FDP

FDP **D**ocumentation **P**roject

MAC 499 - Trabalho de Formatura
Alex Pires de Camargo
Livio Baldini Soares
Rafael Caetano dos Santos

Coordenador: Arnaldo Mandel

A Rede Linux/Sala Pró-Aluno do IME é o único meio de acesso à Internet que os alunos de graduação do IME possuem. Além disso, é o único laboratório em ambiente Linux aberto aos alunos. Este dois atrativos, e uma série de serviços oferecidos pela rede, é que tornaram a rede tão importante para os alunos e, em especial, para os alunos do Bacharelado em Ciência da Computação (BCC).

A história da Rede Linux aqui no IME tem importância por que tem como conseqüência a atual estruturação de sua administração. A rede foi totalmente montada pelos alunos, pelo menos no que diz respeito à parte tecnológica. Atualmente a rede ainda é mantida por alunos do IME, todos alunos do BCC. Normalmente, um aluno do IME (em especial do BCC) leva de 4 a 6 anos para se formar, mas nem todos esses anos podem ser dedicados à administração da rede. A princípio, um aluno deve ter de 2 a 3 anos a serem dedicados à administração da rede. Este tempo é relativamente curto e, portanto, existe a necessidade de recrutamento de novos alunos. Estes novos alunos, em geral, não possuem o conhecimento e a experiência dos antigos administradores, e com certeza não sabem dos inúmeros detalhes que a rede Linux possui.

O projeto consiste em documentar a Rede Linux do IME, sua montagem, desenvolvimento e manutenção, de forma a passar de um estágio de tradição oral personalizada para uma organização permamente, com renovação periódica dos participantes. Neste ponto é importante a memória e experiência dos autores que que participaram da implantação desta rede desde seu início. Dessa forma esperamos que futuros administradores consigam manter a rede em funcionamento. Devemos documentar todos os aspectos que são relevantes à administração da rede Linux, desde a parte técnica de hardware, configurações de serviços e programas, até questões burocráticas.

Outra motivação para esse projeto de documentação é fato de que a rede cresceu sem muito planejamento, de modo que suas mudanças acabaram ficando sem a documentação apropriada, sendo conhecidas apenas pelos respectivos autores. Para que todos os administradores fiquem cientes de todas as partes envolvidas na rede e a facilidade da mudança que procedimentos tomados na rede, ou mudança de configuração numa segunda visitação, este texto facilitará muito a organização da administração.

OBS: Para que este projeto tenha o efeito duradouro que esperamos, este texto deve ser constantemente atualizado, conforme atualizações dos fatos na própria rede

Aqui vai um equema inicial de como faremos a estruturação da documentação.

1 Introdução

Neste documento, tentaremos cobrir todos os aspectos que estão presentes na nossa administração da Rede Linux do IME.

1.1 Ferramentas

Para documentação estamos utilizando o formatador de texto LaTeX em conjunto com o LaTeX2HTML e o editor de texto Emacs.

1.2 Tópicos

Os tópicos que serão cobertos nesta documentação estão relacionados abaixo. Observe que não apresentamos todos os tópicos detalhadamente: apenas os que julgamos mais importantes nas atividades administrativas da rede.

2 Histórico da Rede

A rede GNU/Linux do IME começou através de uma iniciativa pioneira dentro da universidade, protagonizada, na época, por dois alunos de graduação em Ciência da Computação: Félix F. B. de Almeida e Adriano N. Rodrigues.

Desde 1994, Adriano, então aluno do 2º ano, já possuía acesso às estações de trabalho da rede principal do IME/USP pelo fato de participar de projetos de iniciação científica dentro do Instituto (o acesso à rede principal do Instituto é limitado aos docentes e alunos de pós-graduação ou de iniciação científica). Com isso, ele e o seu amigo e colega de classe, Félix, tomaram contato com um sistema operacional UNIX e com as facilidades da Internet, que naquela época eram totalmente desconhecidos do público em geral.

Contagiado por essas facilidades e pelo novo paradigma desvendado pelo ambiente UNIX, Adriano procurava transmitir para Félix os conhecimentos que estava adquirindo como usuário da rede do IME/USP. Não demorou muito para que ambos ficassem sabendo da existência do recém criado Linux. Foi então que Félix, percebendo que com o Linux também poderia usufruir de um ambiente UNIX no seu microcomputador caseiro, decidiu importar um CD e um livro sobre o assunto. Com isso, tanto Félix quanto Adriano começaram a estudar esse sistema operacional incipiente, contando com a ajuda de alguns professores do instituto que já haviam tido alguma experiência com o Linux.

No início de 1995, depois de alguma insistência por parte dos dois alunos, o Departamento de Ciência da Computação cedeu-lhes uma sala pequena e quatro microcomputadores num dos cantos do Instituto para que pudessem se aprofundar no assunto.

Em face das dificuldades encontradas durante o processo de montagem de uma rede Linux, usando esses quatro microcomputadores e com a crescente divulgação que o Linux vinha recebendo devido ao aumento da popularidade da Internet é que Félix, em meados de 1995, teve a idéia de criar um núcleo de apoio e troca de informações para todos aqueles que estivessem na mesma situação que eles, ou seja, aprendendo a instalar e configurar microcomputadores com Linux. Essa idéia foi muito bem recebida pelo departamento, que, para facilitar a divulgação dessa atividade, resolveu permitir o acesso de Félix à rede principal do IME/USP. Esse núcleo de apoio aos usuários de Linux foi batizado de Grupo de Usuários Linux, ou simplesmente GUL.

Depois de um período de experiência, e já contando com o acesso à Internet (gentilmente cedido pelo Instituto), Félix e Adriano resolveram liberá-la para ser usada pelos alunos do Bacharelado em Ciência da Computação em março de 1997, julgando ser de relevante importância para a formação desses alunos o acesso a um email próprio e à world wide web bem como o contato com um sistema operacional UNIX.

Não demorou muito para que esse projeto ganhasse destaque dentro do Instituto e, com isso, mais salas e microcomputadores foram sendo incorporados à rede Linux, bem como mais alunos foram tomando conhecimento e participando dele. As disciplinas do curso de computação também começaram a usar os recursos oferecidos pela rede Linux durante os períodos letivos, o que aumentou ainda mais a adesão ao projeto. Desta maneira, acabaram surgindo dois projetos separados porém relacionados: a Rede GNU/Linux e o Grupo de Usuários Linux.

O GUL também mostrou seu valor quando, no final de 1997, recebeu o prêmio Os Vencedores da revista Info em nome de toda a comunidade Linux do Brasil. Esse prêmio foi entregue numa cerimônia de gala realizada no hotel Renaissance em São Paulo, onde estavam presentes representantes de todas as grandes empresas de informática do país bem como expoentes do mundo jornalístico.

Dado o sucesso do projeto da rede Linux, a universidade decidiu dar apoio a ele através do Programa Pró-Aluno. Com isso, em março de 1999, foi inaugurada a sala Pró-Aluno do IME/USP abrindo a possibilidade de que todos os alunos de graduação do Instituto pudessem utilizá-la, não ficando mais restrita somente aos alunos de Computação. Desse modo, a rede Linux se estabeleceu como plataforma para a graduação dentro do Instituto e o GUL como veículo para intercâmbio de informações entre usuários Linux de todo o país.

Outras unidades dentro da USP e também outras universidades do país estão seguindo o mesmo caminho e estudando a viabilidade de se implantar um projeto como o nosso em seus campi.

Atualmente, a rede Linux é administrada por excelentes alunos da computação que também fazem parte do GUL. Conforme os administradores da rede Linux vão chegando perto da conclusão do curso eles selecionam e treinam uma nova geração de administradores que os substituirão assim que se tornarem bacharéis e deixarem a graduação.

3 Procedimentos Administrativos

Os procedimentos administrativos são variados e geralmente cobrem situações menos técnicas e mais burocráticas. Os procedimentos estão divididos em duas partes.

3.1 Interface com Usuário

Procedimentos realizados para a "satisfação" dos usuários e também para integração do usuário à rede.

3.1.1 Abertura de conta

Requisitos para alguém poder ter conta na rede Para ter conta na rede Linux, é preciso atender algum dos seguintes critérios:

- Ser aluno de graduação do **IME**, regularmente matriculado.
- Ser professor do **IME**.

Processo de abertura de conta De posse da carteira USP do novo usuário, ou cartão de matrícula e RG, pode-se abrir sua conta no momento do pedido através do script `abrir`, no servidor C (ver seção 5.7.4). No momento da abertura da conta, deve-se comunicar ao usuário informações essenciais para sua sobrevivência na rede através de um documento impresso junto de sua senha e login, além de eventuais esclarecimentos verbais.

3.1.2 Suspensão temporário de uma conta

Um usuário pode ter sua conta suspensa, por tempo a ser determinado pelos administradores, se cair em um dos seguintes casos:

Senha quebrada pelo Cracker da rede Caso o usuário tenha sua senha quebrada pelo cracker da rede, sua conta deve ser suspensa automaticamente. A senha permanece suspensa até que o usuário compareça à sala da administração para reabri-la com uma nova senha, o que é feito através do script reabrir (ver seção 5.7.4). Nessa ocasião, o admin deve alertar os perigos decorrentes do uso de senhas fracas, e como criar senhas boas.

Mau comportamento Fica a cargo da administração decidir o que é ou não mau comportamento, pois sempre vão surgir situações novas. Seguem alguns itens que podem ser encarados como mau comportamento.

- Mexer em arquivos de outros usuários sem autorização dos mesmos.
- Tentativa de ataques à rede ou a outros sites.
- Práticas que possam causar danos ao hardware da rede.
- Abuso de recursos, como o espaço da partição `/tmp`.
- Tentar descobrir a senha de outro usuário.
- Emprestar a própria conta, informando a própria senha a outros.

Nesses casos, o tempo de suspensão fica a cargo dos administradores, que devem entrar num consenso quanto ao tipo e ao tempo de punição. Deve-se levar em conta casos parecidos que tenham ocorrido na rede. Dependendo da situação, é conveniente que o usuário seja contatado antes que se decida sobre a suspensão. Em todo o caso, o usuário deve ser informado sobre o motivo e a duração da suspensão.

3.1.3 Reabertura de conta

Quando o tempo de suspensão se esgota, a reabertura deve ser realizada apenas com a presença do usuário, que deve pegar sua nova senha (ver seção 5.7.4).

3.1.4 Cancelamento permante de conta

O usuário tem sua conta permanentemente cancelada quando:

- Assim é da sua vontade.
- Ele deixa de ser aluno regular de graduação do **IME**.

Nesses casos o usuário pode indicar um endereço para redirecionamento dos seus mails, que funciona durante 6 meses, ou até que alguém abra uma conta com o seu login.

3.1.5 Plantão e atendimentos

Todos os administradores devem permanecer de plantão na sala da administração por no mínimo quatro horas semanais, sem coincidência de horários. Pelo menos um dia por semana deve haver alguém de plantão à noite, para atender o pessoal do período noturno.

3.1.6 Mudança de grupo

O usuário é automaticamente mudado de grupo quando muda de curso.

Para essa mudança existe um script que **deve** ser usado a fim de não criar arquivos de senhas incorretos. Veja a seção 5.7.4 para mais detalhes.

3.1.7 Pedidos de usuários. Para aumento de quota, instalação de software, etc

Os usuários devem fazer seus pedidos diretamente à administração, seja pessoalmente ou (preferencialmente) por mail, e fica a cargo desta deferir ou não o pedido.

3.1.8 Suporte ao usuário

O suporte ao usuário da rede é dado pelos FAQs da rede (veja seção 8.1.2, através do endereço de email `admin@linux.ime.usp.br`, e pessoalmente, tanto no horário de plantão (veja 3.1.5 acima), como em outros horários em que os admins estejam disponíveis da sala da administração.

3.2 Procedimentos Internos

Procedimentos que são tomados dentro da rede, que envolvem apenas os administradores. Aqui estão listados mais os aspectos burocráticos dos procedimentos, ou seja, como devemos tomar decisões sobre a parte "política", e não "técnica", da administração.

