Linux:
Administração de Sistemas

Lars Wirzenius
Joanna Oja
Stephen Stafford
Gleydson Mazioli da Silva
Federico Lupi
Fernando Miguel de Alava Soto

Tradução Para o Português

Paulo Aukar
Fernando Miguel de Alava Soto
Alexandre Folle de Menezes

Versão 1.1, julho/2002.

As marcas registradas utilizadas no decorrer deste livro são usadas unicamente para fins didáticos, sendo estas propriedade de suas respectivas companhias.

O capítulo “1 O Processo de Inicialização” é adaptação do texto:

Guia do Administrador de Sistemas Linux / Lars Wirzenius; tradução de Conectiva Informática. São Paulo: Conectiva, 1998. Título original: Linux System Administrator's Guide

Fonte: http://www.conectiva.com.br/doc/livros e
http://www.linuxdoc.org/LDP/sag/index.html

Os capítulos 2 e 6 são adaptações do texto:

Fonte: http://focalinux.cipsga.org.br

O capítulo "9 Scripts de shell" foi escrito por:

O capítulo "12 O X Window" foi escrito por:

Fonte: http://www.mclink.it/personal/MG2508/

Esta apostila é uma coletânea de textos com licença GNU ou livres encontrados na Internet, conforme referências acima. **Este material foi totalmente montado com fins didáticos, sem objetivo comercial. Foram acrescentados exemplos e exercícios desenvolvidos pela Alfamídia Ltda.**

CONTEÚDO

CONTEÚDO 3

1 O PROCESSO DE INICIALIZAÇÃO 6

1.1 UMA VISÃO GERAL DA INICIALIZAÇÃO E ENCERRAMENTO DO SISTEMA 6
1.2 O PROCESSO DE INICIALIZAÇÃO EM MAIORES DETALHES 6
1.3 MAIS INFORMAÇÕES SOBRE O ENCERRAMENTO DO SISTEMA 9
1.4 REINICIANDO O SISTEMA 11
1.5 MODO MONOUSUÁRIO 12
1.6 DISQUETES DE EMERGÊNCIA 12

2 GERENCIADORES DE INICIALIZAÇÃO (BOOT LOADERS) 13

2.1 LILO 13
2.1.1 CRIANDO O ARQUIVO DE CONFIGURAÇÃO DO LILO 13
2.1.2 OPÇÕES USADAS NO LILO 16
2.1.3 UM EXEMPLO DO ARQUIVO DE CONFIGURAÇÃO LILO.CONF 18

3 INIT 20

3.1 O INIT VEM EM PRIMEIRO LUGAR 20
3.2 CONFIGURANDO O INIT PARA INICIALIZAR O GETTY: O ARQUIVO /ETC/INITTAB 21
3.2.1 NÍVEIS DE EXECUÇÃO 23
3.3 CONFIGURAÇÕES ESPECIAIS NO /ETC/INITTAB 24
3.4 INICIANDO EM MODO MONOUSUÁRIO 25

4 KERNEL E MÓDULOS 26

4.1 O KERNEL 32
4.2 MÓDULOS 32
4.3 ADICIONANDO SUPORTE A HARDWARE E OUTROS DISPOSITIVOS NO KERNEL 33
4.4 KMOD 34
4.5 LSMOD 34
4.6 INSMOD 34
4.7 RMMOD 35
4.8 MODPROBE 35
4.9 DEPMOD 36
4.10 MODCONF 36
4.11 RECOMPILANDO O KERNEL 37
4.12 ARQUIVOS RELACIONADOS COM O KERNEL E MÓDULOS 41
4.12.1 /ETC/MODULES 41
4.12.2 MODULES.CONF 42
4.13 APLICANDO PATCHES NO KERNEL 42

5 HARDWARE 44

5.1 PLACA DE EXPANSÃO 44
5.2 NOMES DE DISPOSITIVOS 44

5.3 CONFIGURAÇÃO DE HARDWARE 45
5.3.1 IRQ - REQUISIÇÃO DE INTERRUPÇÃO 45
5.3.1.1 Prioridade das Interrupções 47
5.3.2 DMA - ACESSO DIRETO A MEMÓRIA 47
5.3.2.1 Conflitos de DMA 49
5.3.3 I/O - PORTA DE ENTRADA/SAÍDA 49
5.4 HARDWARES CONFIGURÁVEIS POR JUMPERS, DIP-SWITCHES, JUMPERLESS E PLUG-AND-PLAY 49
5.4.1 JUMPERS 49
5.4.2 DIP-SWITCHES 51
5.4.3 JUMPERLESS (SEM JUMPER) 51
5.4.4 PLUG-AND-PLAY 51
5.4.4.1 Entendendo o arquivo de configuração isapnp.conf 52
5.5 CONFLITOS DE HARDWARE 56
5.6 BARRAMENTO 56
5.7 PLACAS ON-BOARD / OFF-BOARD 58
5.8 HARDWARES ESPECÍFICOS OU "FOR WINDOWS" 60
5.9 DISPOSITIVOS ESPECÍFICOS PARA GNU/LINUX 61

6 IMPRESSÃO 62

6.1 PORTAS DE IMPRESSORA 62
6.2 IMPRIMINDO DIRETAMENTE PARA A PORTA DE IMPRESSORA 62
6.3 IMPRIMINDO VIA SPOOL 63
6.4 IMPRESSÃO EM MODO GRÁFICO 64
6.4.1 GHOST SCRIPT 64
6.5 MAGIC FILTER 67
6.5.1 INSTALAÇÃO E CONFIGURAÇÃO DO MAGIC FILTER 67
6.5.2 OUTROS DETALHES TÉCNICOS SOBRE O MAGIC FILTER 68

7 LIMITANDO O USO DE ESPAÇO EM DISCO (QUOTAS) 69

7.1 INSTALANDO O SISTEMA DE QUOTAS 69
7.2 EDITANDO QUOTAS DE USUÁRIOS/GRUPOS 71
7.3 VERIFICANDO A QUOTA DISPONÍVEL AO USUÁRIO 75
7.4 VERIFICANDO A QUOTA DE TODOS OS USUÁRIOS/GRUPOS DO SISTEMA 76

8 MANUTENÇÃO DO SISTEMA 78

8.1 CHECAGEM DOS SISTEMAS DE ARQUIVOS 85
8.1.1 FSCK.EXT2 85
8.1.2 BADBLOCKS 87
8.2 LIMPANDO ARQUIVOS DE LOGS 88
8.3 TAREFAS AUTOMÁTICAS DE MANUTENÇÃO DO SISTEMA 88
8.4 CRON 89
8.4.1 O FORMATO DE UM ARQUIVO CRONTAB 89

9 GERENCIAMENTO DE CONTAS E CUIDADOS PARA A PROTEÇÃO DE SENHAS 92

9.1 INTRODUÇÃO 92

9.2 CRIAÇÃO, MONITORAÇÃO E SEGURANÇA DE CONTAS 92
9.2.1 DEFININDO VALORES PADRÃO PARA RESTRIÇÃO 95
9.2.2 SENHAS FÁCEIS DE ADIVINHAR E ESCOLHA DE BOAS SENHAS 95
9.2.2.1 Senhas Ruins 96
9.2.2.2 Senhas Boas 96
9.2.3 ATUALIZAÇÃO DE SENHAS DE MÚLTIPLAS CONTAS 97
9.2.4 A SENHA DO USUÁRIO ROOT 97
9.3 TIPOS DE ATAQUES MAIS COMUNS PARA SE CONSEGUIR UMA SENHA 98
9.3.1 DEDUÇÃO 98
9.3.2 ENGENHARIA SOCIAL 98
9.3.3 ATAQUES POR DICIONÁRIO 100
9.3.4 FORÇA BRUTA 100
9.3.5 MONITORAÇÃO DE TOQUES DO TECLADO 100
9.3.6 LOGIN FALSO 100
9.4 MELHORANDO A SEGURANÇA DAS SENHAS ARMAZENADAS EM SEU SISTEMA ... 101
9.4.1 SHADOW PASSWORDS 101
9.4.2 SENHAS MD5 102

10 O X WINDOW 103

10.1 CONFIGURAÇÃO E INICIALIZAÇÃO 104
10.2 PERSONALIZANDO O X 109
10.3 GNOME: UM GERENCIADOR DE AMBIENTE 110
10.4 LOGIN GRÁFICO COM UM DISPLAY MANAGER 111
10.4.1 O GDM 112

1 O PROCESSO DE INICIALIZAÇÃO

1.1 Uma visão geral da inicialização e encerramento do sistema

O ato de ligar o computador e carregar o sistema operacional é chamado *booting* (inicialização). O nome vem da imagem do computador levantando-se, mas o ato em si mesmo é um pouco mais realista.

Durante a inicialização, o computador carrega inicialmente uma pequena porção de código chamado de *bootstrap loader*, o qual carrega e inicia o sistema operacional. O bootstrap loader é normalmente armazenado em uma localização fixa no disco rígido ou disquete. A razão para estes dois passos é que o sistema operacional é grande e complicado, mas a primeira parte de código a ser carregada deve ser muito pequena (poucas centenas de bytes), evitando-se a necessidade de fabricação de *firmwares* muito complicados.

Diferentes computadores implementam inicializações diferentes. Para PCs, o computador (a BIOS) lê o primeiro setor (chamado setor de inicialização) do disquete ou do disco rígido. O bootstrap loader está neste setor. Ele carrega o sistema operacional, onde quer que ele esteja no disco (ou em qualquer outro local onde ele esteja).

Após a carga do Linux, este inicializa o hardware e os controladores de dispositivos. É executado o processo init, o qual inicia os demais processos que permitem o acesso aos usuários e a execução de seus programas. Os detalhes desta fase serão discutidos a seguir.

No encerramento do sistema, todos os processos recebem o aviso para terminarem suas atividades (isso faz com que os arquivos sejam fechados e todos os procedimentos necessários sejam tomados pelos programas); os sistemas de arquivos e áreas de swap são desmontados e finalmente uma mensagem é apresentada na console, sinalizando que o computador pode ser desligado. Caso o procedimento correto não seja seguido, coisas muito desagradáveis podem e *vão* ocorrer; mais importante, o buffer cache do sistema de arquivos pode não ter sido descarregado, o que significa que todos os dados nele serão perdidos e o sistema de arquivos estará inconsistente e possivelmente instável.

1.2 O processo de inicialização em maiores detalhes

Pode-se inicializar o Linux a partir de um disquete ou do disco rígido. A seção de instalação do Guia de Instalação e Introdução ao Linux (getting-started) descreve como instalar o Linux e iniciá-lo da forma que se queira.

Quando um PC é inicializado, a BIOS executará diversos testes para checar se tudo está em perfeita ordem, chamando então o boot. Ela escolherá um dispositivo de disco (tipicamente o primeiro acionador de disquetes ou o primeiro disco rígido, se houver um instalado; a ordem, porém, pode ser configurada) e lerá o primeiro setor do dispositivo. Este é chamado de **setor de inicialização**, e no caso de discos rígidos é também chamado de **master boot record (MBR)**, uma vez que um disco rígido pode conter diversas partições, cada qual com o seus próprios setores de inicialização.

O setor de inicialização contém um pequeno programa, (pequeno o suficiente para estar contido em um setor), o qual tem a função de ler o sistema operacional instalado e inicializá-lo. Quando a inicialização é realizada a partir de um disquete, o setor de inicialização contém um determinado código que apenas lê algumas centenas de blocos (dependendo do tamanho do kernel) em uma determinada área da memória. Em um disquete de boot do Linux, não há sistema de arquivos, o kernel está armazenado em setores consecutivos, uma vez que isso simplifica o processo de inicialização. É possível inicializar o sistema a partir de um disco que contenha um sistema de arquivos utilizando-se o LILO.

Ao ser inicializado a partir de um disco rígido, o código do MBR examinará a **tabela de partições** (também localizada no MBR), identificará a **partição ativa** (a partição indicada como inicializável), lerá o **setor de inicialização** daquela partição e iniciará o programa gravado naquele setor. O código no setor de inicialização faz exatamente o que o setor de inicialização de um disquete faz: lerá o kernel de uma partição e o executará. Os detalhes variam, uma vez que não é muito útil ter uma partição separada somente para manter uma imagem do kernel, uma vez que o código no setor de inicialização não pode simplesmente ler o disco seqüencialmente, e sim tem que encontrar os setores onde quer que o sistema de arquivos o tenha colocado. Há diversas soluções para o problema, mas o mais comum é utilizar o LILO (os detalhes sobre como fazê-lo são irrelevantes neste momento, para maiores informações verifique a documentação do LILO).

Quando o sistema é iniciado utilizando o LILO, ele inicializa o padrão. É possível configurar o LILO para iniciar diversos kernels, ou mesmo iniciar outros sistemas diferentes do Linux, sendo possível usá-lo para escolher qual sistema ou kernel utilizar na hora da inicialização do sistema. Para tanto basta pressionar **alt**, **shift** ou **ctrl** durante a inicialização do sistema (quando o LILO é carregado) e serão apresentadas as opções disponíveis, ao invés da inicialização automática com o sistema definido como padrão. Alternativamente, o LILO pode ser configurado para sempre perguntar qual sistema deve ser carregado, e após um tempo máximo configurável de espera, carregar o sistema definido como padrão.

Com o LILO é possível ainda enviar **argumentos ao kernel**, imediatamente após o nome do sistema operacional.

Note que há outros gerenciadores de inicialização, além do LILO, como o GRUB e o loadlin. Informações sobre estes sistemas podem ser adicionadas em futuras versões deste manual.

Inicializar o sistema a partir de disquete ou um disco rígido tem suas vantagens e desvantagens, porém a segunda alternativa tende a ser a melhor, uma vez que evita os problemas inerentes aos disquetes, além de ser mais rápido. Ainda assim pode haver um certo nível de problemas ao se instalar o sistema a partir de um disco rígido, fazendo com que muitos usuários utilizem disquetes para uma primeira inicialização do sistema, executem a sua instalação no disco rígido, assim como a do LILO, e somente a partir daí passem a utilizar o disco rígido como unidade de inicialização do sistema.

Após a carga do kernel do Linux em memória, independente da forma, e o sistema estando efetivamente em execução, ocorrem os seguintes eventos:

- O kernel é instalado inicialmente em um formato compactado, sendo então descompactado de forma automática, através de um pequeno programa que está localizado no início da imagem do kernel.

- Caso se esteja utilizando uma placa de vídeo Super VGA que o Linux reconheça e que tenha modos de texto especiais (como utilizar 100 colunas por 40 linhas), o Linux pode perguntar que modo se deseja utilizar. Durante a compilação do kernel, é possível predefinir um modo de vídeo então essa pergunta nunca será apresentada. Isso pode ser feito também através do LILO ou do comando rdev.

