmo a garantia implícita de COMERCIALIZAÇÃO OU ADEQUAÇÃO A UMA FINALIDADE

Guia Livre – Referência de Migração para Software Livre do Governo Federal

ESPECÍFICA. Consulte a Licença Pública Geral do GNU para mais detalhes.

Você deve ter recebido uma cópia da Licença Pública Geral do GNU junto com este programa; se não, escreva para a Free Software Foundation, Inc., no endereço 59 Temple Street, Suite 330, Boston, MA 02111-1307 USA.

Inclua também informações sobre como contatar você por correio eletrônico e por meio postal.

Se o programa for interativo, faça com que produza uma pequena notificação como esta, quando for iniciado em um modo interativo:

Versão 69 do Gnomovision, Direitos Autorais Reservados (c) ano nome do autor. O Gnomovision NÃO POSSUI QUALQUER TIPO DE GARANTIA; para detalhes, digite 'show w'. Este é um software livre e você é bem-vindo para redistribuí-lo sob certas condições; digite 'show c' para detalhes.

Os comandos hipotéticos `show w` e `show c` devem mostrar as partes apropriadas da Licença Pública Geral. Naturalmente, os comandos que você utilizar poderão ter outras denominações que não `show w` e `show c`; eles poderão até ser cliques do mouse ou itens de um menu - o que for adequado ao seu programa.

Você também pode solicitar a seu empregador (se você for um programador) ou sua instituição acadêmica, se for o caso, para assinar uma "renúncia de direitos autorais" sobre o programa, se necessário. Segue um exemplo; altere os nomes:

A Yoyodyne Ltda., neste ato, renuncia a todos eventuais direitos autorais sobre o programa `Gnomovision` (que realiza passagens em compiladores), escrito por James Hacker.

<Assinatura de Ty Coon>

1º de abril de 1989, Ty Coon, Presidente

Esta Licença Pública Geral não permite a incorporação do seu programa a programas proprietários. Se seu programa é uma biblioteca de sub-rotinas, você poderá considerar ser mais útil permitir a ligação de aplicações proprietárias à sua biblioteca. Se isso é o que você deseja fazer, utilize a Licença Pública Geral de Biblioteca do GNU, ao invés desta Licença.