3.2.1 Pedidos mensais de materiais

A cada mês, fazemos um pedido de materiais junto ao **D**epartamento de Ciência da Computação (DCC). Apenas materiais considerados "de consumo" podem ser requisitados. Isso inclui materiais que são evidentemente "de consumo", como formulário contínuo, fita para impressora matricial, papel para impressora laser, mas não inclui, por exemplo, toner para impressora laser. Por outro lado, alguns materiais "duráveis", como mouses, podem ser pedidos.

O pedido deve ser entregue até a primeira quinta-feira do mês, na secretaria do **D**CC.

3.2.2 Responsáveis por cada uma das salas da rede

A cada semestre, quando é feita a distribuição de tarefas (veja item a seguir), cada sala é atribuída a um ou dois admins, que passam então a ser responsáveis pelo bom funcionamento da sala. Além das máquinas da sala, deve-se cuidar do aparelho de ar-condicionado, das instalações físicas, cadeiras, etc, fazendo pedidos de conserto à Seção de Materiais quando for o caso. Se um admin perceber um problema numa sala pela qual não seja responsável, deve informar os responsáveis pessoalmente ou, preferencialmente, através de email para *admin*.

3.2.3 Responsáveis por cada um dos serviços da rede

No início de cada semestre, é feita uma reunião entre os admins, na qual se decide quais admins serão responsáveis por quais serviços da rede.

Os serviços basicamente são:

- Servidor de homes
- DNS
- Servidor de mail
- NIS
- Firewall
- Backups
- Instalação de software empacotado
- Instalação de software avulso
- Instalação de novas máquinas
- Manutenção da máquina de testes (ver 7.1.1)
- Manutenção do servidor de CPU (ver 7.1.1)

3.2.4 Políticas

Instalação de Software. Decisão sobre a sua instalação ou não instalação A maior parte da instalação de software é feita durante a instalação das máquinas (ver seção 5.5). A escolha dos pacotes baseia-se na lista de pacotes instalados anteriormente.

Novos programas podem vir a ser instalados por várias razões: por não fazerem parte da distribuição de escolha, por pedidos de usuários ou de professores, lançamentos de programas, obsolescência de programas instalados, etc.

Deve-se evitar instalar programas cuja funcionalidade já esteja presente em outros programas instalados na rede.

Deve-se dar preferência a software livre. #Isto é, caso haja um programa livre que preencha a necessidade em questão, deve ter preferência.

Programas proprietários só podem ser instalados se o **I**ME tiver a licença apropriada.

Casos especiais que fogem às regras. Tanto regras da Pró-aluno, quanto regras internas da rede Casos especiais que fujam às regras da rede devem ser discutidos, inicialmente, por email entre os admins. Exemplos: um programas que já tenha um equivalente instalado, e que seja requisitado por um professor por ser considerado mais adequado ao uso numa disciplina.

Caso não haja consenso, de acordo com a seriedade da questão, deve-se fazer votação por email, ou uma reunião para se discutir e votar.

Nessas decisões, sempre deve-se levar em conta a "jurisprudência" criada.

Aumento da quota de disco A princípio, pedidos de aumento de quota de disco não são aceitos, devido à dificuldade técnica de manter controle dessas exceções, e à questão política de se definir quem pode ter aumento de quota.

Caso seja apropriado, o usuário deve ser orientado a requisitar uma conta na **R**ede IME.

Abertura de conta De acordo com o http://amandy.redealuno.usp.br/caderno/regulamento.htm, para que um aluno possa abrir conta na rede, deve ser aluno regular de graduação do **I**ME.

Abrimos exceções a professores do **I**ME, pois é conveniente para os alunos que os professores tenham acesso ao mesmo ambiente onde desenvolvem seus *EP*s.

3.2.5 Reclamação com o IME

Os administradores, além de cuidarem da parte "lógica" da rede, precisa estar sempre atentos à parte "física" da rede: as salas que compõe a rede, sua higiente, os micros, aparelhos de ar-condicionado, bancadas, cadeiras, lâmpadas, etc.

Caso algum computador apresente mal-funcionamento por problemas de hardware, veja a seção abaixo para verificar o procedimento.

Caso surja algum problema "físico" numa sala, eis alguns dos procedimentos a serem tomados, de acordo com o problema:

- Problemas com ar-condicionado: Contatar o pessoal do patrimônio.
- Problemas com cadeiras, lâmpadas, chão, teto, mesa e similares: Contatar o pessoal dos serviços gerais.
- Problemas com limpeza: Falar diretamente com o responsável pelo serviço de limpeza do IME.

3.2.6 Conserto de micros

Quando se detecta alguma falha de hardware em algum micro, deve-se primeiro analisar a situação desse equipamento, levando-se em consideração o histórico da peça: de onde originou e quais problemas de hardware já teve.

Por ser uma rede heterogênea em relação à origem de seus equipamentos, é necessário verificar previamente que tipo de garantia esta peça possui, e caso esteja na garantia "de fábrica", deve-se entrar em contato com o fornecedor do equipamento para se fazer conserto ou troca. Se não estiver mais na garantia, provavelmente o equipamento pertence ao programa Pró-aluno e portanto goza da manutenção do CCE. Portanto, basta fazer um chamado para a seção de consertos do CCE requisitando o conserto da peça.

O chamado é feito por telefone, e é preciso informar o número de registro USP ou IME da peça, uma descrição sucinta do problema, e a sala onde o equipamento deve ser retirado pelo CCE. É muito importante anotar o número do chamado, junto de uma descrição do equipamento, de seu problema, e da data do chamado. Essas informações são mantidas em papel, numa pasta guardada na sala da administração. Também é recomendável mandar email para a administração informando sobre o chamado. A responsabilidade do chamado é dos admins encarregados da sala onde houve o problema.

Alguns casos especiais podem surgir quando o equipamento não tem garantia nenhuma. Nesse caso, uma solução é efetuar uma compra quando houver verba, para que os próprios administradores façam a substituição desta peça danificada (veja 3.2.8).

Eis um sumário das situações e respectivas providências a serem tomadas:

Consertos de micros:

- Micros com garantia:
  - Ver se o micro pode ser aberto (isso pode ser exigido no momento da compra), para fazermos um diagnóstico melhor.
  - Falar com o pessoal do patrimônio para acionar a empresa que vendeu o micro, passando a eles a peça ou o micro inteiro, dependendo do tipo de garantia.
- Micros sem garantia:
  - Abrir o micro para detectar problemas. Se não for possível resolver, chamar o CCE. Nesse caso, o serviço é cobrado do IME pelo CCE.

Canibalização Quando se tem mais de um micro com problemas diferentes e chega-se a conclusão que não vale a pena (ou é inviável) mandar arrumá-los, é praxe trocar as peças para que apenas um fique com as ruins, de modo a maximizar o número de micros funcionando. Os micros que ficam com as piores peças são os micros canibalizados. Recomendamos que se verifique sempre se existem peças boas em micros canibalizados, e quando alguma peça ruim for passada pra outro micro, deve ser marcada de algum jeito razoável.

3.2.7 Funcionamento das salas

Nesta seção são apresentados os aspectos "burocráticos" que estão presentes no uso das salas que compõe a rede. Atualmente, a rede é composta de 4 salas, todas no bloco A, e uma bancada no Centro de Ensido de Computação (CEC), bloco B. As salas são: 125-A, 126, 258-B, 257 no bloco A, e sala 5 do CEC.

Horário de funcionamento Por decisão do CTA do IME, as salas da rede foram liberadas para uso ininterrupto, 24h por dia, todos os dias da semana, desde que o IME esteja aberto. Em particular, a sala 126, dentre todas as salas do Programa Pró-Aluno da USP, é a única que fica aberta 24h por dia.

Quais grupos podem usar quais salas A rede foi "dividida" em duas, pelo menos em seu aspecto formal. Uma é a rede Pró-Aluno do IME, na qual qualquer aluno de graduação do IME tem direito de acesso. A outra é a rede Linux do BCC, montada por equipamentos adquiridos através de verba do Departamento de Ciência da Comptuação (DCC), de seus docentes e projetos. Portanto, as salas que comportam esses equipamentos são de acesso exclusivo dos alunos do BCC.

Essa distinção entre os usuários na rede foi implementeada através da criação de grupos, que correspondem ao curso do aluno. Para restringir o login em certas máquinas a determinados grupos, foi criado o script `/etc/loga.sh`.

Recursos disponíveis nas salas Devido a essa duplicidade da rede, é preciso cuidado ao escolher a sala de destino de novos equipamentos e recursos. O mesmo vale para o remanejamento de equipamentos existentes. Esta decisão pode estar limitando ou liberando recursos para diferentes grupos. A política atual determina que os recursos são disponibilizados conforme a origem do recurso, ou seja, se o recurso vem do IME, deve ser disponibilizado para todos os usuários, e se for doado pelo DCC, apenas os alunos do BCC devem usufruir desse recursos. Exceções podem ser feitas, desde que discutidas previamente entre os admins, por email ou em reunião.

As salas em si Aqui estão listadas todas as salas da rede, com uma descrição informando que grupos podem usá-las e quais recursos contêm.

- 125-B. Sala para uso exclusivo da administração. Esta sala deve fica trancada na ausência de administradores. Todos os servidores devem ficar nesta sala, para que fiquem protegidos fisicamente. Além de servidores, nesta sala há dois armários, utilizados para guardar equipamentos avulsos da rede. Há também 2 impressoras: uma matricial, utilizada principalmente para imprimir dados de abertura de conta; e uma laser, para uso dos administradores.
- 126. Esta sala contém os micros da rede Pró-Aluno e pode ser utilizada por qualquer usuário da rede, de qualquer grupo. Há também 2 impressoras: uma laser e uma matricial.

- 258-B. Esta sala é de uso exclusivo dos alunos do BCC. Além dos micros, há uma impressora matricial, também de uso exclusivo dos alunos de BCC. Para utilizá-la é necessário logar remotamente no micro que esta ligado a ele.
- 257. Esta sala é de uso exclusivo dos alunos do BCC. Além dos micros, há nela um scanner disponível aos alunos do BCC.

3.2.8 Compras

Esta seção trata do que fazer com verbas: como e o quê comprar.

O que comprar? Primeiro é necessário decidir o que comprar com a verba. Essa decisão depende muito da situação corrente da rede, mas deve ser sempre feita uma discussão entre todos os administradores da rede, por email ou em reunião.

Deve-se levar em conta as necessidades da rede no dado instante. Dois aspectos são muito importantes aqui e devem ser pesados. Primeiro, é o que diz respeito ao conserto ou renovação de equipamentos. Como a verba é geralmente pequena, é impossível que todos os equipamentos sejam "top de linha", portanto deve-se avaliado o quanto da verba deve ser destinado a consertos e reposição de equipamentos quebrados ou ultrapassados. Outra aspecto importante é a aquisição de equipamentos novos. Aqui aqui deve-se considerar os recursos que os usuários ou administradores sintam estar em falta, ou que possam faltar num futuro próximo.

Cotação e especificação de produtos para a rede Após a decisão do que comprar com a verba, uma cotação deste produto no mercado é necessária para verificar se a verba disponível é realmente suficiente para a compra.

Ao menos 3 fornecedores devem ser consultados previamente à compra. Depois dessa cotação, deve ser feita uma especificação **muito detalhada** do equipamento. Recomenda-se fortemente a inclusão do maior número possível de detalhes técnicos sobre o equipamento, de modo a conseguir um produto o mais semelhante possível ao desejado. Observe que não é permitido a menção de marcas de produtos.

Acompanhamento do processo de compra O último estágio da compra de um equipamento se dá após a especificação do produto. É aberto um concurso público, e os fornecedores inscritos devem apresentar um documento contendo a descrição dos equipamentos que satisfaçam as especificações, junto dos respectivos preços. Atualmente, o critério dos concursos é o preço total mínimo.

Escolhido o fornecedor, é obrigação que um administrador acompanhe o processo, verificando que o fornecedor entregue os equipamentos dentro do prazo, e que estes estejam conforme a especificação desejada e o documento apresentado pelo fornecedor.