- Após, o kernel verifica os demais hardwares (discos rígidos, disquetes, placas de rede...) e configura os seus controladores de dispositivos. Neste meio tempo, diversas mensagens são apresentadas, como por exemplo:

```
LILO boot:
Loading linux...
Memory: sized by int13 088h
Console: 16 point font, 400 scans
Console: colour VGA+ 80x25, 1 virtual console (max 63)
pcibios_init : BIOS32 Service Directory structure at 0x000f8630
pcibios_init : BIOS32 Service Directory entry at 0xf8080
pcibios_init : PCI BIOS revision 2.10 entry at 0xf80b0
Probing PCI hardware.
Calibrating delay loop..  ok  - 66.56 BogoMIPS
Memory: 63136k/65536k available (740k kernel code, 384k reserved,
1276k data)
Swansea University Computer Society NET3.035 for Linux  2.0
NET3: Unix domain sockets 0.13 for Linux NET3.035.
Swansea University Computer Society TCP/IP for NET3.034
IP Protocols: IGMP, ICMP, UDP, TCP
VFS: Diskquotas version dquot_5.6.0 initialized
Checking 386/387 coupling...  Ok, fpu using exception 16 erro r
reporting.
Checking 'hlt' instruction...  Ok.
Intel Pentium with F0 0F bug  - workaround enabled.
alias mapping IDT readonly ...  ...  done
Linux version 2.0.36 (root@frajola.conectiva.com.br) (gcc version
2.7.2.3) #2 se
g set 7 11:54:18 EST 1998
```

- O texto exato difere em diversos sistemas, dependendo do hardware disponível, assim como da versão do Linux e de como ele foi configurado.

- O kernel tentará então montar o **sistema de arquivos raiz**. A partição pode ser configurada em tempo de compilação, ou a qualquer tempo através do LILO ou do rdev. O tipo de sistema de arquivos é detectado automaticamente. Caso a sua montagem falhe, devido ao esquecimento de inclusão deste tipo de sistema de arquivos no kernel, este dará uma mensagem de erro fatal (panic) e abortará a inicialização do sistema.

- O sistema de arquivos raiz é normalmente montado com permissões somente de leitura (isto pode ser configurado também com o rdev). É possível, desta maneira, checar o sistema de arquivos enquanto ele está montado, porém não é uma boa idéia tentar fazer isso em sistemas de arquivos montados com permissão de escrita/leitura.

- Após isso, o kernel executará o programa **init** (localizado em /sbin/init) em segundo plano. O init sempre terá o número de processo igual a 1. O init cuida da ativação de diversos serviços. O que ele fará exatamente depende de como está configurado; veja o capítulo sobre o init para maiores informações. No mínimo ele iniciará alguns servidores essenciais.

- O init então passará o sistema para o modo multi-usuário e iniciará o getty, permitindo o uso de consoles virtuais e linhas seriais. O getty é o programa que permite que os usuários acessem o sistema. O init pode também inicializar outros programas, dependendo de como esteja configurado.

Após estes passos o sistema estará ativo e pronto para uso.

1.3 Mais informações sobre o encerramento do sistema

É importante seguir corretamente os passos quando encerrar o sistema. Caso este procedimento não seja cumprido, seus sistemas de arquivos provavelmente estarão prejudicados e os arquivos poderão ficar bagunçados. Isto acontece porque o Linux tem um cache de disco que não grava os dados no disco no momento em que isto é solicitado, mas somente em intervalos de tempo. A performance do sistema aumenta consideravelmente, mas isso também pode causar a perda de dados caso você simplesmente desligue seu computador.

Outra razão contra desligar simplesmente o sistema é que em um sistema multitarefa existem diversas coisas acontecendo em segundo plano, e desligar o computador neste momento pode ser desastroso. Usando o

procedimento correto para o encerramento do sistema, você garante que todos os processos em background podem salvar seus dados.

Para encerrar corretamente um sistema Linux, utilize o comando shutdown. Ele é usado normalmente de duas maneiras.

Caso esteja utilizando um sistema onde você é o único usuário, a maneira correta de se utilizar o shutdown é sair de todos os programas que porventura você esteja rodando, sair de todas as consoles do sistema, acessar uma das consoles como root (ou permanecer no sistema como root mudando o diretório corrente para o diretório pessoal do root para não ter problemas na desmontagem de sistemas de arquivos); executar o comando "shutdown -h now". Isso fará com que o sistema inicie o shutdown imediatamente.

Exercício:
Execute:
```
# shutdown -h now
```

O que aconteceu?

Alternativamente, se o seu sistema tem diversos usuários, use o comando "`shutdown -h +`*`tempo mensagem`*", onde *tempo* é o tempo em minutos até o sistema ser parado, e *mensagem* é a mensagem que será enviada aos usuários que estão conectados neste momento ao sistema.

Exercício:
Execute:
```
# shutdown -h +2 'Iremos instalar um novo disco.  O sistema deverá
voltar ao normal em três horas.'
```

Este procedimento avisará a todos os usuários que o sistema será desligado em dez minutos, e que eles poderão perder seus dados. O aviso é apresentado em todo o terminal que estiver conectado, inclusive xterms:

```
Broadcast message from root (ttyp0) Wed Aug 2 01:03:25 1995...

Iremos instalar um novo disco.  O sistema deverá voltar ao normal em
três horas.
The system is going DOWN for system halt in 10 minutes !!
```

Este aviso é automaticamente repetido algumas vezes antes do encerramento, e a intervalos cada vez menores, à medida que o tempo passa.

Quando o encerramento do sistema tem início, todos os sistemas de arquivos (exceto o raiz) são desmontados, os processos de usuários (casa haja algum ainda utilizando o sistema) são finalizados, servidores são encerrados, e finalmente o sistema de arquivos raiz é desmontado. Quando isso ocorre o processo init apresenta uma mensagem indicando que o computador pode ser desligado. Agora, *e jamais antes disso*, pode-se desligar o equipamento.

Algumas vezes, apesar de ser raro em bons sistemas, é impossível encerrar o sistema de forma adequada. Por exemplo, caso ocorra um erro fatal com o kernel, pode ser impossível executar qualquer novo comando, tornando

o encerramento correto inviável. Tudo o que se pode fazer neste caso é esperar que nenhum dano maior ocorra e então desligar a máquina. Caso os problemas sejam menos sérios (digamos que alguém quebrou o seu teclado com um machado!), e o kernel está rodando normalmente, é aconselhável aguardar alguns minutos para que os dados sejam atualizados, esvaziando assim o cache de disco, e somente após desligar o equipamento.

Alguns usuários gostam de utilizar o comando sync três vezes, aguardando pela gravação do buffer em disco, e desligando o equipamento em seguida. Caso não hajam programas sendo executados, esse procedimento equivale ao shutdown. Ainda assim nenhum sistema de arquivos é desmontado o que pode gerar problemas com o indicador de limpeza do sistema de arquivos do tipo ext2fs. Este método não é recomendado.

(A razão dos três comandos sync remonta aos primeiros tempos do UNIX, onde os comandos eram digitados separadamente, o que dava tempo suficiente para que as operações de E/S fossem finalizadas).

1.4 Reiniciando o sistema

Reinicializar significa iniciar o sistema novamente. Isto pode ser conseguido encerrando o sistema, desligando o computador e então ligando-o novamente. Um método mais prático para ter o mesmo efeito consiste em executar o shutdown com a opção de reinicialização. Isto é conseguido executando-se `shutdown -r now`, sendo que a opção -r significa "reinicialização".

Exercício:
Execute:
```
# shutdown -r now
```

Note que o sistema reinicializa automaticamente.

A maioria dos sistemas Linux executa `shutdown -t3 -r now` quando o conjunto de teclas ctrl-alt-del são pressionadas. Este procedimento reinicializa o sistema. A opção "-t3" faz com que o a reinicialização ocorre 3 segundos após pressionar as teclas A ação do ctrl-alt-del é configurável e poderá ser desejável aguardar alguns minutos para a sua execução em uma máquina multi-usuário.

Exercício:
Pressione:
```
<Ctrl>+<Alt>+<Del>
```

Sistemas que são fisicamente acessíveis a qualquer pessoa, devem ser configurados para não executar nada ao serem pressionadas tais teclas. Isto pode ser feito editando-se o arquivo /etc/inittab e comentando-se a linha respectiva.

1.5 Modo monousuário

O comando shutdown também pode ser usado para colocar o sistema em modo monousuário, no qual ninguém pode acessar o sistema, além do super-usuário root. Isso é muito útil para tarefas de administração que não podem ser executadas caso o sistema esteja sendo executado normalmente.

1.6 Disquetes de emergência

Nem sempre pode ser possível inicializar o sistema a partir do disco rígido. Um erro na configuração do LILO pode tornar o sistema incapaz de ser iniciado. Para essas situações, é necessário ter uma alternativa de inicialização que funcione sempre (desde que o hardware também funcione). Para PCs, isso pode significar inicializar o sistema a partir do acionador de disquetes.

Muitas distribuições permitem a criação de **disquetes de emergência** durante a instalação. É uma excelente idéia criá-los, ainda que algumas vezes ele possa conter somente o kernel do sistema, e presuma que os programas serão utilizados a partir dos discos de instalação da distribuição, para corrigir o que quer que seja. Algumas vezes isso pode não ser suficiente, como por exemplo, quando se deseja restaurar alguns arquivos a partir de cópias de segurança feitos através de softwares não presentes nos discos de instalação.

Pode ser necessário criar um disquete de inicialização personalizado. O *Bootdisk HOWTO* de Graham Chapman (bootdisk-howto) contém as instruções de como se fazer isso. Mas lembre-se de manter os disquetes de emergência e de boot atualizados.

Não se pode utilizar a unidade de disquetes utilizada para montar o disquete de boot para qualquer outra finalidade. Isso pode ser inconveniente caso se tenha somente uma unidade de disquetes. Ainda assim, caso haja memória suficiente, pode-se configurar o disquete de boot para carregar o seu conteúdo para a memória (o disco de inicialização do kernel necessita ser especialmente configurado para isso). Uma vez que o disquete tenha sido carregado em memória, a unidade estará liberada para montar outros discos.

2 GERENCIADORES DE INICIALIZAÇÃO (BOOT LOADERS)

Gerenciadores de Inicialização são programas que carregam um sistema operacional e/ou permitem escolher qual será iniciado. Normalmente este programas são gravados no *setor de boot* (inicialização) da partição ativa ou no *master boot record* (MBR) do disco rígido.

Este capitulo explica o funcionamento de cada um dos principais gerenciadores de partida usados no GNU/Linux, em que situações é recomendado seu uso, as características, como configurá-lo e alguns exemplos de configuração.

2.1 LILO

O LILO (*Linux Loader*) é sem dúvida o gerenciador de partida padrão para quem deseja iniciar o GNU/Linux através do disco rígido. Ele permite selecionar qual sistema operacional será iniciado (caso você possua mais de um) e funciona tanto em discos rígidos *IDE* como *SCSI*.

A seleção de qual sistema operacional e a passagem de parâmetros ao kernel pode ser feita automaticamente ou usando o aviso de boot: do LILO.

2.1.1 Criando o arquivo de configuração do LILO

Os dados para a criação do novo *setor de boot* que armazenará o gerenciador de partida são lidos do arquivo **`/etc/lilo.conf`** Este arquivo pode ser criado em qualquer editor de textos (como o vi). Normalmente ele é criado durante a instalação de sua distribuição GNU/Linux mas por algum motivo pode ser preciso modificá-lo ou personalizá-lo (para incluir novos sistemas operacionais, mensagens, alterar o tempo de espera para a partida automática, etc).

O arquivo `/etc/lilo.conf` é dividido em duas seções: *Geral* e *Imagens*. A seção *Geral* vem no início do arquivo e contém opções que serão usadas na inicialização do Lilo e parâmetros que serão passados ao kernel. A seção *Imagens* contém opções especificas identificando qual a partição que contém o sistema operacional, como será montado inicialmente o sistema de arquivos, tabela de partição, o arquivo que será carregado na memória para inicializar o sistema, etc. Abaixo um modelo do arquivo `/etc/lilo.conf` para sistemas que só possuem o GNU/Linux instalado:

```
      boot=/dev/hda1
compact
install=/boot/boot.b
map=/boot/map
vga=normal
delay=20
```

```
lba32
image=/vmlinuz
    root=/dev/hda1
    label=Linux
    read-only
```

Para criar um novo gerenciador de partida através do arquivo `/etc/lilo.conf` , execute o comando `lilo`.

No exemplo acima, o gerenciador de partida será instalado em /dev/hda1, utilizará um setor de boot compacto (compact), modo de vídeo VGA normal (80x25), esperará 2 segundos antes de processar automaticamente a primeira seção image= e carregará o kernel /vmlinux de /dev/hda1. Para detalhes sobre opções que podem ser usadas neste arquivo veja a página de manual para `lilo.conf`.

Para mostrar o aviso de boot:, você deverá ligar as teclas Caps Lock ou Scrool lock na partida ou pressionar a tecla Shift durante os dois segundos de pausa. Outro método é incluir a opção prompt na seção *global* para que o aviso de boot: seja mostrado automaticamente após carregar o Lilo.

Abaixo uma configuração para computadores com mais de um sistema operacional (Usando GNU/Linux e Windows):

```
boot=/dev/hda
compact
lba32
install=/boot/boot.b
map=/boot/map
vga=normal
delay=20
prompt
image=/vmlinuz
    label=Linux
    root=/dev/hda3
    read-only
other=/dev/hda1
    label=Windows
    table=/dev/hda
```

O exemplo acima é parecido ao anterior, o que foi acrescentado foi a opção prompt na seção *geral* (para que seja mostrado imediatamente o aviso de boot: no momento em que o LILO for carregado), e incluída uma imagem de disco Windows localizado em /dev/hda1. Também foi indicado que o LILO seja instalado no MBR do disco /dev/hda. Note a primeira linha e veja que não diz /dev/hda1. No momento da inicialização é mostrada a mensagem boot: e caso seja digitado Windows e pressionado ENTER, o sistema iniciará o Windows. Caso a tecla Enter seja pressionada sem especificar a imagem, a primeira será carregada (neste caso o GNU/Linux).

Você pode substituir a palavra Linux da opção label por o número 1 e Windows por 2, desta forma o número pode ser digitado para iniciar o sistema operacional. Isto é muito útil para construir um menu usando a opção message.

A seção *Geral* vem do início do arquivo até a palavra delay=20. A partir do primeiro aparecimento da palavra image, other ou range, tudo o que vier abaixo será interpretado como imagens de inicialização.

Por padrão, a imagem carregada é a especificada por default= ou a primeira que aparece no arquivo (caso default= não seja especificado). Para carregar o outro sistema (o Windows), digite o nome da imagem de disco no aviso de boot: (especificada em label=) que será carregada. Você também pode passar parâmetros manualmente ao kernel digitando o nome da imagem de disco e uma opção do kernel ou através do arquivo `/etc/lilo.conf` .

O LILO pode inicializar o seguintes tipos de imagens:

- Imagens do kernel de um arquivo. Normalmente usado para iniciar o GNU/Linux pelo disco rígido e especificado pelo parâmetro image=.

- Imagens do kernel de um dispositivo de bloco (como um disquete). Neste caso o número de setores a serem lidos devem ser especificados na forma *PRIMEIRO-ÚLTIMO* ou *PRIMEIRO+NÚMERO de setores a serem lidos*.

É necessário especificar o parâmetro image= e range=, por exemplo:

```
image=/dev/fd0
   range=1+512
```

Todas as opções do kernel podem ser usadas na inicialização por dispositivo.

O setor de boot de outro sistema operacional (como o DOS, OS/2, etc). O setor de partida é armazenado junto com a tabela de partição no arquivo /boot/map. É necessário especificar o parâmetro OTHER=dispositivo ou OTHER=arquivo e a inicialização através de um setor de partida possui algumas opções especiais como o TABLE= (para especificar a tabela de partição) e o MAP-DRIVE= (identificação da unidade de discos pelo sistema operacional). Veja o exemplo desta configuração abaixo:

```
other=/dev/hda2
    table=/dev/hda
    label=DOS
    map-drive=0x80
    to = 0x81
    map-drive=0x81
    to = 0x80
```

Observações:

- Caso o gerenciador de partida seja instalado no MBR do disco rígido (`boot=/dev/hda`), o setor de boot do antigo sistema operacional será substituído, retire uma cópia do setor de boot para um disquete usando o comando `dd if=/dev/hda of=/floppy/mbr bs=512 count=1` no GNU/Linux para salvar o setor

de boot em um disquete e `dd if=/floppy/mbr of=/dev/hda bs=446 count=1` para restaurá-lo. No DOS, você pode usar o comando `fdisk /mbr` para criar um novo Master Boot Record.

- Após qualquer modificação no arquivo `/etc/lilo.conf` , o comando `lilo` deverá ser novamente executado para atualizar o setor de partida do disco rígido. Isto também é válido caso o kernel seja atualizado ou a partição que contém a imagem do kernel desfragmentada.

- A limitação de 1024 cilindros do Lilo não existe mais a partir da versão 21.4.3 (recomendada, por conter muitas correções) e superiores.

- A reinstalação, formatação de sistemas DOS e Windows pode substituir o setor de partida do HD e assim o gerenciador de partida, tornando impossível a inicialização do GNU/Linux. Antes de reinstalar o DOS ou Windows, verifique se possui um disquete de partida do GNU/Linux.

Para gerar um novo boot loader, coloque o disquete na unidade e após o aviso boot: ser mostrado, digite `linux root=/dev/hda1` (no lugar de `/dev/hda1` você coloca a partição raiz do GNU/Linux), o sistema iniciará. Dentro do GNU/Linux, digite o comando `lilo` para gerar um novo setor de partida.

Agora reinicie o computador, tudo voltará ao normal.

2.1.2 Opções usadas no LILO

Esta seção traz opções úteis usadas no arquivo `lilo.conf` com explicações sobre o que cada uma faz. As opções estão divididas em duas partes: As usadas na seção *Global* e as da seção *Imagens* do arquivo `lilo.conf`.

Global

- `backup=[arquivo/dispositivo]` - Copia o setor de partida original para o arquivo ou dispositivo especificado.

- `boot=dispositivo` - Define o nome do dispositivo onde será gravado o setor de partida do LILO (normalmente é usada a partição ativa ou o Master Boot Record - MBR). Caso não seja especificado, o dispositivo montado como a partição raiz será usado.

- `compact` - Tenta agrupar requisições de leitura para setores seguintes ao sendo lido. Isto reduz o tempo de inicialização e deixa o mapa menor. É normalmente recomendado em disquetes.

- `default=imagem` - Usa a imagem especificada como padrão ao invés da primeira encontrada no arquivo `lilo.conf`.

- `delay=[num]` - Permite ajustar o número de segundos (em décimos de segundos) que o gerenciador de partida deve aguardar para carregar a primeira imagem de disco (ou a especificada por default=). Esta pausa lhe permite selecionar que sistema operacional será carregado.

- `install=setor-boot` - Instala o arquivo setor-boot como novo setor de boot do disco. Se install for omitido, /boot/boot.b é usado por padrão.

- `lba32` - Permite que o LILO quebre o limite de 1024 cilindros do disco rígido, inicializando o GNU/Linux em um cilindro acima deste através do acesso . Note que isto requer compatibilidade com o BIOS, mais especificamente que tenha suporte a chamadas int 0x13 e AH=0x42. É recomendado o seu uso.

- `map=arquivo-mapa` - Especifica a localização do arquivo de mapa (.map). Se não for especificado, /boot/map é usado.

- `message=arquivo` - Especifica um arquivo que contém uma mensagem que será mostrada antes do aviso de boot:. Nenhuma mensagem é mostrada até que seja pressionada a tecla Shift após mostrar a palavra LILO. O tamanho da mensagem deve ser no máximo 65535 bytes. O arquivo de mapa deve ser novamente criado caso a mensagem seja retirada ou modificada. Na mensagem, o caractere FF (CTRL+L) limpa a tela.

- `nowarn` - Não mostra mensagens de alerta.

- `password=senha` - Permite proteger todas as imagens de disco com uma única senha. Caso a senha esteja incorreta, o LILO é novamente carregado.

- `prompt` - Mostra imediatamente o aviso de boot: ao invés de mostrar somente quando a tecla Shift é pressionada.

- `verbose=[num]` - Ativa mensagens sobre o processamento do LILO. Os números podem ser especificados de 1 a 5, quanto maior o número, maior a quantidade de detalhes mostrados.

- `timeout=[num]` - Ajusta o tempo máximo de espera (em décimos de segundos) de digitação no teclado. Se nenhuma tecla é pressionada no tempo especificado, a primeira imagem é automaticamente carregada. Igualmente a digitação de senha é interrompida se o usuário estiver inativo por este período.

Adicionalmente as opções de imagem do kernel append, ramdisk, read-only, read-write, root e vga podem ser especificadas na seção *global*. Opções por Imagem

As opções por imagem iniciam com uma das seguintes opções: `image=`, `other=` ou `range=`. Opções usadas por cada imagem:

- `table=dispositivo` - Indica o dispositivo que contém a tabela de partição para aquele dispositivo. Necessário apenas para imagens especificadas por other=.
- `unsafe` - Não acessa o setor de boot no momento da criação do mapa. Isto desativa algumas checagens, como a checagem da tabela de partição. unsafe e table= são incompatíveis.
- `label=[nome]` - Permite especificar um nome para a imagem. Este nome será usado na linha boot: para inicializar o sistema.
- `alias=[nome]` - Apelido para a imagem de disco. É como um segundo label.
- `optional` - Ignora a imagem caso não estiver disponível no momento da criação do mapa. É útil para especificar kernels que não estão sempre presentes no sistema.
- `password=senha` - Protege a imagem atual com a senha. Caso a senha esteja incorreta, o setor de partida do Lilo é novamente carregado.
- `restricted` - A senha somente é pedida para iniciar a imagem se o sistema for iniciado no modo single.

Também podem ser usados parâmetros de inicialização do kernel no arquivo `/etc/lilo.conf`.

2.1.3 Um exemplo do arquivo de configuração `lilo.conf`

Abaixo um exemplo do arquivo `/etc/lilo.conf` que poderá ser usado em instalações GNU/Linux com o DOS.

```
boot=/dev/hda               #Instala o LILO no MBR do /dev/hda
compact
install=/boot/boot.b
map=/boot/map
message=/etc/lilo.message    #mensagem que será mostrada na tela
default=1  #Carrega a Imagem especificada por label=1 como padrão
vga=normal #usa o modo de video 80x25 ao iniciar  o Linux
delay=20   #aguarda 2 segundos antes de iniciar a imagem padrão
lba32     #permite quebrar o limite de 1024 cilindros na inicialização
prompt     #mostra o aviso de "boot:" logo que o LILO é carregado
image=/vmlinuz      #especifica o arquivo que  contém a primeira imagem
    root=/dev/hda1  #partição onde a imagem acima esta localizada
```

```
    label=1         #identificação da imagem de disco
    read-only       #monta inicialmente como somente leitura
    password=12345  #Usa a senha 12345
    restricted      #somente quando iniciar com o parâmetro single
other=/dev/hda2     #especifica outro sistema que será carregado
    table=/dev/hda  #a tabela de partição dele está em /dev/hda
    label=2         #identificação desta imagem de disco
    password=12345  #pede a senha antes de iniciar este sistema
```

Você pode usar o exemplo acima como base para construir sua própria configuração personalizada do `/etc/lilo.conf` mas não se esqueça de modificar as tabelas de partições para seu sistema. Se você usa o Windows NT 4.0, Windows NT 5.0 (Windows 2000) ou o OS/2, recomendo ler o DOS+Windows+OS/2-HOWTO.

Exercícios:

1. Faça uma cópia de segurança do seu `lilo.conf` atual. Por exemplo:
   `cp /etc/lilo.conf /etc/lilo.conf.BAK`

2. Edite o `lilo.conf` com o vi e altere os campos “label”. Salve e execute o comando `lilo` para reinstalar o gerenciador. Dê boot no sistema e veja o que acontece

3. Edite novamente o `lilo.conf`, restaure os campos “label”.

4. Troque o valor do timeout para 50. Reinstale o LILO, chamando “`lilo`” e reinicie o sistema. Não faça mais nada. O que ocorreu?

5. Entre no Linux novamente. Edite o `lilo.conf`, restabeleça o timeout, e estude uma forma de inverter o sistema que será carregado por default. Edite o arquivo e antes de instalar peça para o professor verificar. Chame o `lilo` e reinicie a máquina.

6. Copie a cópia de segurança do `lilo.conf` que você guardou em /tmp novamente para o /etc. (`cp /etc/lilo.conf.BAK /etc/lilo.conf`). Execute o `lilo` novamente e reinicie a máquina testando as opções Windows e Linux.

3 INIT

Uuno on numero yksi

Este capítulo descreve o processo init, que é o primeiro processo de nível de usuário que é iniciado pelo kernel. O init tem diversas tarefas importantes, como iniciar o getty (após o qual os usuários podem acessar o sistema), implementar os níveis de execução, e tomar conta dos processos órfãos. Este capítulo explica como o init é configurado e como fazer uso dos diferentes níveis de execução.

3.1 O init vem em primeiro lugar

O init é um dos programas absolutamente essenciais para a operação de um sistema Linux, mas que a maioria dos usuários pode ignorar. Uma boa distribuição do Linux conterá a configuração de um init que funcionará com a maioria dos sistemas, e não haverá necessidade de se fazer absolutamente nada. Usualmente o administrador somente irá preocupar-se com o init caso necessite lidar com terminais seriais, modems configurados para atendimento (e não para chamadas) ou caso deseje mudar o nível de execução padrão do sistema.

Quando o kernel auto-inicializa (foi carregado em memória, começa a rodar e inicializa todos os dispositivos e estruturas de dados), ele finaliza as suas tarefas na inicialização do sistema ao iniciar o programa de nível de usuário init. O init é sempre o primeiro processo do sistema (o seu número de processo é sempre igual a 1). Você pode conferir isso chamando:

```
# ps -C init
```

A opção "`-c`" permite listar o processo init. Note que o PID do processo é sempre 1.

O kernel procura pelo init em alguns diretórios que vêm sendo usados historicamente para isso, porém a localização correta no Linux é o /sbin/init. Caso o kernel não consiga encontrá-lo, ele executará o programa /bin/sh e caso isso também falhe, a inicialização do sistema é abortada.

Quando o init começa, ele finaliza o processo de inicialização ao executar uma série de tarefas administrativas como a checagem dos sistemas de arquivos, limpeza do /tmp, início de vários serviços e inicialização do getty para cada terminal e console virtual através dos quais os usuários serão autorizados a acessar o sistema.

Após o sistema ter sido adequadamente inicializado, o init inicia o getty para cada terminal e reinicia após cada saída do usuário do sistema (permitindo que o próximo usuário possa acessar o sistema). Além disso, o init adota também todos os processos órfãos: quando um processo inicia um processo filho e é finalizado antes dele, imediatamente o processo restante

torna-se filho do init. Isso é importante por várias razões técnicas, mas é bom conhecer para entender as listas de processos e a árvore de processos.

3.2 Configurando o init para inicializar o getty: o arquivo /etc/inittab

Quando é inicializado, o init lê o arquivo de configuração /etc/inittab. Enquanto o sistema estiver no ar, ele será lido novamente, caso seja enviado um sinal HUP, tornando desnecessário reinicializar o sistema para que as mudanças do init façam efeito.

Para forçar o init a reler o arquivo de configuração, execute:

```
# kill -HUP 1
```

Esse comando envia um sinal HUP para o processo com PID 1, que é sempre o init.

O arquivo /etc/inittab é um pouco complicado. Começaremos pelo caso mais simples, ou seja, configurando as linhas do getty. As linhas do /etc/inittab consistem de quatro campo delimitados por dois pontos:

```
id:nível:ação:processo
```

Os campos são descritos a seguir. O /etc/inittab pode conter algumas linhas vazias, e linhas que comecem com `#' serão consideradas comentários.

- id: Identifica a linha no arquivo. Para linhas referentes ao getty especifica o terminal em que eles são executados (os caracteres após o /dev/tty no nome do arquivo de dispositivo). Para outras linhas não têm efeito (exceto pelas restrições de tamanho), e devem ser únicas. Deve ter no máximo 4 caracteres (algumas versões do init requerem 2 caracteres).

- nível: Os níveis de execução em que a linha deve ser considerada. Os níveis de execução são definidos através de dígitos sem delimitadores e são melhores descritos na próxima seção.

- ação: Define a ação que deve ser tomada pela linha. Por exemplo, *respawn* para executar novamente o comando do próximo campo, quando este encerra seu processamento ou *once* para executá-lo somente uma única vez. Veja o man do inittab para detalhes de cada ação.

- processo: O comando a ser executado, com todos seus argumentos.