4 Hardware

Esta seção descreve por cima o hardware da rede Linux e alguns dos procedimentos tomados com relação a eles.

4.1 Micros

O principal artigo de hardware que temos na rede são os micros. Apesar de não ser a principal função de um administrador, o acompanhamento desses equipamentos é necessário durante sua compra, instalação e manutanção nas salas.

É com freqüência que os partes dos micros dão problemas. Entre eles, algumas partes que são vítimas de panes constantes são ratos, ventilador do processador, discos rígidos e monitores. E por isso é necessário uma constante vigilância desses equipamentos. Em caso de detecção de pane deve-se seguir as instruções da seção 3.2.6.

4.1.1 Padrões Mínimos

A principal característica que deve ser observada ao incorporar um equipamento à rede é a compatilibilidade com o Linux, em especial tem que ter suporte no kernel do Linux ou que esteja disponível algum driver/módulo que habilite o uso de alguma peça em questão.

O suporte do Linux para diferenciados equipamentos aumentam constantemente e de forma rápida. Geralmente o documento **Hardware-HOWTO** contém uma lista de equipamentos que são e que não são suportados pelo Linux. Como a mudança no kernel do Linux muitas vezes é bastante veloz, o documento disponível na distribuição utilizada pode estar desatualizada e portanto é interessante acessar a versão mais atualizada em *http://users.bart.nl/ patrickr/hardware-howto/Hardware-HOWTO.html.*

Uma outra necessidade que todos os micros da rede devem ter que às vezes é um diferencial com relação aos micros largamente comercializados é a placa de rede. Todos os micros devem conter uma placa de rede compatível com a largura de banda utilizada na rede.

Outro aspecto muito importante sobre padrões de micros aceitos na rede é a quantidade de memória e espaço em disco. A quantidade de memória deve ser suficiente para suportar a maioria dos programas freqüentemente utilizados pelos usuários, como por exemplo, *Netscape*, *Star Office*. A quantidade de espaço de disco deve ser suficiente para suportar a instalação de todos os programas de nossa seleção (ver a seção 5.5). Mas vale lembrar que também são necessário algum espaço extra para deixar espaço para os usuários escreverem no `/tmp/` e os logs serem escritos no `/var/` com folga para não travar o sistema.

Esses programas, quantidade de memória e quantidade de espaço em disco devem variar um pouco (ou bastante) anualmente e por isso deve ser feito um recálculo dessas quantidades. Se for notada alguma defasagem das máquinas disponíveis com o padrão desejável, é necessário então verficar a possibilidade de se fazer upgrade nas partes do equipamento já defasados.

4.2 Periféricos

Atualmente os periféricos disponíveis na rede Linux são um scanner, impressoras e zip-drives internos.

O uso do único scanner que a rede Linux possui é restrito, veja na seção 3.2.7.

A sua utilização do scanner, dos zip-drives (internos e externos, trazidos pelos usuários) e das impressoras estão documentados para os usuários através dos FAQs (veja 8.1.2).

4.3 Cabeamento

A parte de cabeamento da rede não está sob responsabilidade dos administradores. Mas mesmo assim é importante o conhecimento do equipamento que faz o roteamento dos pacotes da rede. A rede Linux faz sua comunicação através dos hubbies instalados nas diversas salas. Por este fato é que a rede Linux não é fisicamente isolada da rede IME (leia mais sobre isso na seção 7.1).

Pode ocorrer que algum ponto da rede não esteja funcionando e aí o administrador deve checar a integridade do ponto na parede e sua conectividade com o hubbie da sala conforme número indicado no próprio ponto.

O órgão responsável pelos cabeamentos e hubbies é a Seção de Informática do IME e deve ser contactado logo que seja detectado algum problema nesse equipamento.

Também é importante notar que o IME possui um switch que faz o principal roteamento entre os pacotes externos e internos da rede do instituto. O nosso servidor principal (com conexão a internet 7.1.1) atualmente está ligado a um dos hubbies de uma das salas, mas achamos que ganharíamos em conectividade se o ponto externo desse servidor fosse ligado diretamente ao switch do instituto.

5 Software

Esta seção é bastante extensa e variada. Ela abrange todo tipo de software que existe na rede, e quando necessário, contém uma descrição detalhada. Aqui tentaremos abordar o porquê da instalação da Debian GNU/Linux, métodos de instalação de software, e por fim, software fabricado pela própria administração, principalmente para facilitar a administração, mas também para tornar possível o controle dos recursos da rede.

5.1 Por que Linux?

A principal motivação para o uso do Linux na rede dos alunos é a falta de acesso a um sistema operacional "de verdade" aos alunos do Bacharelado de Ciência da Computação. Anteriormente à existência da rede o único sistema operacional acessível aos alunos era o Windows. Os alunos sentiram a necessidade de contato com um sistema operacional Unix ou similar que alguns haviam pelo programa de iniciação científica ao acessarem a rede das ET's do IME.

O Linux parecia ser o sistema operacional ideal para isso. Assim os alunos teriam contato a um ambiente de programação/desenvolvimento completo, muito importante para sua formação. Além disso a aprendizado é bastante facilitado pelos inúmeros aplicativos voltado à computação que são distribuídos abertamente para o Linux.

Além disso aqui vão as principais vantagens do Linux, logicamente existem algumas desvantagens que devem ser citadas também.

5.1.1 Vantagens

As principais vantagens que o uso do Linux traz são:

- É um sistema operacional Unix. Assim, é um sistema mais completo e robusto que o Windows.

- É um sistema operacional livre (e grátis). Isso implica em não sobrecarregar financeiramente o Instituto e a Universidade. Também dá ao aluno a possibilidade de olhar e modificar o código fonte dos inúmeros programas (inclusive o kernel em si) distribuídos para o Linux. Isso é de um valor imenso para os alunos, que assim podem ver como vários projetos *de qualidade* são feitos.

- É fácil construir uma rede voltada para a Internet. Já que o Linux "nasceu" e cresceu na Internet, existe uma infinidade de aplicativos que garantem o fácil acesso dos usuários desse sistema operacional aos diferentes tipos de serviços da Internet.

- É fácil fazer uma rede segura. Como é um sistema em constante depuração por milhares de pessoas, tanto o kernel como os utilitários e aplicativos do Linux são bastante confiáveis. Além disso, os furos de segurança são corrigidos muito mais rapidamente que em sistemas proprietários.

- Como há um grande número de redes rodando em Linux, e uma grande comunidade disposta a discutir o Linux em listas de discussão, newsgroups e websites, é fácil encontrar soluções para os problemas administrativos que encontramos.

5.1.2 Desvantagens

Talvez a maior desvantagem de basear a rede no Linux é que a grande maioria dos alunos nunca teve contato com sistemas Unix. Assim, grande parte dos alunos demonstra dificuldade em utilizar a rede, principalmente os que não cursam BCC.

Para amenizar esses problema, tentamos cobrir os pontos mais difíceis através de FAQs (veja seção 8.1.2). Entretanto, aparentemente estes ainda não suficientes (ou adequados) para eliminar todas as dificuldades dos usuários. Além disso, muitos usuários não gostam ou não se preocupam em ler documentação.

Um "problema" relacionado é a ausência de certos aplicativos proprietários comumente usados doméstica e comercialmente, como Word, Excel, Power Point, etc. Esse problema tem sido sanado pelo surgimento de programas compatíveis, como o StarOffice.

5.2 Distribuição usada: Debian

O Linux em si consiste apenas de um kernel que, por si só, é inútil para o usuário final. O que torna o Linux útil são os inúmeros aplicativos que são distribuídos com ele. É aí que entra o papel da distribuição. Uma distribuição consiste em um kernel, um conjunto de utilitários, aplicativos e bibliotecas, e um sistema de instalação e upgrade. Atualmente há várias distribuições do Linux.

Nós utilizamos a distribuição Debian GNU/Linux. Seu kernel é (obviamente) o Linux. Seus utilitários básicos de shell, desenvolvimento, rede, manipulação de arquivos e texto são na maioria provenientes do projeto GNU e do BSD. Há centenas de utilitários e aplicativos de outras fontes.

Todos os programas da distribuição principal (*main*) da Debian é software livre, normalmente licenciado sob a GPL (do projeto GNU) ou alguma licença tipo BSD. Além disso, existe a distribuição *non-free*, destinadas a software proprietário, e a *contrib*, destinada a programas que dependam de software proprietário.

Por ser um projeto muito aberto e dinâmico, mas ao mesmo tempo muito preocupado com segurança e estabilidade, a Debian mantém várias distribuições concorrentemente. A distribuição "de produção", i.e., mais estável e livre de bugs, é a **stable**. Uma vez lançada, a **stable** apenas é alterada em caso de consertos de segurança. Há também a distribuição **testing**, e a **unstable**, que contém as mais recentes atualizações de aplicativos, do kernel e das bibliotecas. Por razões de estabilidade e segurança, utilizamos sempre a versão **stable** da Debian.

5.2.1 Motivos

Um dos principais motivos para o uso da Debian (e em particular, a versão stable) é sua robustez. Como a Debian segue modelo de desenvolvimento bastante aberto, **milhares** de pessoas participam fazendo pacotes, enviando relatórios e correções de bugs, e discutindo mudanças. Assim, a Debian é testada em muitos ambientes e cenários distintos, proporcionando uma estabilidade maior que qualquer outra distribuição. O processo de lançamento de uma versão estável da Debian é bastante rigoroso, e apesar de ser mais demorado que em outras distribuições, julgamos que este é um preço que vale pagar.

A Debian é uma distribuição bastante organizada, onde existem uma política também rigorosa para a criação e manutenção de um pacote. Seu sistema de empacotamento de software permite uma *atualização* muito fácil de uma versão para a outra, o que nem sempre acontece com as outras distribuições Linux.

Outra vantagem do uso da Debian é que traz mais pacotes de caráter acadêmico e de desenvolvimento de software do que outras distribuições.

Por fim, outra vantagem da Debian é ter maior segurança "de fábrica" do que outras distribuições, ou seja, seus progrmas e suas configurações padrões são bem mais confiáveis que os de outras distribuições. Além disso, quando são encontradas falhas de segurança na versão **stable**, os respectivos consertos são lançados num diretório chamado **debian-security**, normalmente num curto espaço de tempo após a descoberta das falhas.

5.3 De onde pegamos a distribuição

Para a instalação e atualização dos clientes e servidores, seria inconveniente ter que sempre baixar os pacotes diretamente do servidor da Debian, devido à baixa taxa de transferência de dados entre este e a nossa rede. velocidade de nosso link. Portanto, criamos um espelho da distribuição Debian no **limbo.ime.usp.br** (ver seção 7.1.1). Apesar desse micro não fazer parte, oficialmente, da rede Linux, atualmente são os administradores desta que tomam conta do **limbo** e de seu espelho. Este espelho é uma cópia dos arquivos em **ftp.debian.org**, principal local onde os pacotes da Debian são guardados. Diariamente, o script que faz o espelho verifica modificações na distribuição e já espelha em nossa versão local.

Este espelho é disponível via *NFS* para qualquer micro da rede IME. Em particular, ele está disponível na rede linux através do diretório `/usr/local/espelho/`, em qualquer cliente da rede (ver 7.1.1). Isto faz com que a instalação se acelere em muito em comparação a pegar os pacotes diretamente da Debian (ou outros espelhos distantes).

5.4 Sincronia entre clientes

A rede é composta de muitos micros com diferentes configurações de hardware, situdados em salas diferentes, às vezes destinados a grupos diferentes. Entretanto, um princípio da administração é que a rede deve ser o mais homogênea possível do ponto de vista do usuário. Assim, deve-se tentar garantir que todos os micros da rede tenham os programas disponíveis, com a mesma configuração. Em particular, é desejável que a forma de logar seja a mesma em todas as máquinas.

Entretanto, devido a limitações e incompatibilidades de hardware, nem sempre é possível atingir total homogeneidade. Por exemplo, algumas máquinas possuem teclados padrão ABNT, enquanto outras seguem o padrão americano.

Em outros casos, a homogeneidade é possível, mas indesejável, pois implica em "nivelar por baixo". Por exemplo, tivemos algumas máquinas defasadas tecnologicamente em que não era possível usar a resolução gráfica máxima das outras (1024x768 pixels). Nesse caso, escolhemos configurar as máquinas mais novas com resolução de 1024x768, mesmo que nas obsoletas só pudéssemos usar 800x600. Outras máquinas não tinham espaço em disco suficiente para a instalação de todos os programas que gostaríamos de instalar. Assim, nessa máquinas deixamos de instalar alguns programas que não julgamos essenciais. Para informações sobre a configuração mínima de hardware na rede, veja 4.1.1.

Mesmo relevando os problemas de hardware, por enquanto ainda não encontramos uma maneira automatizada que garanta esta sincronia entre as instalações dos clientes. Uma ferramenta que é bastante útil neste sentido é o **espalha** (5.7.2), que faz manutenções de qualquer tipo em todos os micros da rede, ao mesmo tempo.

Outra maneira é o próprio uso do sistema de pacotes Debian nos clientes. Isto faz com que tenhamos um grande controle de quais programas estão instalados em um micro. E quanto menos se quebra a política definida em 5.6.1, mais sincronizados os clientes tendem a ficar.

Alem dessas duas ferramenteas, não temos mais nenhum mecanismo para fazer esta sincronização. Apesar disso, verificamos que os clientes parecem ter configurações idênticas a maior parte do tempo, descontadas as diferença de hardware citadas acima.

5.5 Instalação de software

Quando é feita a primeira instalação em um cliente, ou num grupo de clientes, é necessário definir todos os programas que serão instalados. Inicialmente todos os programas a serem instalados são pacotes que já existem na distribuição atual da Debian. Pacotes adicionais (ver seção 5.6.4) podem ser criados e incluídos na seleção de pacotes.

Geralmente, a seleção do conjunto inicial de pacotes é feita no micro de testes da rede (7.1.1) de tal forma que todas as dependências sejam satisfeitas. Vários dos admins, senão todos, devem aprovar a seleção feita para tentar cobrir todos os tipos de programas úteis para nossos usários. Por outro lado, deve-se ter o cuidado de não instalar programas de uso muito limitado, ou programas que dupliquem desnecessariamente a funcionalidade outros, a fim de economizar espaço nos discos dos clientes.

Eis um guia passo-a-passo para instalação seleção inicial de programas:

- **Acerto da fonte de pacotes (espelho).** Antes do começar o processo de seleção de pacotes, é necessário decidir a fonte de onde serão pegos. O método que usamos para recolher os pacotes é o **apt-get**, um front-end para o **dpkg** que possue recursos "inteligentes" como ordenação automática da instalação de pacotes. A rede IME possui um espelho da Debian/binary-i386 (7.1.1), portanto deve-se arrumar o arquivo `/etc/apt/sources.list` para que o **apt** pegue os pacotes do espelho desejado.

  Deve-se incluir também as fontes de pacotes criados pelos administradores para uso interno da rede.

- **Escolha dos pacotes.** O uso do **dselect** é recomendado para fazer a escolha de pacotes já que nele o problema de inter dependências dos pacotes é mais facilmente resolvido e sua interface permite uma identificação do pacote mais rápida. No **dselect** deve-se selecionar o método de acesso como sendo o **apt**.

  Daí começa o processo de escolha em si. Este processo é muito longo, pois a quantidade de pacotes disponíveis já é muito alta e tende a crescer bastante de uma versão para outra da Debian. Por isso, é indispensável ter a seleção de pacotes da versão anterior para que nenhum pacote importante ou amplamante utilizado seja esquecido.

  Uma ajuda na decisão de escolha de pacotes está vindo da própria Debian. Esta é a criação dos chamados *meta-packages*, pacotes que não contêm programa algum, mas sim dependências a serem resolvidas. Vide o exemplo dos pacotes *task-**.

- **Após a escolha.** Ao final da escolha dos pacotes e sua instalação no micro, saia do dselect e execute o comando:

```
$ dpkg --get-selections  > /tmp/selections
```

Esse comando gravará em */tmp/selections* a listagem de todos os pacotes selecionados e instalados.

Isto será útil para que possamos instalar os mesmos pacotes em todos os clientes da rede. Para isso, ao fazer a instalação de um cliente comum, deve-se primeiro acertar o arquivo `/etc/apt/sources.list`, "desselecionar" todos os pacotes do dselect e então executar o comando:

```
$ dpkg --set-selections  < /tmp/selections
```

Esse comando selectionará todos os pacotes listados no arquivo */tmp/selections*.

Não se pode esquecer de guardar o arquivo de seleções, pois este será essencial na instalação dos outros clientes (como descrito acima). Após isto, todos os programas "padrões" já devem estar instalados neste cliente.

Depois desse processo, pode ser necessária a instalação de programas adicionais, descrita a seguir.

5.6 Instalação de software que não está na instalação original da rede

Após a instalação original da rede, é comum haver a necessidade de instalação de software adicional. Os motivos para isto podem ser vários, desde esquecimento ao fazer a seleção inicial de pacotes, pedido de usuário, ou decisão posterior de fazer upgrade em apenas alguns pacotes.

Esta seção cobre os variados casos e estratégias que são necessárias para este tipo de instalação de software. É importante observar que deve-se sempre manter a instalação da rede uniforme. Veja 5.4 para mais informações.

5.6.1 Pacotes existente na Debian

Caso o programa que se deseja instalar já esteja empacotado na distribuição atual da Debian, geralmente é instalado em cada cliente (há exceções, veja 5.6.3).

Para efetuar a instalação do um pacote em cada cliente usamos dois utilitários: **apt-get** e **espalha** (5.7.2). No arquivo de entrada do **espalha** deve-se colocar uma linha no espírito de:

```
apt-get update  # opcional, depende se houve atualização no espelho
apt-get install pacote-novo
```

Depois, basta rodar o **espalha**, e verificar se a execução ocorreu corretamente em todos os clientes para que, novamente, não ocorra falta de sincronia entre os clientes da rede.

5.6.2 Instalação no `/usr/local`

O `/usr/local/` é um diretõrio compartilhado a partir de um servidor (7.1.1). É possível fazer a instalação de programas dentro da estrutura do `/usr/local` no respectivo servidor. Este tipo de instalação é mais rara do que a instalação vista em 5.6.1.

O principal motivo para a instalação de programas no `/usr/local` é a inexistência de pacotes Debian para o programa em questão. Nesse caso, a compilação e configuração precisam ser feitas "manualmente". Para evitar o extensivo trabalho de realizar esse processo em cada máquina, o que também seria indesejável por dificultar a sincronização da rede (veja 5.4), a instalação é feita apenas no `/usr/local/`.

Outra razão está descrita na seção 5.6.3.

Para a instalação, deve-se obter o código fonte do programa e colocá-lo no diretório `/usr/src/` do servidor de `/usr/local/` (7.1.1). Deve-se então compilá-lo num subdiretório. Após a compilação, verifique se o pacote fonte já fornece um mecanismo de instalação automática (ver Makefile, README, INSTALL, etc), e configure a instalação para ser efetuada no `/usr/local/`. Se não houver tal mecanismo, é necessário fazer a cópia das partes relevantes para as estruturas de diretório abaixo de `/usr/local/`.

Para uma solução alternativa quando o pacote não existe na Debian, mas se quer fazer a instalação nos clientes individualmente, veja a seção 5.6.4.

5.6.3 Estudo de instalação de pacote: tamanho × utilização

Há uma exceção para a regra de instalar o programa em todos os clientes quando ele estiver disponível em pacote Debian: quando o programa em questão é muito grande, ocupando muito espaço em todos os clientes, e não é frequentemente/popularmente utilizado. Neste caso, deve ser feita a instalação no `/usr/local/`, que é distribuído na rede.

Veja que os dois lados devem ser balanceados para que se chegue numa conclusão de modo a otimizar os recursos da rede. Por um lado, temos que um programa grande executado a partir do diretõrio remoto faz com que a rede fique mais congestionada e, além disso, pode sobrecarregar o servidor de `/usr/local`. Por isso, para este tipo de instalação devem ser escolhidos programas com pouco uso dentro da rede. No caso de decisão desse tipo de instalação veja 5.6.2.

Por outro lado, se instalarmos um programa excessivamente grande nos clientes e ele não for muito utilizado, ocorre um um desperdício de espaço em todos os clientes. No caso de decisão desse tipo de instalação, veja 5.6.1.

5.6.4 Geração de pacote seguido de instalação local do próprio

Quando não há pacote Debian para um programa, e se queira instalá-lo em todos os clientes, uma solução é criar o pacote Debian. Essa geração de pacotes é feita através do próprio **dpkg** com a opção *-b, --build.*

Dessa forma, ganha-se um controle muito maior sobre gerenciamento de versões, instalações e remoções. E todo esse controle será feito automaticamente pelo **dpkg**.

Após criado o pacote, ele deve ser instalado como qualquer pacote comum (ver 5.6.1).

5.7 Software Caseiro

Aqui estão relacionados e documentados os programas (scripts, em sua marioria) confeccionados pelos próprios administradores da rede. Cada programa aqui tem um propósito distinto, mas todos têm em comum o objetivo de facilitar a vida administrativa.

5.7.1 `/usr/local/Instalacao`

Esse diretório armazena vários scripts e arquivos para instalação de software nos clientes de nossa rede. Um exemplo é o script **espalha** que será explicado na seção 5.7.2.

5.7.2 Espalha

O `espalha` é um sistema destinado a executar automática e simultâneamente uma série de comandos (um script) em todos as máquinas da rede, ou em algum grupo de máquinas. A vantagem de utilizar o `espalha` é a facilidade para instalar programas, alterar dados, etc, em todos os clientes desejados. O sistema é relativamente simples e necessita de apenas três arquivos, citados abaixo. É importante lembrar que este script deve ser rodado a partir do servidor de `/usr/local/` (veja 7.1.1), que é o único que tem "livre acesso" aos clientes, i.e., o *root* neste servidor não precisa de senha para entrar como *root* nos demais.

- `/usr/local/Instalacao/scripts/espalha` Este é um script em **bash** que será executado nos clientes. Nele pode-se colocar qualquer tipo de instruções. Alternativamente, pode-se alterá-lo para rodar como um script em **perl**, porém isto normalmente não é feito, já que é um script muito utilizado para tarefas simples, e como pode ser usado nos primeiros estágios da instalação, nem sempre se pode requerer que o **perl** esteja instalado.

- `/usr/local/Instalacao/executa*` Há uma série de scripts com o prefixo `executa` que faz a parte de rodar o script **espalha** acima. Há um script a ser rodado em modo texto e outro em ambiente gráfico. Com este é possível abrir vãrios terminais concorrentes, cada um conectado a um cliente remoto, de modo que o espalhamento seja feito em paralelo. Fica a cargo do administrador escolher qual script é mais adequado numa determinada situação. Todos estes scripts lêem da entrada padrão uma lista de nomes de clientes, nos quais o script **espalha** será executado.

- `/usr/local/Instalacao/clientes.*` Este arquivo é opcional e serve para facilitar a entrada dos nomes dos clientes nos quais se deseja executar o **espalha**. Estes arquivos contêm os nomes dos micros que estão em cada uma das salas que compõe a rede. Exemplo de uso:

  ```
  $ ./executa < clientes.126
  ```

Isto executará o script **espalha** em todos os micros cujos nomes estão contidos no arquivo `clientes.126`. Espera-se que neste arquivo estejam os nomes dos micros da sala 126-A.

Para gerar todos os arquivos do tipo `clientes.*` pode-se executar o comando:

```
$ for i in ‘ypmatch clients126 netgroup‘; do
    echo $i | cut -d\. -f1 | tr -d \(
  done
```

5.7.3 Geraarchive

Come descrito na seção 5.6.4, podemos criar diversos pacotes para instalarmos programas e configurações nos clientes de forma que a instalação fique mais homogênea. Para que o esses pacotes fiquem registrados no banco de dados do `dpkg`, de modo que sejam facilmente encontrados pelo comando `apt-get`, é necessário incluir no arquivo `sources.list` (da configuração do apt) um diretõrio que contenha um arquivo com a descrição de todos os pacotes naquele subdiretório.