Para iniciar o getty no primeiro terminal virtual (/dev/tty1), em todos os modos de execução multi-usuário (de 2 a 5), pode-se informar a seguinte linha:

```
1:2345:respawn:/sbin/getty 38400 tty1
```

O primeiro campo diz que a linha deve ser executada para /dev/tty1. O segundo que ele aplica-se aos níveis de execução de 2 a 5. O terceiro que o comando deve ser reinicializado quando o processo termina (ou seja quando um usuário desconectar-se de um terminal, o getty será executado novamente para que outro usuário possa conectar-se). O último campo executa o processo getty no primeiro terminal virtual.

Note que existem 6 terminais configurados no seu sistema após a instalação (tty1 a tty6) – isto se alguém ainda não acrescentou mais terminais. Você pode mudar o terminal que está usando pressionando as teclas Alt+F*n*, onde *n* é um número de 1 a 12. Você pode ter até 24 terminais no Linux, acessando os primeiros 12 através da tecla Alt da esquerda mais as teclas de função F1 a F12 e os outros 12 através da tecla Alt da direita mais as mesmas teclas de função. Experimente acrescentar um novo terminal em seu inittab. Com o vi, edit o arquivo /etc/inittab e acrecente a seguinte linha logo após a linha do getty na tty6:

```
7:23:respawn:/sbin/getty 38400 tty7
```

Em seguida, salve o arquivo e mande um sinal HUP para o init.

```
# kill -HUP 1
```

Experimente acessar o terminal 7 apertando Alt+F7 (Alt da esquerda).

Exercício:
Acrescente terminais de 1 a 15.
Remova todos os terminais, exceto os 6 primeiros, deixando o sistema da mesma forma como foi instalado.

Caso um comando falhe ao ser executado, e o init esteja configurado para reinicializá-lo (com a ação respawn, por exemplo), isso certamente consumirá uma grande quantidade de recursos pois o processo de iniciar o comando se repetirá indefinidamente. Para prevenir esse tipo de problema, o init verificará a freqüência de reinicialização do comando e caso esta seja muito grande, o init aguardará automaticamente por cinco minutos antes de iniciá-lo novamente.

Exercício
Acrecente a linha:
`eco:23:respawn:echo`
no `/etc/inittab`*. Mande o sinal HUP para o init e veja a mensagem que aparece. O comando echo somente repete a mensagem passada por argumento (neste caso nenhuma) e retorna. Como colocamos a ação respawn, o init tenta reinicializá-lo, e o echo retorna imediatamente, ficando nesta processo indefinidamente. Note a mensagem no terminal:*
`INIT: Id “eco” respawing too fast: disabled for 5 minutes.`
Remova agora a linha que acrescentou e envie um sinal HUP novamente ao init para evitar que ele fique tentanto reiniciar o echo a cada 5 minutos.

3.2.1 Níveis de execução

Nível de execução é o estado do init e de todo o sistema que define que serviços estarão operacionais. Eles são identificados por números, de acordo com a tabela. Não há nenhum consenso de como utilizar os níveis definidos para usuário (de 2 a 5). Alguns administradores de sistema utilizam os níveis de execução para definir quais subsistemas serão executados, por exemplo se o X estará disponível ou as funcionalidades de rede e assim por diante. Outros têm todos os subsistemas sendo ativados e sendo finalizados individualmente, sem mudar o nível de execução, já que este pode ser um pouco complexo para controlar seus sistemas. Cada administrador deve definir qual o método mais adequado às suas necessidades, porém seguir a forma definida pela distribuição em uso deve ser o meio mais simples.

Tabela: Níveis de execução

0	Desligar.
1	Monousuário.
2	Multi-usuário, sem NFS.
3	Multi-usuário.
4	Não usado.
5	X11.
6	Reinicialização.

Níveis de execução são configurados no /etc/inittab por linhas como a seguinte:

```
l2:2:wait:/etc/rc.d/rc 2
```

O primeiro campo é um rótulo arbitrário; o segundo significa que ele se aplica ao nível de execução 2. O terceiro (wait) significa que o init deve executar o comando contido no quarto campo uma única vez, quando o sistema entrar neste nível, e que o init deve aguardar que ele seja concluído. O /etc/rc.d/rc executa todos comandos necessários para iniciar e parar os serviços previstos para o nível 2.

O comando no quarto campo executa todo o trabalho duro de configurar um nível de execução. Ele inicia os serviços que ainda não estejam sendo

executados e finaliza os serviços que não devem rodar neste nível. Exatamente qual o comando a ser utilizado ou como o nível está configurado depende de cada distribuição do Linux.

Quando o init é iniciado, ele procura por uma linha no /etc/inittab que especifique o nível de execução padrão:

```
id:3:initdefault:
```

Pode-se informar ao init para iniciar o sistema em um outro nível de execução, passando ao kernel argumentos como single ou emergency. Isso permite escolher o modo monousuário.

Enquanto o sistema está sento executado o comando telinit pode mudar o modo de execução, o que faz com que o init execute o comando apropriado definido no /etc/inittab.

Experimente mudar para o modo monousuário executando:

```
# telinit 1
```

O modo monousuário é apropriado para manutenções onde você quer ter certeza que ninguém mais estará executando processos no sistema. Veja uma seção sobre isso mais adiante.

3.3 Configurações especiais no /etc/inittab

O arquivo /etc/inittab tem algumas funcionalidades especiais que permitem ações diferenciadas em situações especiais. Estas funcionalidades são definidas através de palavras chaves utilizadas no terceiro campo. Alguns exemplos:

- `powerwait`: Permite que o init encerre o sistema na falta de energia elétrica. Assume que o sistema está utilizando uma unidade de alimentação extra (no-break) e que o software da unidade informará sobre a falta de energia.

- `ctrlaltdel`: Permite ao init reinicializar o sistema, quando as teclas control-alt-del forem pressionadas simultaneamente. Veja que o administrador pode configurar o C-A-D para executar outra função. Isto é aplicável, por exemplo, nos casos em que o sistema esteja em uma localização pública.

- `sysinit`: comando que deve ser executado quando o sistema for inicializado. Este comando pode limpar o conteúdo do /tmp, por exemplo.

Esta lista não é completa. Veja a página de manual do *inittab*(5) para todas as possibilidades e detalhes de como utilizá-las.

3.4 Iniciando em modo monousuário

Um nível de execução extremamente importante é o modo **monousuário**, no qual somente o administrador do sistema utiliza a máquina e o menor número possível de serviços (inclusive logins) estarão disponíveis. Este modo de execução é necessário para algumas tarefas administrativas, tais como na execução do fsck na partição /usr, isto requer que a partição esteja desmontada, o que não pode ocorrer a menos que todos os serviços do sistema estejam finalizados.

Um sistema em execução pode mudar para monousuário através do comando telinit. Durante a inicialização do sistema a palavra single ou emergency, na linha de comando do kernel, faz com que o init seja informado do nível de execução a iniciar (a linha de comando do kernel pode variar de sistema para sistema. Depende de como você está inicializando seu sistema).

A inicialização em modo monousuário pode ser necessária para executar-se o comando fsck manualmente, antes de qualquer montagem ou acesso a uma partição /usr com problemas (qualquer atividade em um sistema de arquivos inconsistente pode trazer mais problemas, devendo o fsck ser executado o mais rapidamente possível).

Os scripts de inicialização do init automaticamente entrarão em modo monousuário caso o comando fsck executado de forma automática apresente algum problema durante a inicialização do sistema. Esta é uma tentativa de prevenir que o sistema utilize um sistema de arquivos danificado e que o fsck não possa corrigir automaticamente. Tais casos são relativamente raros e usualmente envolvem um disco rígido com problemas ou uma versão experimental do kernel, porém é desejável que se esteja preparado.

Como medida de segurança, um sistema adequadamente configurado pedirá a senha do root antes de iniciar um interpretador em modo monousuário. De outra forma seria fácil simplesmente informar uma linha ao LILO e ganhar acesso ao sistema como super-usuário. Caso o problema esteja no arquivo /etc/passwd, o melhor é ter-se à mão um disquete de inicialização.

4 COMANDOS DO SISTEMA

4.1 O comando `uptime`

Mostra o tempo de execução do sistema desde que o computador foi ligado.

uptime

4.2 O comando `dmesg`

Mostra as mensagens de inicialização do kernel. São mostradas as mensagens da última inicialização do sistema.

dmesg|less

4.3 O comando `df`

Mostra o espaço livre/ocupado de cada partição.
df [*opções*]
onde:
opções

-a

Inclui sistemas de arquivos com 0 blocos.

-h, --human-readable

Mostra o espaço livre/ocupado em *MB, KB, GB* ao invés de blocos.

-H

Idêntico a -h mas usa 1000 ao invés de 1024 como unidade de cálculo.

-k

Lista em Kbytes.

-l

Somente lista sistema de arquivos locais.

-m

Lista em Mbytes (equivalente a --block-size=1048576).

--sync

Executa o sync antes de mostrar os dados.

-T

Lista o tipo de sistema de arquivos de cada partição

-t *tipo*

Lista somente sistema de arquivos do tipo *tipo*.

-x *tipo*

Não lista sistemas de arquivos do tipo *tipo*.

Exemplos: df, df -h, df -t vfat.

4.4 O comando du

Mostra o espaço ocupado por arquivos e sub-diretórios do diretório atual.
du [*opções*]
onde:
opções

-a, --all

Mostra o espaço ocupado por todos os arquivos.

-b, --bytes

Mostra o espaço ocupado em bytes.

-c, --total

Faz uma totalização de todo espaço listado.

-D

Não conta links simbólicos.

-h, --human

Mostra o espaço ocupado em formato legível por humanos (Kb, Mb) ao invés de usar blocos.

-H

Como o anterior mas usa 1000 e não 1024 como unidade de cálculo.

-k

Mostra o espaço ocupado em Kbytes.

-m

Mostra o espaço ocupado em Mbytes.

-S, --separate-dirs

Não calcula o espaço ocupado por sub-diretórios.

-x

Não faz a contagem de diretórios em sistemas de arquivos diferentes do atual.

Exemplo: du -h, du -hc.

4.5 O comando sync

Grava os dados do cache de disco na memória RAM para todos os discos rígidos e flexíveis do sistema. O cache um mecanismo de aceleração que permite que um arquivo seja armazenado na memória ao invés de ser imediatamente gravado no disco, quando o sistema estiver ocioso, o arquivo é gravado para o disco. O GNU/Linux procura utilizar toda memória RAM disponível para o cache de programas acelerando seu desempenho de leitura/gravação.

sync

O uso do sync é útil em disquetes quando gravamos um programa e precisamos que os dados sejam gravados imediatamente para retirar o disquete da unidade. Mas o método recomendado é especificar a opção sync durante a montagem da unidade de disquetes (para detalhes veja fstab, Seção 5.9.1.

4.6 O comando `reboot`

Reinicia o computador.

4.7 O comando `halt`

Prepara o computador para desligamento.

4.8 O comando poweroff

Desliga o computador.

4.9 O comando shutdown

Desliga/reinicia o computador imediatamente ou após determinado tempo (programável) de forma segura. Todos os usuários do sistema são avisados que o computador será desligado . Este comando somente pode ser executado pelo usuário root ou quando é usada a opção -a pelos usuários cadastrados no arquivo /etc/shutdown.allow que estejam logados no console virtual do sistema.

shutdown [*opções*] [*hora*] [*mensagem*]

hora

Momento que o computador será desligado. Você pode usar HH:MM para definir a hora e minuto, MM para definir minutos, +SS para definir após quantos segundos, ou now para imediatamente (equivalente a +0).

O shutdown criará o arquivo /etc/nologin para não permitir que novos usuários façam login no sistema (com excessão do root). Este arquivo é removido caso a execução do shutdown seja cancelada (opção -c) ou após o sistema ser reiniciado.

mensagem

Mensagem que será mostrada a todos os usuários alertando sobre o reinicio/desligamento do sistema.

opções

-h

Inicia o processo para desligamento do computador.

-r

Reinicia o sistema

-c

Cancela a execução do shutdown. Você pode acrescentar uma mensagem avisando aos usuários sobre o fato.

-a

Permite que os nomes de usuários contidos no arquivo /etc/shutdown.allow possam utilizar o shutdown para reinicializar/desligar o sistema. Deve ser colocado um nome de usuário por linha. O limite máximo de usuários neste arquivo é de 32.

Este arquivo é útil quando o shutdown é usado para controlar o pressionamento das teclas CTRL+ALT+DEL no /etc/inittab.

-k

Simula o desligamento/reinicio do sistema, enviando mensagem aos usuários.

-f

Não executa a checagem do sistema de arquivos durante a inicialização do sistema. Este processo é feito gravando-se um arquivo /fastboot que é interpretado pelos scripts responsáveis pela execução do fsck durante a inicialização do sistema.

-F

Força a checagem do sistema de arquivos durante a inicialização. É gravado um arquivo chamado /forcefsck que é interpretado pelos scripts responsáveis pela execução do fsck durante a inicialização do sistema.

-n

Faz com que o shutdown ignore a execução do init fechando todos os processos.

-t [num]

Faz com que o shutdown envie um sinal de término aos processos e aguarde [num] segundos antes de enviar o sinal KILL.

O shutdown envia uma mensagem a todos os usuários do sistema alertando sobre o desligamento durante os 15 minutos restantes e assim permite que finalizem suas tarefas. Após isto, o shutdown muda o nível de execução através do comando init para 0 (desligamento), 1 (modo monousuário), 6 (reinicialização). É recomendado utilizar o símbolo "&" no final da linha de comando para que o shutdown seja executado em segundo plano.

Quando restarem apenas 5 minutos para o reinicio/desligamento do sistema, o programa login será desativado, impedindo a entrada de novos usuários no sistema.

O programa shutdown pode ser chamado pelo init através do pressionamento da combinação das teclas de reinicialização CTRL+ALT+DEL alterando-se o arquivo /etc/inittab. Isto permite que somente os usuários autorizados (ou o root) possam reinicializar o sistema.

Exemplos:

- "shutdown -h now" - Desligar o computador imediatamente.
- "shutdown -r now" - Reinicia o computador imediatamente.
- "shutdown 19:00 A manutenção do servidor será iniciada às 19:00" - Faz o computador entrar em modo monousuário (init 1) às 19:00 enviando a mensagem *A manutenção do servidor será iniciada às 19:00* a todos os usuários conectados ao sistema.

- "shutdown -r 15:00 O sistema será reiniciado às 15:00 horas" - Faz o computador ser reiniciado (init 6) às 15:00 horas enviando a mensagem *O sistema será reiniciado às 15:00 horas* a todos os usuários conectados ao sistema.
- shutdown -r 20 - Faz o sistema ser reiniciado após 20 minutos.
- shutdown -c - Cancela a execução do shutdown.
- shutdown -t 30 -r 20 - Reinicia o sistema após 20 minutos, espera 30 segundos após o sinal de término para enviar o sinal KILL a todos os programas abertos.

5 KERNEL E MÓDULOS

Este capítulo descreve em detalhes o que é o kernel, módulos, sua configuração e programas relacionados.

5.1 O Kernel

É o sistema operacional (o GNU/Linux) é ele controla os dispositivos e demais periféricos do sistema (como memória, placas de som, vídeo, discos rígidos, disquetes, sistemas de arquivos, redes e outros recursos disponíveis). Muitos confundem isto e chamam a distribuição de sistema operacional, isto é errado!

O *kernel* faz o controle dos periféricos do sistema e para isto ele deve ter o seu suporte incluído. Para fazer uma placa de som *Sound Blaster* funcionar, por exemplo, é necessário que o kernel ofereça suporte a este placa e você deve configurar seus parâmetros (como interrupção, I/O e DMA) com comandos específicos para ativar a placa e fazê-la funcionar corretamente. Existe um documento que contém quais são os periféricos suportados/não suportados pelo GNU/Linux, ele se chama Hardware-HOWTO.

Suas versões são identificadas por números como 2.0.36, 2.0.38, 2.1.10, 2.2.12, as versões que contém um número par entre o primeiro e segundo ponto são versões estáveis e que contém números ímpares neste mesmo local são versões instáveis (em desenvolvimento). Usar versões instáveis não quer dizer que ocorrerá travamentos ou coisas do tipo, mas algumas partes do kernel podem não estar testadas o suficiente ou alguns controladores podem ainda estar incompletos para obter pleno funcionamento. Se opera sua máquina em um ambiente crítico, prefira pegar versões estáveis do kernel.

Após inicializar o sistema, o kernel e seus arquivos podem ser acessados ou modificados através do ponto de montagem /proc.

Caso você tenha um dispositivo (como uma placa de som) que tem suporte no GNU/Linux mas não funciona veja a seção “Como adicionar suporte a Hardwares e outros dispositivos no kernel”.

5.2 Módulos

São partes do kernel que são carregadas somente quando são solicitadas por algum aplicativo ou dispositivo e descarregadas da memória quando não são mais usadas. Este recurso é útil por 2 motivos: Evita a construção de um kernel grande (estático) que ocupe grande parte da memória com todos os drivers compilados e permite que partes do kernel ocupem a memória somente quando forem necessários.

Os módulos do kernel estão localizados no diretório `/lib/modules/versão_do_kernel/*` (onde versão_do_kernel é a versão atual do kernel em seu sistema, caso seja 2.2.10 o diretório que contém seus módulos será `/lib/modules/2.2.10`.

Os módulos são carregados automaticamente quando solicitados através do programa `kmod` ou manualmente através do arquivo `/etc/modules`, `insmod` ou `modprobe`. Atenção: Não compile o suporte ao seu sistema de arquivos raiz como módulo, isto o tornará inacessível.

5.3 Adicionando suporte a Hardware e outros dispositivos no kernel

Quando seu hardware não funciona, mas você tem certeza que é suportado pelo GNU/Linux, é preciso seguir alguns passos para fazê-lo funcionar corretamente:

- Verifique se o kernel atual foi compilado com suporte ao seu dispositivo. Também é possível que o suporte ao dispositivo esteja compilado como módulo. Dê o comando "dmesg | more" para ver as mensagens do kernel durante a inicialização e verifique se aparece alguma coisa referente ao dispositivo que deseja instalar (alguma mensagem de erro, etc). Caso não apareça nada é possível que o driver esteja compilado como módulo, para verificar isto entre no diretório /lib/modules/versao_do_kernel e veja se encontra o módulo correspondente ao seu dispositivo (o módulo da placa *NE 2000* tem o nome de ne.o e o da placa *Sound Blaster* de sb.o, por exemplo).

- Caso o kernel não tiver o suporte ao seu dispositivo, você precisará recompilar seu kernel ativando seu suporte. Veja "Recompilando o Kernel"

- Caso seu hardware esteja compilado no kernel, verifique se o módulo correspondente está carregado (com o comando lsmod). Caso não estiver, carregue-o com o `modprobe` (por exemplo, `modprobe` sb io=0x220 irq=5 dma=1 dma16=5 mpuio=0x330), para detalhes veja a seção "`modprobe`"

O uso deste comando deverá ativar seu hardware imediatamente, neste caso configure o módulo para ser carregado automaticamente através do programa modconf ou edite os arquivos relacionados com os módulos (veja "Arquivos relacionados com o Kernel e Módulos"). Caso não tenha sucesso, será retornada uma mensagem de erro.

5.4 kmod

Este é o programa usado para carregar os módulos automaticamente quando são requeridos pelo sistema. Ele é um daemon que funciona constantemente fazendo a monitoração, quando verifica que algum dispositivo ou programa está solicitando o suporte a algum dispositivo, ele carrega o módulo correspondente.

Ele pode ser desativado através da recompilação do kernel, dando um kill no processo ou através do arquivo /etc/modules. Caso seja desativado, é preciso carregar manualmente os módulos através do `modprobe` ou `insmod`.

5.5 lsmod

Lista quais módulos estão carregados atualmente pelo kernel. O nome lsmod é uma contração de ls+módulos - Listar Módulos. A listagem feita pelo lsmod é uma alternativa ao uso do comando cat /proc/modules.

A saída deste comando tem a seguinte forma:

```
Module            Size  Pages    Used by
nls_iso8859_1     8000      1          1 (autoclean)
nls_cp437         3744      1          1 (autoclean)
ne                6156      2          1
8390              8390      2     [ne] 0
```

A coluna *Module* indica o nome do módulo que está carregado, a coluna *Used* mostra qual módulos está usando aquele recurso. O parâmetro *(autoclean)* no final da coluna indica que o módulo foi carregado manualmente (pelo `insmod` ou `modprobe`) ou através do kmod e será automaticamente removido da memória quando não for mais usado.

No exemplo acima os módulos *ne* e *8390* não tem o parâmetro *(autoclean)* porque foram carregados pelo arquivo /etc/modules. Isto significa que não serão removidos da memória caso estiverem sem uso.

Qualquer módulo carregado pode ser removido manualmente através do comandos `rmmod`.

5.6 insmod

Carrega um módulo manualmente. Para carregar módulos que dependem de outros módulos para que funcionem, você tem duas opções: Carregar os módulos manualmente ou usar o `modprobe` que verifica e carrega as dependências correspondentes.

A sintaxe do comando é:

`insmod [`*`módulo`*`] [`*`opções_módulo`*`]`

Onde:

- módulo: É o nome do módulo que será carregado.

- opções_módulo: Opções que serão usadas pelo módulo. Variam de módulo para módulo, alguns precisam de opções outros não, tente primeiro carregar sem opções, caso seja mostrada uma mensagem de erro verifique as opções usadas por ele. Para detalhes sobre que opções são suportadas por cada módulo, veja a sua documentação no código fonte do kernel em /usr/src/linux/Documentation

Exemplo: `insmod ne io=0x300 irq=10`

5.7 rmmod

Remove módulos carregados no kernel. Para ver os nomes dos módulos atualmente carregados no kernel digite lsmod e verifique na primeira coluna o nome do módulo. Caso um módulo tenha dependências e você tentar remover suas dependências, uma mensagem de erro será mostrada alertando que o módulo está em uso.

Exemplo: `rmmod ne`

5.8 modprobe

Carrega um módulo e suas dependências manualmente. Este comando permite carregar diversos módulos e dependências de uma só vez. O comportamento do `modprobe` é modificado pelo arquivo `/etc/modules.conf`.

A sintaxe deste comando é:

`modprobe [`*`módulo`*`] [`*`opções_módulo`*`]`

Onde:

- `módulo`: É o nome do módulo que será carregado.

- `opções_módulo`: Opções que serão usadas pelo módulo. Variam de módulo para módulo, alguns precisam de opções outros não, tente primeiro carregar sem opções, caso seja mostrada uma mensagem de erro verifique as opções usadas por ele. Para detalhes sobre que opções são suportadas por cada módulo, veja a sua documentação no código fonte do kernel em `/usr/src/linux/Documentation`

Nem todos os módulos são carregados corretamente pelo `modprobe`, o plip, por exemplo, mostra uma mensagem sobre porta I/O inválida mas não caso seja carregado pelo `insmod`.

Exemplos:

```
modprobe ne io=0x300 irq=10

modprobe sb io=0x220 irq=5 dma=1 dma16=5 mpuio=0x330
```

5.9 depmod

Verifica a dependência de módulos. As dependências dos módulos são verificadas pelos scripts em /etc/init.d usando o comando depmod -a e o resultado gravado no arquivo /lib/modules/versao_do_kernel/modules.dep. Esta checagem serve para que todas as dependências de módulos estejam corretamente disponíveis na inicialização do sistema. O comportamento do depmod pode ser modificado através do arquivo /etc/modules.conf . É possível criar a dependência de módulos imediatamente após a compilação do kernel digitando depmod -a [*versão_do_kernel*].

Exemplo: `depmod -a`

5.10 modconf

Este programa permite um meio mais fácil de configurar a ativação de módulos e opções através de uma interface através de menus. Selecione a categoria de módulos através das setas acima e abaixo e pressione enter para selecionar os módulos existentes. Serão pedidas as opções do módulo (como DMA, IRQ, I/O) para que sua inicialização seja possível, estes parâmetros são específicos de cada módulo e devem ser vistos na documentação do código fonte do kernel no diretório /usr/src/linux/Documentation. Note que também existem módulos com auto-detecção mas isto deixa o sistema um pouco mais lento, porque ele fará uma varredura na faixa de endereços especificados pelo módulo para achar o dispositivo. As opções são desnecessárias em alguns tipos de módulos.

As modificações feitas por este programa são gravadas no diretório /etc/modutils em arquivos separados como /etc/modutils/alias - alias de módulos, /etc/modutils/modconf - opções usadas por módulos, /etc/modutils/paths - Caminho onde os módulos do sistema são encontrados. Dentro de /etc/modutils é ainda encontrado um sub-diretório chamado arch que contém opções específicas por arquiteturas.

A sincronização dos arquivos gerados pelo modconf com o /etc/modules.conf é feita através do utilitário update-modules. Ele é normalmente executado após modificações nos módulos feitas pelo modconf.

Exercício:

Execute o modconf e remova o módulo de suporte à rede.

```
# modconf
```

Saia do modconf, e tente ver informações sobre a interface eth0

```
# ifconfig eth0
```

Veja que obteve uma mensagem de erro

Agora entre novamente no modconf, reinstale o módulo de rede e depois execute o “update-modules”

```
# modconf
# (entre, configure e saia do modconf )
# update-modules
```

Tente novamente ver as informações da placa de rede:

```
# ifconfig eth0
```

Veja que a placa está funcionando novamente

5.11 Recompilando o Kernel

Será que vou precisar recompilar o meu kernel? você deve estar se perguntando agora. Abaixo alguns motivos para esclarecer suas dúvidas:

- Melhora o desempenho do kernel. O kernel padrão que acompanha as distribuições GNU/Linux foi feito para funcionar em qualquer tipo de sistema e garantir seu funcionamento e inclui suporte a praticamente tudo. Isto pode gerar desde instabilidade até uma grade pausa do kernel na inicialização quando estiver procurando pelos dispositivos que simplesmente não existem em seu computador!

- A compilação permite escolher somente o suporte aos dispositivos existentes em seu computador e assim diminuir o tamanho do kernel, desocupar a memória RAM com dispositivos que nunca usará e assim você terá um desempenho bem melhor do que teria com um kernel pesado.

- Incluir suporte a alguns hardwares que estão desativados no kernel padrão (SMP, APM, Firewall, drivers experimentais, etc).

- Se aventurar em compilar um kernel (sistema operacional) personalizado em seu sistema.

- Impressionar os seus amigos, tentando coisas novas.

Serão necessários uns 70Mb de espaço em disco disponível para copiar e descompactar o código fonte do kernel e alguns pacotes de desenvolvimento

como o gcc, cpp, binutils, gcc-i386-gnu, bin86, make, dpkg-dev, perl, kernel-package (os três últimos somente para a distribuição Debian).

Para recompilar o kernel usando o método padrão, siga os seguintes passos:

1. Descompacte o código fonte do kernel (através do arquivo kernel-source-x.x.x.tar.gz) para o diretório /usr/src.
   ```
   # cd /usr/src
   # tar -zxvf kernel-source-x.x.x.tar.gz
   ```

2. Após isto, entre no diretório onde o código fonte do kernel foi instalado com cd /usr/src/linux (este será assumido o lugar onde o código fonte do kernel se encontra, se não houver este diretório, crie um link simbólico para o diretório onde está o kernel do Linux):
   ```
   # cd /usr/src
   # ln -s kernel-source-x.x.x linux
   # cd /usr/src/linux
   ```

3. Caso esteja sendo feita uma atualização do kernel, copie o arquivo ".config" da pasta do kernel antigo. Depois disso, use o comando make oldconfig para atualizar o arquivo de configuração.

4. Como usuário root, digite make menuconfig. Você também pode usar make config (configuração através de perguntas) ou make xconfig (configuração em modo gráfico). O make menuconfig e make xconfig precisam de pacotes adicionais para funcionar corretamente. Para o make menuconfig, instale o pacote "libncurses", através do dselect.
   Em cada opção do menu você pode marcar através da tecla de <espaço> as opções que você quer que estejam presentes no Kernel. As que podem ser carregadas como módulos aparecem como os sinais de menor e maior ("< >"), as outras aparecem com colchetes ("[ ]"). Para compilar como módulo, pressione a tecla de <espaço> até aparecer um "M" na opção. Para incluir a opção no kernel, deve aparecer um "*". Note que nem todos as opções podem ser compiladas como módulos.
   Se estiver em dúvida sobre a opção, digite "?" para ter uma explicação sobre o que aquela opção faz. Se não souber do que se trata, é recomendável não mudar a opção. Não se preocupe se esquecer de incluir o suporte a alguma coisa, você pode repetir o passo make menuconfig (todas as suas escolhas são gravadas no arquivo .config). Apenas deve tomar cuidado para não desmarcar opções importantes, que podem fazer com que o seu sistema não reinicie. Neste caso, é sempre bom fazer uma cópia do kernel anterior.
   Após, saia do menuconfig com <ESC> peça para salvar a configuração (Responda Yes quando o programa perguntar *"Do you*

wish to save the new kernel configuration?" (Você quer salvar a nova configuração de kernel?).