Este pacote deve ter o nome de `Packages.gz` e deve ter um formato específico (leia manual do *deb-control*). O script `geraarchive` automaticamente gera este arquivo. O script fica em `/usr/local/localdeb/`. Os pacotes .deb devem ficar em `/usr/local/localdeb/dists/ nome da distribuição /local/binary-i386/`.

Após adicionar a linha:

```
deb file:/usr/local/localdeb potato local
```

no arquivo `/etc/apt/sources.list`, deve-se executar o comando *apt-get update*. Assim, todos os pacotes locais estarão no banco de dados do dpkg e poderão ser instalados.

5.7.4 Manutenção de contas

Aqui descrevemos os scripts fabricados para abertura, fechamento, suspensão e cancelamento de contas. Todo acesso ao `passwd` do `NIS` e toda manipulação de contas de usuários está encapsulado por esses scripts. Assim, **não se deve** manipular o `passwd` e correlatos diretamente. Os scripts, todos escritos em Perl, estão em `vila:/root/scripts/`.

Primeiro, serão explicados os scripts referentes a abertura de contas. Depois, de suspensão temporária, mudança de grupo e por fim, fechamento da conta.

Os scripts são descritos a seguir:

abrir Este script serve para abertura de contas dos novos usuários. Ele apenas precisa do número USP do usuário e o login escolhido por ele. Deve-se orientar o usuário a escolher um login apropriado. O próprio script verifica se o aluno está matriculado, qual o seu curso (a fim de colocá-lo no grupo correto) e nome completo. O programa verifica o arquivo `vila:/var/etc/user_info`, que dispõe de três campos do seguinte formato:

`Número USP:Nome completo:Data de entrada no IME`

Ao final da execução, o script imprimirá uma folha na impressora da sala 125 contendo o login do usuário, uma senha gerada aleatoriamente pelo script (usando comando **pwgen**) e uma mensagem inicial contida em **vila:/etc/welcome**.

Este script executa mais algumas ações além de mexer no arquivo de senhas da rede. Veja o próximo item para mais detalhes.

ulots.pl Este script também serve para abertura de contas, mas é usado apenas excepcionalmente, pois não faz checagem de dados (como nome, matrícula no IME, número USP). Este script é usado em casos de abertura de conta especiais. Por isso, as verificações padrões não são feitas neste caso. **Cuidado**: saiba o que está fazendo ao usar esse script. Na dúvida, use o anterior.

Na verdade, o script *abrir*) é um front-end para este script, que também imprime a mensagem ao final da execução.

Aqui vai um esquema passo-a-passo com todos os arquivos criados ou modificados para a inserção de um novo usuário:

1. O arquivo de senhas da rede `vila:/var/etc/passwd` ganha uma nova entrada, tomando cuidado para verificação de *lock* (arquivo `/etc/ptmp`) e de que o login é único.
2. Cria o diretório **$HOME** do usuário copiando os arquivos em `vila:/etc/skel/` para `vila:/home/$grupo/$login/`, acertando as permissões de todos os arquivos.
3. Adicionar o usuário no arquivo de email alias do qmail `vila:/var/qmail/alias/.qmail-users-$grupo` (de acordo com grupo) para que este usuário receba emails enviados a *users-$grupo@linux.ime.usp.br*.
4. Arruma a *quota* desse usuário de acordo com seu grupo, copiando a configuração de quota do usuário *pad$grupo*.
5. Cria o arquivo utilizado pelo contador disponível no **cgi-bin**.
6. Cria todos os arquivos necessários pelo sistema impressão (ver 5.7.9).

suspender.pl Este script serve para suspensão de um usuário. Ele arruma a entrada no `passwd` do `NIS` (colocando um * na frente da senha criptografada) e escreve no arquivo `vila:/root/Contas_Suspensas/`login_do_usuário o motivo dado para a suspensão (senha fraca, mau comportamento, etc) e a data em que a suspensão ocorreu (e qualquer outro dado que o operador achar relevante). Assim, é sempre possível consultar o histórico do usuário.

reabrir.pl Este script serve para reabrir a conta de um usuário previamente suspenso. Ele também escreve uma mensagem no arquivo `vila:/root/Contas_Suspensas/`login_do_usuário, relatando a data da reabertura de conta. Automaticamente, uma nova senha é gerada (também usando o programa **pwgen**), e de novo são impressos o login do usuário, a senha e uma mensagem. Essa mensagem é diferente da mensagem de abertura de conta, e está contida em **vila:/etc/senhas** (como geralmente a suspensão de conta se dá devido à senhas fracas, esse seria uma folheto de informações sobre como montar senhas "boas").

Mesmo não sendo o uso correto desse script, ele é utilizado para trocar a senha do usuário que esquece sua senha, ou quando há falha na geração de senha pelo script de abertura de conta. Usamos este script porque ele já faz verificações sobre locks sobre o *passwd* sendo usado, entre outras coisas. Assim, o operador não terá que se preocupar com esse detalhes.

assusta.pl Quando a lista dos alunos matriculados no IME é publicada, temos que fazer essencialmente duas modificações em relação às contas dos alunos da rede.

A primeira modificação é cancelar a conta dos alunos que não constam mais como matriculados. Para isso, criamos um esquema para fazer o fechamento das contas, que está descrito abaixo.

A outra modificação é mudar o grupo dos alunos que tiveram a matrícula modificada aqui no IME. Também existe um esquema para isso que faz essa mudança de grupo.

Em particular, este script serve para fazer o **aviso** dessas modificações para os usuários em questão. Isso mostrou-se bastante útil por vários motivos. O primeiro é que os scripts que detectam a necessidade de cancelamento e de mudança de grupos não são sempre perfeitos. Um problema que tivemos é que o sistema de matrícula da USP registra os nomes dos alunos às vezes de forma diferente de nós (por fazerem abreviações nos nomes). Este aviso serve para que os próprios alunos nos avisem que este problema existe e podemos consertar os erros. Outra vantagem de usar esse script, é o próprio aviso ao usuário, que tendo o aviso poderá se planejar para as mudanças que ocorrerão.

Na verdade existem duas versões desse script, uma para mudança de grupo e outra para cancelamento de conta, `/root/grupos/assusta.pl` e `/root/fechamento/assusta.pl`, respectivamente. Mas eles fazem essencialmente as mesmas coisas só que mandam mensagens diferentes.

mudagrupo.pl Quando um aluno muda de curso no IME, isso acarreta numa mudança de grupo de sua conta na rede. Em particular, isso é comum entre os ingressantes do grupo básico de matemática, que no início no segundo ano têm de se matricularem num curso dentre matemática aplicada, matemática pura, ou estatística. Assim, esse script é muito útil, visto que este cenário é comum aqui na rede.

Este script recebe dois argumentos: o **login do usuário** e o **novo grupo** do usuário. Na verdade, este script não faz nada além de fazer verificações de parâmetros (se o usuário e grupos estão válidos) e permissões. Todo as ações mais relevantes acontecem na chamada para o script `mgrp.pl` que está descrito abaixo.

mpgrp.pl Este é o script que efetivamente muda o grupo de um usuário. Entretanto, este script **não** deve ser utilizado diretamente. O script que a ser usado é o `mudagrupo.pl` (veja acima).

Primeiro, este script cria cópias de segurança dos arquivo `passwd` e `.qmail-users-*`. Depois, as mudanças são análogas às listados no script `ulots.pl` (para abertura de contas). Mas ao invés de inserir novas entradas, apenas se troca os arquivos e dados nos arquivos necessários para a mudança de grupo. Por exemplo, o home do usuário não é destruído para se criar um novo (a partir no skel), mas apenas mudado o local e permissão.

cancelar Como discutido anteriormente, todo aluno eventualmente terá sua conta fechada. Este script é utilizado para o fechamento. O script tem apenas um argumento, o **login do usuário**. Este script em si não faz nada além da checagem da existência do usuário na rede (via comando **ypmatch**), e depois utiliza o script abaixo (`excluir.pl`), para realmente fazer o trabalho.

excluir.pl Este script é o que faz todo o trabalho de fechamento da conta do usuário. **Este script não deve ser chamado diretamente**, o correto é utilizar o script descrito acima (`cancelar`). O que o script faz é apagar todos os arquivos e desfazer as modificações feitas durante a abertura (ver o script `ulots.pl`). Antes do início das modificações/remoções são feitos backups dos arquivos importantes para a integridade do sistema. Assim, se algum erro ocorrer, podemos reaver todos os dados necessários em pouco tempo.

Os arquivos de que são feitos backups são:

- /var/etc/passwd - O arquivo de senhas da rede Linux.
- /var/qmail/alias/.qmail-users* - Os arquivos que fazem redirecionamento de emails para todos os usuários de um determinado grupo.
- /root/Contas_Suspensas/$login - O arquivo de log que registra todas as vezes que um usuário foi suspenso, e o motivos.
- /home/$grupo/$login - O **HOME** do usuário (todos os seus arquivos). Isso é muito importante, pois muitas o usuário precisa de algum arquivo da sua conta após esta ter sido fechada.

compara Este script serve para criar o arquivo de entradas para o script `cancelar`. Ele compara a lista dos usuários da rede com a lista de matriculados no IME. Ele devolve os usuários que não estão na lista de matriculados, ou seja, são usuários que possivelmente terão suas contas fechadas.

5.7.5 Script para backup

Por utilizarmos o sistema de arquivos padrão do Linux (ext2), não há uma maneira confiável de recuperar arquivos apagados acidentalmente pelo usuãrio, ou perdidos devido a degeneração ou falha do dispositivo físico que contém tais arquivos.

Assim, backups constantes são muito importantes. Há dois conjuntos de arquivos a serem protegidos desta maneira: os arquivos de configuração e programas de nossos servidores, como por exemplo o diretório `/etc`, scripts, etc; e os arquivos dos usuários (os homes).

Os backups devem ser mantidos em outro meio físico (que não os próprios discos). Atualmente, usamos fitas magnéticas tipo DAT. Portanto, todos os arquivos são guardadas de forma consecutiva (a fita não tem sistema de arquivos), e os scripts abaixo utilizam os comandos e devices para a escrita nas fitas que dispomos.

Um projeto posterior seria fazer os backups em um outro disco comum para que o backup seja feito de forma mais fácil e rápida. Esses arquivos ficariam disponíveis aos próprios usuários. Infelizmente, atualmente não temos os recursos necessários para isso.

Home dos usuários Até hoje, em todas as ocasiões em que realizamos recuperação de dados para usuários, o motivo foi o uso negligente do comando **rm**. Como muitos dos usuários não são familiarizados com o ambiente Unix, é relativamente comum cometerem erros ao apagarem arquivos. Assim não é raro um usuário pedir recuperação de arquivos.

O backup dos homes dos usuários deve ser feito **diariamente**, pois normalmente há muitas alterações de arquivos por dia, seja por recebimento de emails ou edição de EPs, e é extremamente indesejável que essas mudanças sejam perdidas. Por outro lado, queremos ter os homes num estado bastante diverso (cronologicamente) assim não validamos alguma mudança que o usuário não desejava. Pensando nisso fazemos os backups **completo** uma vez por semana, e backups **incrementais** diariamente.

Configuração do servidores Por serem manejados por usuários mais qualificados (apenas por administradores) e por sofrerem bem menos mudanças que os homes dos usuários, estes arquivos não precisam ter backups tão frequentes. Assim, os backups dos servidores são feitos mensalmente.

5.7.6 Killer

Ás vezes, por falha dos programas envolvidos, ou por desleixo do usuário, mesmo após este se deslogar seus programas (*processos*) não são terminados. Alguns casos, como o programa **wget**, não são considerados problemáticos, mas programas como o **netscape** que consome memória, processamento, e banda de rede, devem ser considerados. Antes da existência do **killer**, a administração recebia dos usuários inúmeros pedidos para finalizar esses processos "zumbis". Como atender esses pedidos era trabalhoso e inefetivo, criamos o script **killer**, cujo objetivo é finalizar periodicamente todos os processos de usuários que estejam rodando "indevidamente".

O pacote **killer** contém baiscamente três arquivos. O funcionamento e propósito de cada um deles estão detalhados a seguir.

- `/usr/sbin/killer/programas`

  Este arquivo deve conter um nome de programa por linha. Estes programas devem ser programas "candidatos" a não finalizarem corretamente. Por exemplo, o conteúdo do arquivo poderia ser algo do tipo:

  ```
  netscape
  GoodStuff
  emacs
  gimp
  ```

  Mas, por exemplo, não colocaríamos um **wget** nesta lista, ao menos que a administração decida que este programa não deve ser utilizado sem que o usuário esteja logado na máquina. Se for detectado que algum outro programa não estiver sendo finalizado corretamente e o desejo é que o programa não seja rodado pós *logout* então deve-se adicionar este programa à esta lista para que o **killer** o verifique e o finalize, caso necessário.

- `/usr/sbin/killer/killer`

  Este arquivo é só um "wrapper" entre o script `mata.pl` e o arquivo texto `programas`. O que ele faz é, para cada linha do arquivo programas (que deve conter um nome de programa) chama o script **mata.pl** com o parâmetro da linha atual do arquivo `programas`. Assim este sisteminha deve verificar e finalizar todas as ocorrências "ilegais" de todos os programas listados no arquivo `programas`.

  O pacote **killer** já insere no *crontab* do *root* uma linha para que o script **killer** seja rodado a cada cinco minutos.

- `/usr/sbin/killer/mata.pl`

  Este script é o mais importante do sistema, pois é ele que realiza as "ações" do sistema. Este script recebe um argumento, que é o nome do programa que estamos querendo verificar se deve ser finalizado. O script faz o basicamente o seguinte;

  1. Verifica se existe um processo com uma instância do programa em questão.
  2. Verifica se o usuário dono do processo está logado.
  3. Se estiver, cai fora, pois não queremos finalizar nenhum processo de usuário que esteja logado "licitamente" no cliente.
  4. Se não estiver, então finaliza o programa, em dois passos, primeiro manda um sinal **SIGTERM** e depois de dois segundos manda um sinal **SIGKILL**. Depois manda um email para o usuário explicando o que aconteceu com seu processo e também deixa um registro em `/var/log/killer` para que seja posteriormente visto quais processos foram finalizados pelo **killer**.

5.7.7 Estatísticas. Scripts variados que realizam as estatísticas do uso dos recursos da rede

5.7.8 Scripts relacionado com emails

O uso de email é uma parte bastante importante na rede já que para muito usuários é a umas das partes mais importante no acesso à internet que eles possuem. Aqui iremos mostrar alguns scripts que ajudam o manejamento de emails, tanto voltado para a vida do usuário, quanto voltado para a administração dos servidores de email.

popmail Muitos usuários desejam utilizar o serviço de **POP** para a leitura dos emails que estão na rede de suas casas. O **POP** atualmente apenas funciona como o formato *maildir* de guardar emails, e o padrão customava ser *mbox* que guarda todos os emails no arquivo `$HOME/Mailbox`. Para fazer essa mudança é que foi criado este script.

Aqui na rede utilizamos o **qmail** como programa que gerencia o sistema de emails. E o entregar do email do qmail, o **qmail-local**. Este segue as regras que estão determinadas no arquivo `$HOME/.qmail` do usuário (ver página manual do **dot-qmail**). Portanto utilizamos este arquivo de regras para mudar o comportamento da entrega de email para este usuário. O script é simples e executa os seguintes passos:

- Verificar a existência do arquivo `$HOME/.qmail` e caso ele exista, movê-lo para `$HOME/.qmail-old` para que dados não sejam perdidos.
- Criar um novo arquivo `$HOME/.qmail` com o conteúdo:

  ```
  ./Maildir/
  ```

  Para que os novos emails sejam entregues no diretório `Maildir/` do home o usuário e no formato *maildir* (observe a última barra no nome do diretório).
- A verificação da existência de um diretório `Maildir/` no home do usuário, e caso ele não exista, então utiliza-se o script **maildirmake** para a criação do diretório, com permissões e subdiretórios apropriados.

nopopmail Este script serve, justamente, para reverter as mudanças feitas se usado o comando **popmail**. O que ele faz basicamente é tirar o arquivo `$HOME/.qmail`, assim fazendo com que o **qmail-local** volte a entregar os emails no format *mbox* no arquivo `$HOME/Mailbox`.

Portanto este script executa os seguintes passos:

- Move o arquivo `.qmail` para `.qmail-old` caso existe no home do usuário.
- Transforma todos os email existentes (no formato *maildir*) para o formato *mbox* utilizando o script **maildir2mbox**.
- Por fim, remove o diretório `Maildir` do home do usuário.

pinq Este é um script para que os usuários que utilizem o **pine** que estão com configuração **POP**, ou seja, estão recebendo seus emails em format *maildir*, consigam continuar a ler email utilizando o **pine**. O **pine**, na verdade, apenas tem suporte para leituras de arquivos do tipo *mbox*, então o script basicamente transforma os emails atuais (de tipo *maildir*) em formato *mbox*, utilizando o script **maildir2mbox**.

O script é extramente simples, aqui está ele, na sua íntegra:

```
#!/bin/sh
/usr/local/bin/maildir2mbox && exec pine ${1+"$@"}
```

users.sh Assim que um usuário é registrado na rede, ele ganha uma entrada num redirecionamento de email, conforme seu grupo, do tipo *users-$grupo@linux.ime.usp.br*. Esse redirecionamentos estão explicitados do home do usuário *alias* usado pelo **qmail**. Os arquivos em estão são `vila:/var/qmail/alias/.qmail-users*`. Em primeiro momento não fazíamos restrições a estes endereços, inclusive o email *user@linux.ime.usp.br* que manda email para todos os usuários de todos os grupos.

Mas logo decidimos que esse email deveria ser "protegido" para que apenas os administradores e os professores pudessem enviar email (isso foi decidido principalmente pelo mau uso dos próprios alunos deste endereço). Assim foi necessário criar um script para verificar se o usuário que enviou o email é autorizado ou não.

Para isso utilizamos o fato do **qmail-local** permitir o uso de programas, e em especial, a análise de seu código de erro para a verificação do endereço que está enviando o email (ver página manual de **qmail-command**).

Atualmente este script verifica se a pessoal que está mandando o email consta na lista de usuários admins (`vila:/var/qmail/alias/.qmail-admin`) ou professores (`vila:/var/qmail/alias/.qmail-users-prof`).

users-grupo.sh Após algum tempo da criação do script anterior, percebemos a necessidade de restringir os emails entre os diferentes grupos. Assim um usuário do grupo *bcc* não poderá mandar emails para outros grupos.

Para isso o script verifica se o usuário que enviou o email é um dos usuários autorizados (admin, professores, ou pessoal com alguma função especial que estão listados em `vila:/var/qmail/alias/.autorizados`), ou é pertencente ao grupo que corresponde ao grupo que estará recebendo o email.

5.7.9 Sistema de Quotas de Impressão

O sistema de quotas de impressão foi criado aqui na rede devido a um problema de escassez de recursos que temos aqui no Instituto. Tanto o número de folhas quanto o número de tonners são bastante controlados pelo Instituto então a única saída que encontramos é criar este sistema que permite com que o usuário imprima apenas uma quantidade de páginas por mês.

O sistema de impressão é feito apenas para nossa impressora laser 7.2. Neste sistema cada usuário tem uma quota de páginas fixas por mês e o sistema vai deixando imprimir até a quota acabe. Abaixo segue um esquema para começar a entender o sistema, entretanto é necessário destacar que vários aspectos do sistema não é coberto aqui será exposto apenas quando se olha os códigos dos scripts envolvidos.

Sobre o lpr Primeiramente é importante destacar o que o lpr faz ao ser chamado para o impressora **varzea**. A informação abaixo foi retirado do arquivo `/etc/lpd_printcap` no 7.1.1.

```
:as=|/usr/local/printers/filtros/as -Pvarzea:
:ae=|/usr/local/printers/filtros/ae -Pvarzea:
:if=/usr/local/printers/filtros/filtro.varzea -Pvarzea:
```

- as - Accounting at start. É o primeiro script que roda, serve para dar uma verificada na impressora e arrumar umas coisinhas para podermos começar a imprimir.
- if - Filter. Este é o script onde as coisas realmente "acontecem". A quota é verificada, a impressão é realizada, a nova quota setada, etc.
- ae - Accounting at end. Este é o último script rodado, serve para "desfazer"as precarações feitas pelo **as** indicando que a impressão realizou-se corretamente.

Comunicação com a impressora Durante a impressão é necessária manter comunicações com a impressora para coleta de seus dados. Esta comunicação serve para, basicamente duas coisas, primeiro para verificar se a impressora está pronta para imprimir (isto que dizer, sem falta de energia, papel, tonner, etc). E depois para verificar quantas folhas foram realmente impressas do pedido, assim podemos avaliar quanto de quota devemos descontar do usuário.

Essa comunicação atualmente é feita pelo programa *npadmin*. Verificamos apenas o display da impressora que acusa se algum erro existe. Portanto executando o comando,

```
$ npadmin --display varzea
```

teremos o conteúdo do display da impressora **varzea**. Eventualmente esta comunicação pode ser feita por outros meios. Antigamente nós criamos um daemon em java para comunicar com a impressora.

Sobre a estrutura de diretórios do sistema O sistema de impressão acabou ficando muito grande e por isso pode parecer complexo de entender, por isso vamos tentar explicar a estrutura de diretórios que foi utilizado no nosso sistema. Assim ficará mais fácil, posteriormente descutir o que cada script faz.

`/usr/local/printers/` - diretório "raiz"do sistema de impressão.

- `filtros/` - Aqui estão os filtros de impressão (**as, filtro, ae**) e outros scripts auxiliares para os filtros.

- `scripts/` - Aqui estão alguns scripts relacionados ao sistema de quotas; como inicializar as quotas do mês, ou gerar todos os arquivos e diretórios do sistema (**atualiza_mes, inicializa_quotap, montaquota**). Também existe um utilitário interessante **soma_do_mes** que escreve quantas folhas foram impressas naquele mês (atual).

- `logs/` - Neste diretório estão os "logs"de tudo que acontece que o sistema de impressão. Só que na verdade aqui é guardado muito mais do que os logs (isto será visto mais a frente na exposição dos scripts).

  - `cancelados/` - informações e diretórios onde ficam os usuários cancelados (esse diretório não é implícito no sistema, mas feito "na mão").

  - `quotas/` - Esses são os arquivos de quotas como devem ser quando estão ZERADOS. Existe um arquivo para cada usuário da rede. O formato desse arquivo é:

    O arquivo de quota consiste de 3 números. O primeiro é o número de páginas já impressas neste mês. O segundo corresponde ao número total de páginas imprimíveis por um determinado usuário. E o terceiro número é o número de páginas que algum admin "emprestou"para o usuário. (OBS: Esse "empréstimo"deve ocorrer sob condições bem específicas e o sistema se encarrega de retirar essa folhas do próximo mês).

    Então por exemplo se olhar no arquivo 'livio' posso ter o seguinte conteúdo:

    ```
    35/50/5
    ```

    Indicando que já imprimi 35 folhas. O máximo que posso imprimir (normalmente) é de 50 folhas. E que ganhei um empréstimo de 5 folhas NESTE mês. Ou seja AINDA posso imprimir $50 + 5 - 35 = 20$ folhas. Com o agravante de que se eu usar todas essas 20 folhas, no mês seguinte vou começar o mês:

    ```
    5/50/0
    ```

    Ou seja, já utilizei 5 das minhas 50 folhas para imprimir.

  - `tmp/` - diretório onde são guardados arquivos temporários durante alguma impressão.

  - `varzea/` - diretório onde é guardado todas as operações (impressões, etc) realizadas com a impressora **várzea**. Neste diretório existem outros diretórios que indicam a data (o mês) das impressões. Assim temos os logs separados por mês e temos um link (atual) apontando para o mês atual. Todos os diretórios estão no formato ano-mês (06-1999, 02-2000, 10-2003, etc).