5. Digite o comando `make dep` para verificar as dependências dos módulos.
   ```
   # make dep
   ```

6. Digite o comando `make clean` para limpar construções anteriores do kernel.
   ```
   # make clean
   ```

7. Digite o comando `make zImage` para iniciar a compilação do kernel estático (outro comando compila os módulos). Aguarde a compilação, o tempo pode variar dependendo da quantidade de recursos que adicionou ao kernel, a velocidade de seu computador e a quantidade de memória RAM disponível.
   ```
   # make zImage
   ```

8. Caso tenha acrescentado muitos itens no Kernel, é possível que o comando `make zImage` falhe no final (especialmente se o tamanho do kernel estático for maior que 505Kb). Neste caso use `make bzImage`. A diferença entre *zImage* e *bzImage* é que o primeiro possui um limite de tamanho porque é descompactado na memória básica (recomendado para alguns Notebooks), já a *bzImage*, é descompactada na memória estendida e não possui as limitações da *zImage*.
   ```
   # make bzImage
   ```

9. Após terminada a compilação do kernel estático, execute `make modules` para compilar os módulos referentes àquele kernel. A velocidade de compilação pode variar de acordo com os motivos do passo anterior.
   ```
   # make modules
   ```

A compilação neste ponto está completa, você agora tem duas opções para instalar o kernel: Substituir o kernel anterior pelo recém compilado ou usar os dois. A segunda opção é recomendada caso você não tenha certeza de que o kernel funcionará corretamente e deseja iniciar pelo antigo no caso de alguma coisa dar errado. Neste caso, optaremos por manter o anterior por motivos de segurança. As instruções para substituir o kernel são logo após.

1. Execute o comando make modules_install para instalar os módulos recém compilados do kernel em `/lib/modules/versao_do_kernel`.

2. Copie o arquivo bzImage (ou zImage) que contém o kernel de `/usr/src/linux/arch/i386/boot/bzImage` para `/boot/vmlinuz-X.XX.XX` (onde X.XX.XX é a versão nova do kernel)
   ```
   # cp /usr/src/linux/arch/i386/boot/bzImage
   /boot/vmlinuz-X.XX.XX
   ```

3. Crie um link simbólico no diretório raíz (/) apontando para o novo kernel. Como exemplos será usado /vmlinuz-novo.
   ```
   # ln -s /boot/vmlinuz-X.XX.XX /vmlinuz-novo
   ```

4. Modifique o arquivo `/etc/lilo.conf` para incluir a nova imagem de kernel. Por exemplo:

   Antes da modificação:
   ```
   boot=/dev/hda
   compact
   lba32
   prompt
   timeout=200
   delay=200
   map=/boot/map
   install=/boot/boot.b
   image=/vmlinuz
       root=/dev/hda3
       label=linux
       read-only
   ```

   Depois da modificação:
   ```
   boot=/dev/hda
   compact
   lba32
   prompt
   timeout=200
   delay=200
   map=/boot/map
   install=/boot/boot.b
   image=/vmlinuz
       root=/dev/hda3
       label=linux
       read-only
   image=/vmlinuz-novo
       root=/dev/hda3
       label=linux-novo
       read-only
   ```

   Se você digitar “linux” no aviso de boot: do Lilo, o kernel antigo será carregado, caso digitar “linux-novo” o novo kernel será carregado.

5. Execute o comando `lilo` para gravar o novo setor de boot para o disco rígido.

6. Reinicie o computador

7. Carregue o novo kernel escolhendo a opção “linux-novo” no aviso de boot: do LILO. Caso tiver problemas, escolha a opção “linux” para iniciar com o kernel antigo e verifique os passos de configuração (o arquivo `lilo.conf` foi modificado corretamente?.

8. Para substituir o kernel, seguindo a partir do passo 8:

9. É recomendável renomear o diretório /lib/modules/versão_do_kernel para /lib/modules/versão_do_kernel.old, isto será útil para restauração completa dos módulos antigos caso alguma coisa der errado.

10. Execute o comando `make modules_install` para instalar os módulos do kernel recém compilado em `/lib/modules/versão_do_kernel`.

11. Copie o arquivo zImage que contém o kernel de `/usr/src/linux/arch/i386/boot/bzImage` para `/boot/vmlinuz-2.XX.XX` (*2.XX.XX* é a versão do kernel anterior)

12. Verifique se o link simbólico /vmlinuz aponta para a versão do kernel que compilou atualmente (com ls -la /). Caso contrário, apague o arquivo /vmlinuz do diretório raíz e crie um novo link com `ln -s /boot/vmlinuz-2.XX.Xx /vmlinuz` apontando para o kernel correto.

13. Execute o comando `lilo` para gerar um novo setor de partida no disco rígido.

14. Reinicie o sistema (`shutdown -r now`).

15. Caso tudo esteja funcionando normalmente, apague o diretório antigo de módulos que salvou e o kernel antigo de /boot. Caso algo tenha dado errado e seu sistema não inicializa, inicie a partir de um disquete, apague o novo kernel, apague os novos módulos, renomeie o diretório de módulos antigos para o nome original, ajuste o link simbólico /vmlinuz para apontar para o antigo kernel e execute o `lilo`. Após reiniciar seu computador voltará como estava antes.

Se quiser mais detalhes sobre a compilação do kernel, consulte o documento *kernel-howto*.

5.12 Arquivos relacionados com o Kernel e Módulos

Esta seção descreve os arquivos usados pelo kernel e módulos, a função de cada um no sistema, a sintaxe, etc.

5.12.1 `/etc/modules`

A função deste arquivo é carregar módulos especificados na inicialização do sistema e mantê-los carregado todo o tempo. É útil para módulos de placas de rede que precisam ser carregados antes da configuração de rede feita pela distribuição e não podem ser removidos quando a placa de rede estiver sem uso (isto retiraria seu computador da rede).

Seu conteúdo é uma lista de módulos (um por linha) que serão carregados na inicialização do sistema. Os módulos carregados pelo arquivo /etc/modules pode ser listados usando o comando lsmod.

Se o parâmetro auto estiver especificado como um módulo, o kmod será ativado e carregará os módulos somente em demanda, caso seja especificado noauto o programa kmod será desativado. O kmod é ativado por padrão nos níveis de execução 2 ao 5.

Ele pode ser editado em qualquer editor de textos comum ou modificado automaticamente através do utilitário modconf.

5.12.2 `modules.conf`

O arquivo /etc/modules.conf permite controlar as opções de todos os módulos do sistema. Ele é consultado pelos programas `modprobe` e depmod. As opções especificadas neste arquivo facilitam o gerenciamento de módulos, evitando a digitação de opções através da linha de comando.

Note que é recomendado o uso do utilitário modconf para configurar quaisquer módulos em seu sistema e o utilitário update-modules para sincronização dos arquivos gerados pelo modconf em /etc/modutils com o /etc/modules.conf (geralmente isto é feito automaticamente após o uso do modconf). Por este motivo não é recomendável modificá-lo manualmente, a não ser que seja um usuário experiente e saiba o que está fazendo.

Por exemplo: adicionando as linhas:

```
alias sound sb
options sb io=0x220 irq=5 dma=1 dma16=5 mpuio=0x330
```

Permitirá que seja usado somente o comando `modprobe sb` para ativar a placa de som.

5.13 Aplicando Patches no kernel

Patches são modificações geradas pelo programa diff em que servem para atualizar um programa ou texto. Este recurso é muito útil para os desenvolvedores, pois podem gerar um arquivo contendo as diferenças entre um programa antigo e um novo (usando o comando diff) e enviar o arquivo contendo as diferenças para outras pessoas.

As pessoas interessadas em atualizar o programa antigo, podem simplesmente pegar o arquivo contendo as diferenças e atualizar o programa usando o patch.

Isto é muito usado no desenvolvimento do kernel do GNU/Linux em que novas versões são lançadas freqüentemente e o tamanho kernel completo

compactado ocupa cerca de 18MB. Você pode atualizar seu kernel pegando um patch seguinte a versão que possui em ftp://ftp.kernel.org.

Para aplicar um patch que atualizará seu kernel 2.2.13 para a versão 2.2.14 você deve proceder da seguinte forma:

1. Descompacte o código fonte do kernel 2.2.13 em `/usr/src/linux` ou certifique-se que existe um link simbólico do código fonte do kernel para `/usr/src/linux`.

2. Copie o arquivo patch-2.2.14.gz de ftp://ftp.kernel.org para `/usr/src`.

3. Use o comando `gzip -dc patch-2.2.14|patch -p0 -N -E` para atualizar o código fonte em `/usr/src/linux` para a versão 2.2.14.

4. Alternativamente você pode primeiro descompactar o arquivo `patch-2.2.14.gz` com o gzip e usar o comando `patch -p0 -N -E < patch-2.2.14` para atualizar o código fonte do kernel. O GNU/Linux permite que você obtenha o mesmo resultado através de diferentes métodos, a escolha é somente sua.

Caso deseja atualizar o kernel 2.2.10 para 2.2.14, como no exemplo acima, você deverá aplicar os patches em seqüência (do patch 2.2.11 ao 2.2.14). Vale a pena observar se o tamanho total dos patches ultrapassa ou chega perto o tamanho do kernel completo.

6 HARDWARE

Hardware é tudo que diz respeito à parte física do computador. Nesta seção serão abordados assuntos relacionados com a configuração de hardwares, escolha de bons hardwares, dispositivos for Windows, etc.

6.1 Placa de expansão

É um circuito eletrônico encaixado na placa mãe que tem por objetivo adicionar novas funcionalidades ao computador. Esta placa pode ser, por exemplo, uma:

- placa de som - para fazer o computador emitir sons, músicas, ligar um joystick, etc.
- fax-modem - para enviar/receber fax, conectar-se a internet, BBS, acesso remoto, bina, etc.
- rede - para permitir a comunicação com outros computadores em uma rede interna
- controladora de periféricos - Para ligar discos rígidos, unidades de disquete, impressora, mouse, joystick, etc.
- SCSI - Para ligar unidades de disco rígidos e periféricos de alto desempenho.
- Controladora de Scanner - Para ligar um Scanner externo ao micro computador.

O encaixe da placa mãe que recebe as placas de expansão são chamados de *Slots*.

6.2 Nomes de dispositivos

Seria terrível se ao configurar CADA programa que utilize o mouse ou o modem precisássemos nos referir a ele pela IRQ, I/O, etc... para evitar isso são usados os *nomes de dispositivos*.

Os *nomes de dispositivos* no sistema GNU/Linux são acessados através do diretório /dev. Após configurar corretamente o modem, com sua porta I/O 0x2F8 e IRQ 3, ele é identificado automaticamente por /dev/ttyS1 (equivalente a COM2 no DOS). Daqui para frente basta se referir a /dev/ttyS1 para fazer alguma coisa com o modem.

Você também pode fazer um link de /dev/ttyS1 para um arquivo chamado /dev/modem usando: ln -s /dev/ttyS1 /dev/modem, faça a configuração dos seus

programas usando /dev/modem ao invés de /dev/ttyS1 e se precisar reconfigurar o seu modem e a porta serial mudar para /dev/ttyS3, será necessário somente apagar o link /dev/modem antigo e criar um novo apontando para a porta serial /dev/ttyS0.

Não será necessário reconfigurar os programas que usam o modem pois eles estão usando /dev/modem que está apontando para a localização correta. Isto é muito útil para um bom gerenciamento do sistema.

Abaixo uma tabela com o nome do dispositivo no GNU/Linux, portas I/O, IRQ, DMA e nome do dispositivo no DOS (os nomes de dispositivos estão localizados no diretório /dev):

Disp. Linux	Disp. DOS	IRQ	DMA	I/O
ttyS0	COM1	4	-	0x3F8
ttyS1	COM2	3	-	0x2F8
ttyS2	COM3	4	-	0x3E8
ttyS3	COM4	3	-	0x2E8
lp0	LPT1	7	3(ECP)	0x378
lp1	LPT2	5	3(ECP)	0x278
hda1	C:	14	-	0x1F0,0x3F6
hda2	D: *	14	-	0x1F0,0x3F6
hdc1	D: *	15	-	0x170,0x376

* A designação de letras de unidade do DOS não é padronizada como no GNU/Linux, e depende da existência de outras unidades físicas/lógicas no computador.

6.3 Configuração de Hardware

A configuração consiste em ajustar as opções de funcionamento dos dispositivos (periféricos) para comunicação com a placa mãe. Um sistema bem configurado consiste em cada dispositivo funcionando com suas portas I/O, IRQ, DMA bem definidas, não existindo conflitos com outros dispositivos. Isto também permitirá a adição de novos dispositivos ao sistema sem problemas.

É importante conhecer bem a configuração dos dispositivos do sistema para saber identificar e corrigir possíveis problemas de conflitos e o que deve ser modificado, caso seja necessário.

Os parâmetros usados para configurar dispositivos de hardware são a *IRQ*, *DMA* e *I/O*. Nem todo dispositivo usam estes três parâmetros, alguns apenas a *I/O* e *IRQ*, outros apenas a *I/O*, etc.

6.3.1 IRQ - Requisição de Interrupção

Existem dois tipos básicos de interrupções: as usadas por dispositivos (para a comunicação com a placa mãe) e programas (para obter a atenção do processador). As *interrupções de software* são mais usadas por programas,

incluindo o sistema operacional e *interrupções de hardware* mais usado por periféricos. Daqui para frente será explicado somente detalhes sobre interrupções de hardware.

Os antigos computadores 8086/8088 (XT) usavam somente 8 interrupções de hardware operando a 8 bits. Com o surgimento do AT foram incluídas 8 novas interrupções, operando a 16 bits. Os computadores 286 e superiores tem 16 interrupções de hardware numeradas de 0 a 15. Estas interrupções oferecem ao dispositivo associado a capacidade de interromper o que o processador estiver fazendo, pedindo atenção imediata.

As interrupções do sistema podem ser visualizadas no kernel com o comando `cat /proc/interrupts`. Abaixo um resumo do uso mais comum das 16 interrupções de hardware:

IRQ	Descrição
0	Timer do Sistema - Fixa
1	Teclado - Fixa
2	Segundo Controlador de Interrupção Programável – Fixa. Esta interrupção é usada como ponte para a IRQ 9, e não pode ser utilizada.
3	Normalmente usado por /dev/ttyS1
4	Normalmente usado por /dev/ttyS0
5	Normalmente a segunda porta paralela. Muitos micros não têm a segunda porta paralela, assim é comum encontrar placas de som e outros dispositivos usando esta IRQ.
6	Controlador de Disquete – Fixa Esta interrupção pode ser compartilhada com placas aceleradoras de disquete usadas em unidades de fita.
7	Primeira porta paralela. Muitas impressoras não usam IRQs.
8	Relógio em tempo real do CMOS – Fixa Não pode ser usado por nenhum outro dispositivo.
9	Interrupção livre para dispositivos
10	Interrupção livre para dispositivos
11	Interrupção livre para dispositivos
12	Normalmente livre para dispositivos. O mouse PS/2, quando presente, utiliza esta interrupção.
13	Processador de dados numéricos – Fixa. Não pode ser usada ou compartilhada
14	Primeira controladora IDE. Não pode ser compartilhada.
15	Segunda controladora IDE. Não pode ser compartilhada.