- atual/ - No mês atual (que é na verdade um link para o diretório do mês atual, conforme descrito acima) temos todas as informaçãoes de impressões realizadas no mês. Tanto dos usuários quanto da impressora em geral.
  - `log` - Arquivo que loga todas as tentativas de impressão feita no mês.
  - `usuarios/` - Diretório que contém todos os usuários (em forma de subdiretório).
  - `usuarios/(`*username*`)/` - O diretório do respectivo usuário username. Neste diretório existem 3 coisas. O arquivo `log` que guarda todas as tentativas de impressão do usuário *username*. O arquivo `quota` que contém a quota do usuário conforme o padrão de arquivo descrito acima e um diretório `arquivos` que guarda uma cópia do arquivo recebido pelo script e uma outra cópia filtrada.

`/usr/local/quotap/` - diretório "raiz"do sistema de quotas.

- `scripts/` - Por enquanto temos aqui a fonte do programa **quotap** (que colocamos em /usr/local/bin) que lê o arquivo do usuário em /usr/local/quotap/usuarios/atual/(login_do_usuário) e imprime uma mensagenzinha para o usuário indicando sua quota de papel do mês atual.
- `usuarios/atual/` - Aqui existe um arquivo para cada usuário indicando sua quota. Isto é apenas uma cópia do arquivo `/usr/local/printers/logs/varzea/atual/(username)/quota`. Então se você quiser mexer REALMENTE na quota de alguém você deve mexer no arquivo */usr/local/printers/logs/varzea/atual/(username)/quota*, copiar a atualização: */usr/local/quotap/usuarios/atual/(alguém)*

Bom agora que demos uma olhadinha na estrutura de diretórios vou começar a explicar o que os scripts (as, filtro, e ae) fazem.

Sistema de locks na impressão Para tentarmos sempre deixar o sistema mais "consistente"possível foi criado um sistema de lock que indica se existe uma impressão atualmente ou que a última impressão realizada não foi finalizada corretamente. Esse **lock** é criado em /usr/local/printers/logs/varzea/atual/.lock Nele temos 6 linhas:

- Um número maluco que na verdade é um 'date +%s' (data em segundos desde 1970). Apenas serve como controle.
- O username da tentiva da impressão.
- O número de páginas na tentativa de impressão.
- A nova quota desse usuário (já descontados a essa impressão).

- O contador tipo 1 da impressora.
- O contador tipo 2 da impressora.

Esse lock serve para sabermos, na próxima impressão, e a impressão apresentou algum erro e não foi "corretamente"finalizada. Com esse lock logamos o erro possível para que se algum usuário reclamar podemos ter um log verificando sua reclamação.

Mais importante ainda serve para verficarmos, ao final da impressão, se a impressora realmente imprimiu a quantidade de páginas desejada pelo usuário. Para isso verificamos os contadores do lock e novamente "perguntamos" para a impressão seu contador atual para verificar se está consistente.

Caso haja alguma inconsistência devolvemos ao usuário o número de folhas que aparantemente não foram impressas.

Para cuidar desse lock temos um script que é utilizado pelo **as, ae**. Esse é o script `arruma_lock.pl`. A sua função é determinada pela opção *-s* ou *-e*.

- *-s* A opção -s é utilizada pelo **as** para verificar se o lock ainda existe (se existe isso é indicador de algum erro). O que ocorre neste caso é que o lock é lido e todas suas informações são impressas no *log* da impressora, e depois o lock é apagado.
- *-e* A opção -e é utilizada pelo **ae** para verificar se o número de páginas impressas foi conforme esperado (ou seja) o contador subiu na mesma quantidade que o número de páginas enviadas pelo usuário. Consertamos aqui qualquer inconsistência (devolvendo a quota) e apagamos o lock.

Os scripts em si (as, filtro e ae) Agora explicaremos, mais ou menos por cima, o que cada script faz. Todos os scripts aceitam dois argumentos IMPORTANTES. O primeiro é o *-P**impressora*** que obviamente indica qual a impressora. Depois temos a opção *-n**usuário*** que indica qual o usuário que está tentando realizar a impressão.

- **as** Primeiro chama-se o `arruma_lock.pl` com a opção *-s*. Depois pegamos os contadores iniciais da impressora. Note que existem dois contadores. O primeiro diz quantas páginas foram impressas nesta impressora (desde sempre) e o outro diz quantas páginas foram impressas desde que a impressora foi ligado por último. Para pegar esses contadores utilizamos a função `&pega_contador()` no módulo `funções_comuns.pm`. Imprimimos todas as informações nos logs devidos e caímos fora.
- **filtro** Aqui é feita toda a parte REAL da sistema (inclusive a impressão). Primeiro pegamos o arquivo dado pelo usuário e filtramos ele com o filtro `/etc/magicfilter/varzea-filter` que filtra os arquivos para impressão. Depois de filtrarmos os arquivos pegamos o postscript resultante e contamos suas páginas. Aqui utilizamos dois métodos para contagem de páginas, se a contagem der errada então a impressão é cancelada e um email é mandado para o usuário explicando o

motivo do cancelamento. Depois disso o arquivo do usuário já está liberado para imprimir, então basta verificar a quota. Se o usuário não tiver quota de papel suficiente então a impressão é cancelada e o usuário recebe um email, também explicando que sua quotap acabou, caso contrário a nova quota é setada. Antes que a impressão seja realizada o `lock` é colocado (conforme explicado na seção anterior). Depois disso logamos mais umas coisas no log e mandamos imprimir.

- **ae** Aqui o lock é verificado para pegarmos casos em que a impressão, por algum motivo, imprimiu menos folhas do que o esperado. Agora chamamos novamente o `arruma_lock.pl` desta vez com opção *-e*. Lá primeiro verificasse as informações do lock, como contador da impressora antes de começar a imprimir e número de páginas que o usuário queria imprimir. Então perguntamos para impressora novamente qual é valor do contador. Para verificar a corretude da quantidade de quota retirada basta verificar se o contador novo menos o contador velho resulta no número de páginas que o usuário mandou imprimir.

  Se for então tudo ocorreu bem então podemos finalizar este impressão. Caso contrário, verificamos quantas folhas faltou para o fim da impressão, e devolvemos esta quantidade da quota do usuário, já enviando-lhe um email a respeito do problema.

6 Segurança

A parte de segurança está dividida em duas partes. A segurança interna, que abrange os cuidados necessários dentro da rede, e a segurança externa, que são os cuidados necessários na interação das partes de dentro da rede com a parte da fora (Internet).

6.1 Interna

6.1.1 Física

Aqui classificamos como segurança "física" toda a parte referente a possíveis ataques devido ao acesso físico que os usuários tem a certos elementos da rede, como por exemplo clientes e cabeamento. Deve-se tomar certos cuidados para que terceiros não usem da disponibilidade desses recursos para ter acesso não autorizado a informações restritas.

Clientes e Servidores

- Colocar boas senhas na bios.
- Desligar drives na bios. O Linux costuma ignorar várias coisas da bios, e detecta os drives mesmo que estes não estejam habilitados na bios.
- Se os gabinetes permitirem, manter cadeados p/ que não se tenha acesso fácil a parte interna dos micros.

6.1.2 Lógica

Na parte "lógica", entram coisas mais ligadas ao S.O. usado. Vamos dividir em duas partes, uma para cliente e outra para os servidores.

Cliente

- O acesso remoto de um cliente para o outro é feito por ssh. Além de fazer um bom tunelamento para programas que rodam no X, tudo corre pela rede criptografado.
- As senhas locais (basicamente o root) são todas "escondidas" atráves de shadow.
- Colocar senhas no lilo, para que o usuário não consiga ter acesso a outras níveis de inicialização no prompt do Lilo.

Servidor

- Usuário não loga nos servidores nível I, nem no console, nem remotamente.
- Não deixamos servidores de cgi disponível para acesso externo. Existe um servidor que pode ser enxergado internamente, que monta os homes read-only. O problema maior como o cgi é que ele roda no servidor, e pode-se usar isso para pegar informações neste ou até derrubar o servidor.
- Regularmente (1 vez por mês), rodar um cracker no arquivo de senhas dos usuários. Não chutando senhas aleatórias, mas procurando coisas problemáticas, como senhas só de números, só de letras minúsculas, e coisas afins. Se o cracker quebrar alguma senha, a conta deste usuário deve ser suspensa até que ele venha pessoalmente na administração, onde terá a conta reaberta com uma senha nova, além de um esclarecimento sobre o problema de ter senhas fracas.
- As senhas dos usuários não são guardadas em shadow nos servidores, e sim num arquivo separado. Quando o nis exporta o passwd para os clientes, ele "simula" um shadow, de modo que um root no cliente não tem acesso ao mapa.

6.2 Externa

6.2.1 Firewall

Especificações a que deve atender o firewall da rede, que fica no servidor A:

- Liberar as portas de web, pop3, ssh, que são redirecionadas para os respectivos servidores destes serviços, e logados;
- Liberar a porta de smtp e ssh pro próprio servidor A, tudo logado;

- Bloquear todas as outras portas, logando tudo.
- Ignorar (DENY) protocolos ruindows que correm pela rede pois ela pode não ser isolada de outras sub-redes do IME.
- Logar todas as conexões feitas de dentro (rede interna) pra fora da rede.
- Os logs devem ficar em um arquivo separado, uma sugestão é o kerninfo.log. **Não deixar os logs do firewall no syslog**.

6.2.2 SSH

O acesso de fora para dentro da rede e entre os micros da rede deve ser feito por ssh. E sempre deve haver um micro na sala dos servidores dedicados a receber a conexão redirecionada do servidor A, que é o único na rede com ip real. No caso seria o servidor E.

6.2.3 Masquerading

Toda a rede terá ip's "falsos". Apenas um único servidor, no caso o servidor A, terá ip real e servirá de gateway para conexão da rede interna com o exterior. Pra isso, será usado o recurso de IP-Masquerading do Linux.

6.2.4 Sem CGI no APACHE

O Apache não deixará disponível aos usuários cgi, php e afins. Evita-se assim abrir buracos no meio do apache através de scripts mal feitos dos usuários.

7 Estrutura da Rede

A documentação da estrutura da rede é dividida em duas partes: a física e a lógica.

7.1 Estrutura física

7.1.1 Servidores

Nível I

A (spinoza) Esse servidor é o único com ip real na rede. Ele recebe todo o tráfego vindo de fora, fazendo filtragem de pacotes e IP-Masquerading para os cliente. Redirecionará os serviços de ssh, web e pop para os servidores E, B e C, respectivamente. Tudo isso é feito no script /etc/init.d/gateway, as regras estão na seção de segurança. Terá um servidor de ssh nele, em outra porta, pra receber conexões diretamente pra ele (pra efeitos de manutenção remota), e terá um servidor de mail rodando. Esse servidor de mail pega os mails de fora e manda pro servidor C, e recebe os mails do servidor C (destinados pra fora de rede), e faz o envio. Deverá tem um espaço grande alocado pra logs e para a fila do servidor de mail, pois caso caia a luz, este servidor fica de pé guardando os mails que chegam na rede, até a luz voltar e tudo ser enviado pro servidor C. Esse servidor de smtp não poderá ser usando como cliente de relay, ou seja, os usuários não poderão de casa usar o smtp desse servidor para destinar mails através do browser.

B (bacon)

- DNS: Será o servidor de dns da rede. Resolverá os endereços internos e pegará os externos da rede. Todos os clientes farão requisições de resolução de nomes para esse servidor.
- NTP (Network time protocol): Todos os clientes usarão esse servidor como servidor de ntp. Assim teremos todos os relógios dos micros sincronizados, o que resolve muito problemas da datas de arquivos e mensagem. Esse servidor ntp se sincronizará com um servidor na rede externa, a critério da administração corrente. Uma sugestão é usar o servidor da Fapesp, ntp.ansp.br.
- `/usr/local` compartilhado. Este servidor compartilhará os programas que forem instalados em /usr/local. Esse diretório será exportado para o /usr/local de todos os clientes, que mapearão /usr/local para o automount.
- Sistemas de quotas de impressão: Este servidor será o responsável pela hospedagem de todo o sistema de impressão, centralizando o controle de todas as impressoras e seus respetivos sistemas de quotas.
- SQUID: Deve ser instalado aqui um squid, que fará um cache de todas as requisições www e ftp internas. Os clientes só poderão acessar a web através do squid, para melhorar a eficiência do cache e economizar banda.
- APACHE: Será o servidor web usado para disponibilizar as páginas dos usuários pra fora da rede.

C (vila) Basicamente é o servidor de disco da rede. Por conveniência rodamos o servidor de mail e nis nele.

- Homes: Nesse servidor ficarão abrigados os homes. Em baixo de /home existem diretórios com os grupos, (bcc, bas, be, etc.), e dentro desses diretórios os homes dos respectivos usuários. Esses subdiretórios em /home deverão ser exportados um a um, para que os cliente usem o automounter, montando só o “grupo” necessário.
- Aqui ficará também o servidor de mail principal da rede. Os clientes estarão enviando os mails para cá por QMQP (um protocolo do qmail), e aí decide-se o que fazer. Se ele é pra um usuário da rede, já faz-se a entrega em disco. Caso contrário,

o mail é enviado para o spinoza, que fará a entrega na rede externa. Aqui ficarão as listas internas (admin, p.e.)

- NIS: Aqui ficará o servidor principal de nis da rede. Os cliente buscarão nesse servidor o passwd, group, netgroup, mapas do autofs, configurações do lpd.
- Como esse é o servidor principal de disco, aqui será feito o backup, com pelo menos dois drives de fita disponíveis. Existem scripts comentados na sessão de scripts caseiros para automatizar as tarefas, mas tudo baseado no comando *tar*

Nível II

D (jacuzzi) -> CPU Esse servidor é o micro mais possante da rede, pelo menos até a confecção desse documento. A principal finalidade dele é deixá-lo a disposição dos usuários, para login remoto, de tal modo que programas mais pesados possam ser rodados nele. A finalidade principal é que compiladores e máquinas virtuais (p.e., java) possam ser rodados nele, ao invés dos clientes mais lentos que temos na rede.

E (epicurus) Micro dedicado a servir às conexões externas para a rede, via ssh. Só deve permitir login remoto de fora da rede, para que não haja sobrecarga nesse servidor.

F (kant) Servidor destinado a servir cgi e php pra rede interna. Monta homes read-only. Contém um servidor web, mas só dos clientes internos ele pode ser enxergado.

Nível III

G (giordano) Micro para eventuais testes. Instalação de novas distribuições, testes de algum novo serviço, installs show, e afins. É o micro da bagunça, que pode ser "detonado" e reinstalado várias vezes. Só deve permitir acesso dos administradores, pois não existe muita preocupação com segurança nesse micro.

H (limbo.ime) Esse micro não percente a rede linux, pois não tem ip nosso, e sua instalação não atende necessariamente aos nossos padrões. Tem um ip real. Mas fica aos cuidados da administração da rede, que faz a manutenção do espelho debian, que é muito útil para a rede e para todo IME. As páginas do gul também estão hospedadas nesse micro, que mantém também um hypernews.

7.1.2 Mapa Cabeamento

Não cabe aqui anexar uma figura com o cabeamento da rede. O pessoal da S.I. (sessão de informática) tem o mapeamento da rede inteira em papel. O importante é ter uma idéia de cabo as salas são cabeadas no IME. Cada sala (inclusive a 125) tem um ou mais hubs, conectados a um dos switches centrais da rede, que são todos interligados. Não existe isolamento física da rede. A rede linux e a do IME compartilham o mesmo barramento nos switches principais.

7.2 Estrutura Lógica

Serviços Disponíveis aos usuário

Home Todos usuários tem direito a um espaço em disco, que varia de acordo com as necessidade de cada curso, e com a ajuda de custo de cada curso para manter essa estrutura. Não existe exceção para aumento de quota pros alunos, e o valor da quota dos usuários deve levar em conta toda a estrutura necessária para manter essa cota. Deve-se olhar valor possível ocupado, valor total ocupado, qualidade dos discos onde ficarão as quotas, capacidade de fazer backup de todos os arquivos, capacidade do servidor aguentar a carga de compartilhamento de todos esses arquivos.

Mail (POP3) Os usuários tem acesso aos seus mails através de um servidor pop3, disponível dentro e fora da rede. Esse servidor pop3 roda no servidor C, e pode ser acessado de fora através de um redirecionamento de porta no servidor A.

Página da Internet Dentro do espaço em disco de cada usuário, existe um diretório www, onde os usuários podem colocar as suas páginas.

CGI Interno O usuário pode colocar cgi's em suas páginas, mas esses cgi's só podem ser rodados num servidor interno, e acessados de um cliente da rede.

Acesso externo via SSH O usuário pode se conectar via ssh na rede, através de shell.linux.ime.usp.br

Impressoras Existem a disposição 1 impressora laser com sistema de quotas para os usuários, e algumas impressoras matricias. Existe uma impressora na sala dos servidores para uso exclusivo da administração.

Conta Individual Cada usuário tem uma conta única, que não pode ser usada por outra pessoa, nem que esta seja estudante do IME.

Acesso Web Acesso irestrito a Internet.

Software desenvolvimento Acesso a ferramentas GNU de desenvolvimento, para confecção de eps para os cursos.

BCC O BCC tem algumas coisas a mais a sua disposição na rede, primeiro pelo maior uso que é necessário, já que 90% das matérias do curso pedem eps, e segundo porque a grande maioria dos equipamentos da rede foram cedidos pelo Departamento de Computação, para suprir as necessidades dos alunos de graduação do BCC. Alguns dos diferenciamentos:

- Quota em disco de 50Mb.
- Servidor de cpu (jacuzzi), para login remoto. É um micro muito possante, que pode ser usado para compilar coisas e rodar aplicativos pesados.

8 Documentação

Aqui descrevemos a documentação da rede, distinguindo entre os manuais destinados aos usuários e à documentação destinada aos administradores.

8.1 Para os usuários

A documentação para o usuário inclui os manuais já incluídos na Debian e os escritos pela administração especificamente para esta rede.

8.1.1 Documentação da Debian

A Debian, assim como outras grandes distribuições do Linux, prima por sua vasta documentação. Esta inclui as fontes tradicionais de documentacao:

- **Info**. O formato texinfo, introduzido pelo projeto GNU, pode tanto convertido para TeX e daí impresso, como ser convertido para o formato info e então ser consultado através do programa info, ou pelo emacs através do comando M-x info. O formato info, ao contrário das man pages, é estruturado e fornece referência cruzada através de links, no estilo HTML.

  O GNU Coding Standards recomenda que todos os programas do projeto GNU incluam documentação em formato info. Assim, grande parte dos utilitários que compõem a base da Debian, incluindo fileutils, textutils, shellutils e binutils tem documentação info. Como as manpages são opcionais para o projeto GNU, e é trabalhoso manter duas fontes de documentação, muitas vezes as manpages nao sao atualizadas. Portanto, sempre que existirem, os manuais info devem ser recomendados ao usuario, em detrimento das manpages.

- **Man pages**: são a fonte mais tradicional de documentação em sistemas Unix. São armazenadas em formato fonte nroff em `/usr/share/man[1-8]`, e visualizadas através do comando man. Os diretórios man[1-8] correspondem às seções (de acordo com a própria manpage man(1)):

  - **1** Programas executáveis ou comandos do shell
  - **2** Chamadas de sistema (funções fornecidas pelo kernel)
  - **3** Chamadas de biblioteca
  - **4** Arquivos especiais (normalmente encontrados em `/dev`)
  - **5** Formatos de arquivo e convenções
  - **6** Jogos
  - **7** Macro-pacotes e convenções, como man(7) e groff(7).
  - **8** Comandos de administração do sistema (somente para root, normalmente)

  Embora o projeto GNU, como mencionado acima, prefira documentação info, a política da Debian determina que todo pacote deve ter uma manpage. Quando não houver documentação info, deve-se recomendar aos usuários que consultem a manpage apropriada.

  Há um importante comando relacionado ao man: apropos. Cada manpage tem uma curta descrição associada. O comando apropos recebe uma expressão regular e devolve todas as descrições de manpages que casem com essa expressão. Esse comando deve ser recomendado a usuários que requisitem informações sobre um tópico.

- **Documentação Adicional**. A política da Debian determina que todo pacote deve instalar sua licença de distribuição e copyright no arquivo `/usr/share/doc/PACOTE/copyright` (alguns pacotes ainda usam `/usr/doc/PACOTE`, mas isso não está em conformidade com o FHS), onde PACOTE é o nome do pacote.

  No mesmo diretório, normalmente há documentação adicional no arquivo `README` e outros arquivos. Arquivos de exemplo ficam no subdiretório `examples/`.

Além dessas fontes, há também os **HOWTO**s do *Linux Documentation Project* (LDP). Entretanto, por serem mais úteis aos administradores que aos usuários, são descritos abaixo em 8.2.

8.1.2 Documentação escrita pelos admins

Apesar da documentação fornecida pela Debian ser bem abrangente, não é suficiente para a nossa rede, pois a maioria usuários não tem experiência prévia com Unix/Linux. Além disso, há vários aspectos específicos da rede que não são cobertos pela documentação tradicional. Assim, fez-se necessário a criação de FAQs.

Os FAQs da rede, todos em formato HTML, foram sendo criados de acordo com a demanda: perguntas que eram feitas frequentemente aos admins foram sendo incluídas, de acordo com o assunto, em documentos HTML. Com o tempo, alguns FAQs foram sendo introduzidos sempre que havia alguma grande mudança na rede, mesmo antes de haver a demanda pela documentação. Por exemplo, com a aquisição da impressora laser e do scanner, os FAQs foram feitos concorrentemente às configurações. Muitos desses FAQs sao na verdade "mini-HOWTOs", i.e., são escritos num formato de guia passo-a-passo, e não "perguntas e respostas".

Essa é a política atual: toda mudança nalgum serviço, ou acréscimo de serviço, deve vir acompanhada de um FAQ que explique, passo-a-passo, o que o usuário deve fazer para tomar proveito da mudança, e como resolver potenciais problemas decorrentes dela.

8.2 Para Admins

A documentação para os admins inclui a documentação incluída na Debian já descrita acima. Além disso, abrange documentos da Debian mais direcionados ao administrador, e a documentação interna, escrita pelos admins, para os admins.

8.2.1 Da Debian

Além das manpages e manuais info, descritos em 8.1.1, a Debian fornece os **FAQ**s do Linux e da Debian, e **HOWTO**s do *Linux Documentation Project* (LDP).

HOWTOs **HOWTO**s são guias detalhados para diversos tópicos em administração de sistemas, de redes, programação, configuração de periféricos, etc. O foco normalmente é configuração, não utilização.

Alguns HOWTOs de leitura "obrigatória" para o admin são (omitimos o sufixo -HOWTO):

- DOS-Win-to-Linux
- Filesystems
- Hardware
- Installation
- Kernel
- Keyboard-and-Console
- Mail-Administrator
- Mail-User
- NFS
- NIS
- Net

- Portuguese
- Printing
- Printing-Usage
- Security
- XFree86

Há também os mini-HOWTOs, documentos menores e normalmente não tão estruturados quanto os HOWTOs. Alguns mini-HOWTOs muito importantes para o admin:

- IP-Masquerade
- LILO
- Large-Disk
- Partition
- Quota

8.2.2 Documentação interna

Dado o caráter semi-voluntário da administração, e da satisfação pessoal relativamente baixa que admins costumam ter em escrever documentação, a documentação interna (escrita pelos admins, para os admins) não é tão rica e sistemática quanto poderia ser (pelo menos até a escrita deste documento).

Entretanto, já é parte da política interna da rede escrever um pequeno LEIAME para cada script e diretório especial criado, com explicações sucintas sobre o funcionamento do script (ou função do diretório), e a indicação do admin responsável.

Também é essencial comentar apropriadamente o código de toda ferramenta criada.

Além disso, muitos dos programas criados pelos admins, como o sistema de abertura de contas e o sistema de impressão, estão descritos neste documento (veja seção 5.7).

Sumário