Dispositivos ISA, VESA, EISA, SCSI não permitem o compartilhamento de uma mesma IRQ, talvez isto ainda seja possível caso não haja outras opções disponíveis e/ou os dois dispositivos não acessem a IRQ ao mesmo tempo, mas isto é uma solução precária.

Conflitos de IRQ ocorrem quando dois dispositivos disputam uma mesma IRQ, e normalmente ocasionam a parada ou mal funcionamento de um dispositivo e/ou de todo o sistema. Para resolver um conflito de IRQs, deve-se conhecer quais IRQs estão sendo usadas por quais dispositivos (usando cat /proc/interrupts) e configurar as interrupções de forma que uma não entre em conflito com outra. Isto normalmente é feito através dos jumpers de placas ou através de software (no caso de dispositivos jumperless ou plug-and-play).

Dispositivos PCI são projetados para permitir o compartilhamento de uma mesma IRQ pois as manipulam de forma diferente. Se for necessário usar uma interrupção normal, o chipset (ou BIOS) mapeará a interrupção para uma interrupção normal do sistema (normalmente usando alguma interrupção entre a IRQ 9 e IRQ 12).

6.3.1.1 Prioridade das Interrupções

Cada IRQ no sistema tem um número que identifica a prioridade que será atendida pelo processador. Nos antigos sistemas XT as prioridades eram identificadas em seqüência de acordo com as interrupções existentes:

```
IRQ 0 1 2 3 4 5 6 7 8
PRI 1 2 3 4 5 6 7 8 9
```

Com o surgimento do barramento AT (16 bits), as interrupções passaram a ser identificadas da seguinte forma:

```
IRQ 0  1  2  (9  10  11  12  13  14  15)  3  4  5  6  7  8
PRI 1  2  3   4   5   6   7   8   9  10  11 12 13 14 15 16
```

Note que a prioridade segue em seqüência através da ponte da IRQ 2 para IRQ 9. Os dispositivos com prioridade mais baixa são atendidos primeiro, mas é uma diferença de desempenho praticamente imperceptível de ser notada nos sistemas atuais.

6.3.2 DMA - Acesso Direto a Memória

A *DMA* é usada para permitir a transferência de dados entre dispositivos I/O e a memória sem precisar do processador para fazê-lo. Ele livra esta carga do processador e resulta em uma rápida transferência de dados.

O PC padrão tem dois controladores de DMA. O primeiro controla os canais 0, 1, 2, 3 e o segundo os canais 4, 5, 6, 7, assim temos 8 canais. No entanto, o canal 4 é perdido porque é usado pelo *controlador de acesso direto a memória*. Os canais 0-3 são chamados de canais baixos porque podem somente mover um byte (*8 bits*) por transferência enquanto canais altos movem 2 bytes (*16 bits*) por transferência.

Os dados movidos usando a DMA **não** são movidos através do controlador de DMA. Isto oferece uma limitação porque a DMA somente pode mover dados entre os dispositivos (portas I/O) e a memória. Não é possível mover dados entre as portas ou entre a memória.

Existem dois controladores de DMA nos computadores AT e superiores. Ao contrário do que acontece com os dois controladores de IRQ, o primeiro controlador é ligado ao segundo e não o segundo ao primeiro. Os canais de DMA altos (5 ao 7) somente podem ser acessados por dispositivos de 16 bits (aqueles que utilizam a segunda parte do slot AT). Como resultado temos 8 canais de DMA, de 0 a 7, sendo que a DMA 4 é usada como ligação entre eles.

Os canais de DMA em uso no sistema podem ser visualizados com cat /proc/dma. Abaixo uma listagem de uso mais comum dos canais de DMA.

DMA	Barramento	Uso
0		Usada pelo circuito de refresh de memória DRAM
1	8/16 bits	Normalmente usado por placas de som (canal 8 bits), adaptadoras SCSI, placas de rede ou controladora de scanner.
2	8/16 bits	Normalmente usado pela controladora de disquetes ou controladoras de tapes.
3	8/16 bits	Usado pela porta paralela ECP, placa de som, controladoras de unidade de fita, controladoras SCSI ou controladora de scanner antiga.
4		Usada como ponte para a outra controladora de DMA (0-3)
5	16 bits	Normalmente usada pela placa de som (canal 16 bits), placas controladoras SCSI, placas de rede ou controladora de scanner.
6	16 bits	Placa de som (canal 16 bits), controladora de scanner ou placa de rede.
7	16 bits	Placa de som (canal 16 bits), controladora de scanner ou placa de rede.

Somente dispositivos ISA e derivados dele, como o EISA e VESA, usam os canais de DMA padrão. Os atuais dispositivos de alta taxa de transferência (normalmente PCI) possuem seu próprio controlador de DMA embutido, muito mais rápido do que a DMA padrão. Este controlador de DMA é chamado de *Bus Mastering* e muito usado nos discos rígidos atuais e pode atingir taxas de 33,3MB/s (no modo 2) e 66MB/s (no modo 4 - requer um cabo IDE com aterramento para evitar interferências de ruídos externos).

6.3.2.1 Conflitos de DMA

Um canal de DMA não pode ser compartilhado entre dispositivos. Ainda é possível configurar dois dispositivos para usarem um mesmo canal de DMA, desde que ele não seja usado ao mesmo tempo. Isto acontece com Scanners paralelos que compartilham a mesma porta paralela com a impressora. Se você for uma pessoa que explora os recursos de multitarefa de seu Linux e seu desempenho, evite estes tipos de dispositivos, prefira aqueles que utilizam seus próprios recursos.

Quando ocorre um conflito de DMA, os dados podem ser misturados e ocorre desde coisas estranhas até o travamento total do sistema. Este tipo de conflito é difícil de se diagnosticar, a não ser que o técnico seja experiente o bastante e tenha desconfiado do que o problema se trata...

6.3.3 I/O - Porta de Entrada/Saída

Cada dispositivo possui um endereço de porta. O endereço é uma localização da memória usada pelo computador para enviar dados ao dispositivo e onde o dispositivo envia dados ao computador. Ao contrários da IRQ e DMA, o dispositivo pode usar mais de uma porta de Entrada/Saída ou uma faixa de endereços. Por exemplo, uma placa de som padrão usa as portas 0x220, 0x330 e 0x388, respectivamente audio digital, midi e opl3.

As placas de rede normalmente transferem grandes quantidades de dados, assim ocupam uma faixa de endereços. Minha NE2000, por exemplo, ocupa a faixa de endereços 0x260 a 0x27F (0x260-0x27F). O tamanho da faixa de endereços varia de acordo com o tipo de dispositivo.

Os endereços de *I/O* em uso no sistema podem ser visualizados com o comando cat /proc/ioports.

Endereços das portas de entrada/saída não podem ser compartilhados

6.4 Hardwares configuráveis por jumpers, dip-switches, jumperless e Plug-and-Play.

6.4.1 Jumpers

Hardwares configuráveis por *jumpers* (pinos metálicos protegidos por uma capa plástica) tem sua configuração alterada através da colocação, retirada ou mudança de posição. Hardwares configuráveis por jumpers são os preferidos por técnicos de informática muito experientes.

Estes hardwares possuem a característica de somente terem seus parâmetros modificados através da mudança da posição dos jumpers da placa, desta forma se obtém uma configuração fixa (não podendo ser modificada por

qualquer tipo de programa) e o dispositivo estará sempre pronto para ser ativado após a inicialização de qualquer sistema operacional.

O único inconveniente é a necessidade de se retirar a placa do computador para se ter acesso aos jumpers de configuração, a não ser que estejam manualmente acessíveis. Alguns hardwares configuráveis através de jumpers podem também funcionar como Plug-and-Play, através de um ajuste da posição dos jumpers para Plug-and-Play.

Normalmente as placas controladoras SIDE, rede, bons modelos de fax-modens, placas de som, SCSI, etc., são configuradas por jumpers e possuem um mapa de configuração gravado em seu circuito impresso que explica as posições de como os jumpers devem ser posicionados para operar na configuração desejada. Normalmente é possível escolher uma entre vários tipos de configuração, mas é recomendado optar por valores padrões.

As disposição dos jumpers são normalmente definidas em *fechado/aberto* e *multi-posição*. Na disposição *fechado/aberto*, o jumper pode ou não ser colocado, definindo a configuração do dispositivo:

: : | : :

Esta disposição é facilmente encontrada na seleção de IRQ e I/O em placas de fax-modem.

Na disposição *multi-posição*, os pinos de encaixe são numerados de 1 a 3 (ou 1 a 4, 1 a 5, etc) e os pinos podem ou não ser colocados na placa e a posição que são colocados também influencia os valores escolhidos para o funcionamento do dispositivo (a posição 1-2 especificam um valor enquanto 2-3 especificam outro). A associação entre a posição dos jumpers e a configuração desejada é feita consultando o mapa desenhado no circuito impresso da placa ou o manual de instruções da placa.

A configuração de jumper através de multi-posição é normalmente usada em placas mãe para definir a *freqüência de operação do barramento*, a *freqüência de multiplicação* ou o *tipo do processador*.

Se não possuir o mapa de configuração de sua placa e/ou o manual de instruções, será necessário fazer um mapeamento manual da placa, mas para isto você precisará conhecer detalhadamente a configuração de portas I/O, DMA, IRQ usadas na máquina que será usada e anotar as diferenças obtidas através da modificação da pinagem do dispositivo. Isto não é fácil, mas técnicos de informática experientes conhecerão as armadilhas encontradas pelo mapeamento manual de placas e farão o esquema de configuração completo do dispositivo, obtendo um excelente manual de instruções. Nesta hora a experiência conta mais que o uso de programas de diagnóstico.

Outra característica de hardwares configurados através de jumpers é que raramente apresentam problemas de funcionamento, a não ser que seus parâmetros como IRQ, DMA, ou I/O estejam em conflitos com outro dispositivo, mas isso não é culpa do fabricante e nem mesmo do dispositivo...

6.4.2 Dip-Switches

É a mesma coisa que os hardwares configuráveis por jumpers exceto que são usados *dip-switches* no lugar de jumpers. O *dip-switches* é um conjunto de chaves numeradas que podem ser colocadas para cima ou para baixo (como um disjuntor ou vários interruptores *LIGA/DESLIGA* colocados um ao lado do outro) para se modificar a configuração do dispositivo.

Normalmente as chaves estão acessíveis na parte metálica da placa (onde os hardwares são conectados) para permitir a fácil mudança de configuração sem retirar a placa. É ainda comum encontrar isto em algumas placas de fax-modem.

6.4.3 Jumperless (sem jumper)

Os hardwares *jumperless* não possuem jumpers e são configurados através de um programa que acompanha a própria placa. Neste programa é escolhida a IRQ, DMA, I/O e a configuração é salva na própria placa ou restaurada após cada inicialização por um programa carregado na memória. Devido a configuração via software, se obtém uma configuração fixa com muito mais facilidade do que via jumpers (por não haver a necessidade de se retirar a placa).

A maioria das placas jumperless podem funcionar também como Plug-and-Play. Existem muitas placas de rede, fax-modem, scanner jumperless no mercado.

6.4.4 Plug-and-Play

O *Plug-and-Play* é um protocolo que lê os valores de operação disponíveis para a placa e permitem que o usuário possa especificar facilmente qual será sua IRQ, DMA, I/O.

A diferença em relação ao modo jumperless é que o mesmo programa de configuração Plug-and-Play permite configurar todas as placas Plug-and-Play e a placa somente recebe os valores de IRQ, DMA e I/O após ser ativada por este programa, normalmente o isapnp no Linux. Isto significa que a placa não tem nenhum parâmetro de IRQ, DMA e I/O na partida do sistema.

Desta forma, somente sistemas operacionais que possuem suporte ao Plug-and-Play (como o GNU/Linux, Windows) ou programas acionadores PnP (como o ICU para o DOS) podem ativar e usar estes tipos de placas.

Placas Plug-and-Play permitem muita flexibilidade de configuração de dispositivos. O programa usado para a configuração de placas Plug-and-Play no GNU/Linux é o isapnp e a configuração de todas as placas Plug-and-Play são definidas no arquivo /etc/isapnp.conf.

Veja a próxima seção para entender como funciona o arquivo de configuração isapnp.conf e assim poder ativar seu dispositivo Plug-and-Play.

6.4.4.1 Entendendo o arquivo de configuração isapnp.conf

Segue abaixo um exemplo de arquivo /etc/isapnp.conf gerado através do pnpdump para a configuração de uma placa de Som Sound Blaster com porta IDE embutida no GNU/Linux.

O objetivo é configurar a placa Sound Blaster para operar na seguinte configuração:

- IO=0x220
- IRQ=5
- DMA=1
- DMA16=5
- MIDI=0x330
- OPL=0x388
- IDE operando como placa controladora quartenária na porta 0x168/0x36e - Nós queremos ligar um HD na placa de som, **SIM** o GNU/Linux permite isso, e ele será configurado como /dev/hdg1
- JOYSTICK na porta 0x220 - É bom para jogos e controle do xmms

Observe que as linhas iniciando com # são apenas comentários e não serão interpretadas pelo isapnp:

```
# $Id: pnpdump.c,v 1.21 1999/12/09 22:28:33 fox Exp $
# Release isapnptools-1.21 (library isapnptools-1.21)
#
# Para detalhes do formato do arquivo de saída, veja a página de
# manual do isapnp.conf
#
# A seção abaixo faz o isolamento da placa através da BIOS
(normalmente
# não precisa ser alterado).  Com a configuração abaixo, os dados
sobre
# dispositivos serão obtidos diretamente da BIOS.
# Em placas mãe que não suportam Plug-and-Play, é necessário apenas o
# parâmetro (ISOLATE) para que o isapnp possa assumir totalmente
# controle para identificação dos dispositivos Plug-and-Play
    (READPORT 0x0273)
    (ISOLATE PRESERVE)
    (IDENTIFY *)
    (VERBOSITY 2)
    (CONFLICT (IO FATAL)(IRQ FATAL)(DMA FATAL)(MEM FATAL)) # ou
WARNING
# Card 1: (serial identifier fc 10 01 fb 5d 28 00 8c 0e)
# Vendor Id CTL0028, Serial Number 268565341, checksum  0xFC.
# Version 1.0, Vendor version 1.0
# ANSI string -->Creative SB16 PnP<--
#
# Descomente os valores desejados abaixo, selecionando a
```

```
# configuração requerida.
# Note que o valor padrão equivale ao primeiro parâmetro
# disponível (Minimum)
# "(CONFIGURE" inicia um bloco de configuração e finaliza com "(ACT
Y)"
# Para ativar as configurações selecionadas, basta descomentar a linha
# "#(ACT Y)" no final do bloco de configuração.
    (CONFIGURE CTL0028/268565341 (LD 0
#     ANSI string -->Audio<--
# Pela string acima, esta é a configuração de Audio da Sound Blaster
# Hora de múltiplas escolhas, escolha apenas uma!
#     Inicia funções dependentes, classificada por prioridade
aceitável
#
#       IRQ 5, 7 ou 10.
     (INT 0 (IRQ 5 (MODE +E)))
# Foi especificada a IRQ 5 na configuração acima
#       Primeiro canal DMA 0, 1 ou 3.
#             Somente DMA de 8 bits
#             Dispositivo lógico não é um bus master
#             DMA may execute in count by byte mode
#             DMA may not execute in  count by word mode
#             DMA channel speed in compatible mode
     (DMA 0 (CHANNEL 1))
# O valor da DMA 8 bits padrão é 0 (o mais baixo), mas este não
# é o valor que desejamos.  Ajustamos o valor para 1.
#       Next DMA channel 5, 6 or 7.
#             16 bit DMA only
    #             Logical device is a bus master
    #             DMA may not execute in count by byte mode
    #             DMA may execute in count by word mode
    #             DMA channel speed in compatible mode
    (DMA 1 (CHANNEL 5))
    # O canal DMA 16 bits desejado para a Sound Blaster é o 5.   Apenas
    # descomentamos a linha acima.
    #       Logical device decodes 16 bit IO address lines
    #             Minimum IO base address 0x0220
    #             Maximum IO base address 0x0280
    #             IO base alignment 32 bytes
    #             Number of IO addresses required: 16
    (IO 0 (SIZE 16) (BASE 0x0220))
    # Apenas descomentamos a linha.
    #       Logical device decodes 16 bit IO address lines
    #             Minimum IO base address 0x0300
    #             Maximum IO base address 0x0330
    #             IO base alignment 48 bytes
    #             Number of IO addresses required: 2
    (IO 1 (SIZE 2) (BASE 0x0330))
    # O valor padrão é 0x0300 para a porta MIDI, mas nós desejamos
usar o
    # valor 0x0330.  Descomentamos a linha e alteramos o valor da I/O.
    #       Logical device decodes 16 bit IO address lines
    #             Minimum IO base address 0x0388
    #             Maximum  IO base address 0x0388
    #             IO base alignment 1 bytes
    #             Number of IO addresses required: 4
    (IO 2 (SIZE 4) (BASE 0x0388))
    # Apenas descomentamos a linha.  0x0388 é um valor padrão para OPL
    #     Fim de funções dependentes
     (NAME "CTL0028/268565341[0]{Audio              }")
```

```
    (ACT Y) #Descomentamos para ativar este bloco de configuração
acima     ))

   ########################################
   # Logical device id CTL2011
   #
   # Descomente os valores desejados abaixo, selecionando
   # a configuração requerida.
   # Note que o valor padrão equivale ao primeiro parâmetro
   # disponível (Minimum)
   # "(CONFIGURE" inicia um bloco de configuração e finaliza com
   # "(ACT Y)"
   # Para ativar as configurações selecionadas, basta descomentar
   # a linha
   # "#(ACT Y)" no final do bloco de configuração.
     (CONFIGURE CTL0028/268565341 (LD 1

   #     Compatible device id PNP0600
   #     ANSI string -->IDE<--
   # Pela string acima sabemos que esta é  a configuração da IDE
   # embutida na SB
   # Hora de múltiplas escolhas, escolha apenas uma!
   #     Inicia funções dependentes: Prioridade Preferida
   #       IRQ 10.
     (INT 0 (IRQ 10 (MODE +E)))
   # Descomentamos e aceitamos o valor acima, pois  não entra em
   # conflito com nenhum outro dispositivo do sistema.
   #       Logical device decodes 16 bit IO address lines
   #             Minimum IO base address 0x0168
   #             Maximum IO base address 0x0168
     (IO 0 (SIZE 8) (BASE 0x0168 ))
   # Descomentamos e aceitamos o valor acima, pois não entra em
   # conflito com nenhum outro dispositivo do sistema.
   #       Logical device decodes 16 bit IO address lines
   #             Minimum IO base address 0x036e
   #             Maximum IO base address 0x036e
   #             IO base alignment 1 bytes
   #             Number of IO addresses required: 2
     (IO 1 (SIZE 2) (BASE 0x036e))
   # Descomentamos e aceitamos o valor acima, pois não entra em
   # conflito com nenhum outro disposi tivo do sistema.
   #     End dependent functions
      (NAME "CTL0028/268565341[1]{IDE                }")
      (ACT Y) # Descomentando esta linha, a placa IDE da
              # Sound Blaster passará a funcionar como
              # IDE quartenária (de acordo com os recursos passados)
     ))
     ########################################
     # Logical device id CTL7001
     #
     # Descomente os valores desejados abaixo, selecionando
     # a configuração requerida.
     # Note que o valor padrão equivale ao primeiro parâmetro
     # disponível (Minimum)
     # "(CONFIGURE" inicia um bloco de configuração e finaliza
     # com "(ACT Y)"
     # Para ativar as configurações selecionadas, basta descomentar
     # a linha
     # "#(ACT Y)" no final do bloco de configuração.
     (CONFIGURE CTL0028/268565341 (LD 3
     #     Compatible device id PNPb02f
```

```
#     ANSI string -->Game<--
# Pela string acima sabemos que é a Entrada para Joystick
#     Logical device decodes 16 bit IO address  lines
#         Minimum IO base address 0x0200
#         Maximum IO base address 0x0200
#         IO base alignment 1 bytes
#         Number of IO addresses required: 8
 (IO 0 (SIZE 8) (BASE 0x0200))
 (NAME "CTL0028/268565341[3]{Jogo               }")
 (ACT Y) # Sem muitos comentários...  descomentamos a linha
         # IO acima e
         # ativamos a configuração (descomentando (ACT Y)).
         # A diferença é que especificamos o nome GAME
         # para o recurso através da
         # linha (NAME "CTL0028/268565341[3]{Jogo            }")
         # Este nome será mostrado quando o Joystick for ativado
))
# Returns all cards to the 'Wait for Key' state
(WAITFORKEY)
```

Note ainda que o isapnp.conf gerado através do pnpdump contém vários tipos de prioridades de configuração para o mesmo bloco de configuração e a prioridade que usamos acima foi priority acceptable para o bloco de audio da Sound Blaster e priority preferred para a porta IDE e Joystick. Os tipos de prioridades disponíveis são:

- priority preferred - Configuração preferida para o funcionamento do hardware. É a recomendada pelo fabricante do hardware e também recomendável se você não tem muita experiência na configuração de hardwares, pois lista somente uma configuração por recurso. Se a placa entrar em conflito com outras placas usando priority preferred, tente a priority acceptable.

- priority acceptable - Lista todas as configurações aceitas pelo seu hardware. Ela é minha opção preferida, pois permite analisar dinamicamente todas as configurações permitidas pelo hardware e escolher qual é a mais adequada para funcionar sem problemas no sistema.

- priority functional - Pode conter 1 ou mais blocos de prioriade funcional por hardware. Note que alguns recursos do hardware podem não estar disponível neste tipo de prioridade. É útil para uso em casos de conflito, quando o hardware pode ser colocado em funcionamento de forma alternativa ou parcial.

Após a gravação do arquivo /etc/isapnp.conf, basta você digitar isapnp /etc/isapnp.conf para ativar a configuração dos dispositivos listados com as configurações que você escolheu. Se o isapnp lhe mostrar mensagens de conflito ou qualquer outro problema, verifique as configurações do hardware e modifique, se necessário. Depois execute novamente o /etc/isapnp.conf. Para detalhes sobre outros parâmetros não explicados aqui, veja a página de manual do isapnp.conf.

A maioria das distribuições GNU/Linux configura os dispositivos Plug-and-Play existentes neste arquivo automaticamente na inicialização (como é o caso da Debian e a Red Hat). Se este não for o seu caso, coloque a linha isapnp /etc/isapnp.conf em um dos scripts de inicialização de sua distribuição.

6.5 Conflitos de hardware

Ocorre quando um ou mais dispositivos usam a mesma *IRQ*, *I/O* ou *DMA*. Um sistema com configurações de hardware em conflito tem seu funcionamento instável, travamentos constantes, mal funcionamento de um ou mais dispositivos e até mesmo, em casos mais graves, a perda de dados.

Sempre que possível conheça e utilize os valores padrões para a configuração de periféricos, isto pode livrá-lo de conflitos com outros dispositivos e mal funcionamento do sistema. Alguns programas de diagnóstico ou de auto-detecção podem não localizar seu dispositivo caso ele esteja usando um valor muito diferente do padrão.

Para resolver conflitos de hardware será necessário conhecer a configuração de cada dispositivo em seu sistema. Os comandos cat /proc/interrupts, cat /proc/dma e cat /proc/ioports podem ser úteis para se verificar as configurações usadas.

Lembre-se que o barramento PCI permite o compartilhamento de IRQs entre placas PCI.

6.6 Barramento

O tipo de *slot* varia de acordo com o barramento usado no sistema, que pode ser um(s) do(s) seguinte(s):

- ISA 8 bits: Industry Standard Architecture - É o padrão mais antigo, encontrado em computadores PC/XT.
- ISA 16 bits: Evolução do padrão ISA 8 Bits, possui um conector maior e permite a conexão de placas de 8 bits. Sua taxa de transferência chega a 2MB/s.
- VESA: Video Electronics Standard Association - É uma interface feita inicialmente para placas de vídeo rápidas. O barramento VESA é basicamente um ISA com um encaixe extra no final. Sua taxa de transferência pode chegar a 132MB/s.
- EISA: Enhanced Industry Standard Architecture - É um barramento mais encontrado em servidores. Tem a capacidade de bus mastering, que possibilita a comunicação das placas sem a interferência da CPU.

- MCA: Micro Channel Architecture - Barramento 32 bits proprietário da IBM. Você não pode usar placas ISA nele, possui a característica de bus mastering, mas pode procurar por dispositivos conectados a ele, procurando configuração automática. Este barramento estava presente no PS/1 e PS/2, hoje não é mais usado.

- PCI: Peripheral Component Interconnect - É outro barramento rápido produzido pela Intel com a mesma velocidade que o VESA. O barramento possui um chipset de controle que faz a comunicação entre os slots PCI e o processador. O barramento se configura automaticamente (através do Plug-and-Play). O PCI é o barramento mais usado por Pentiums e está se tornando uma padrão no PC.

- AGP: Accelerated Graphics Port - É um novo barramento criado exclusivamente para a ligação de placas de video. É um slot marrom (em sua maioria) que fica mais separado do ponto de fixação das placas no chassis (comparado ao PCI). Estas placas permitem obter um desempenho elevado de vídeo se comparado as placas onboards com memória compartilhada e mesmo PCI externas. O consumo de potência em placas AGP x4 podem chegar até a 100W, portanto é importante dimensionar bem o sistema e ter certeza que a fonte de alimentação pode trabalhar com folga.

- PCMCIA: Personal Computer Memory Card International Association - É um slot especial usado para conexões de placas externas (normalmente revestivas de plástico) e chamadas de *cartões PCMCIA*. Estes cartões podem adicionar mais memória ao sistema, conter um fax-modem, placa de rede, disco rígido, etc.
  Os cartões PCMCIA são divididos em 3 tipos:
  Tipo 1: Tem a espessura de 3.3 milímetros, e podem conter mais memória RAM ou memória Flash.
  Tipo 2: Tem a espessura de 5 milímetros e capacidade de operações I/O. É um tipo usado para placas de fax-modem, rede, som. Computadores que aceitam cartões PCMCIA do tipo 2, mantém a compatibilidade com o tipo 1.
  Tipo 3: Tem a espessura de 10.5 milímetros e normalmente usado para discos rígidos PCMCIA. Slots PCMCIA do tipo 3 mantém a compatibilidade com o tipo 2 e 1.

- AMR: Audio Modem Raise - Pequeno barramento criado pela Intel para a conexão de placas de som e modem. Placas de som e modem AMR usam o HSP (host signal processor) e são como as Placas on-board e todo o processamento é feito pela CPU do computador. Sua vantagem é o preço: um modem ou placa de som AMR custa em torno de R$ 25,00.

- CNR: Communication and Networking Rise - Pequeno barramento criado pela Intel para a conexão de placas de som, modems e placas de rede. Este é um pequenino slot marrom que é localizado no ponto de fixação das placas no chassis do gabinete. Elas são como as Placas on-board e todo o processamento é feito pela CPU do computador.

6.7 Placas on-board / off-board

Placas *on-board* são embutidas na placa mãe (*motherboard*). Placas *off-board* são placas externas encaixadas nos slots de expansão da placa mãe.

No início da era do PC/XT todos as placas eram embutidas na placa mãe (na época eram somente a placa de vídeo e controladora). Com o surgimento do padrão AT, diversas empresas de informática desenvolveram dispositivos concorrentes e assim o usuário tinha a liberdade de escolha de qual dispositivo colocar em sua placa mãe (ou o mais barato ou o de melhor qualidade e desempenho), isto permitiu a adição de periféricos de qualidade sem romper com seu orçamento pessoal (comprando uma placa de som, depois uma de fax-modem, placa de vídeo melhor, etc).

Atualmente parece que voltamos ao ponto de partida e tudo vem embutido na placa mãe (*on-board*) e o usuário não tem como escolher qual dispositivo usar em seu computador. É muito difícil (praticamente impossível) encontrar uma placa mãe que satisfaça completamente as necessidades do usuário ou recomendações de um bom técnico de informática (a não ser que seja um técnico experiente e encontre alguma alternativa).

Certamente o único dispositivo que funciona melhor se embutido na placa mãe é a *placa controladora de periféricos*. Esta placa é usada para se conectar unidades de disquete, discos rígidos, CD-ROM, portas seriais, paralelas, joystick ao computador. Os HDs conectados em uma controladora embutida conseguem ter um desempenho muito maior do que em placas conectadas externamente, sem causar nenhum tipo de problema. Felizmente os modelos atuais de placas mãe trazem a placa controladora de periféricos embutida.

Hardwares embutidos na placa mãe (como fax-modem, vídeo, som) são em média 30% mais baratos que os vendidos separadamente mas quase sempre são usados dispositivos de baixo desempenho e qualidade para reduzir o preço da placa mãe e quase sempre usados hardwares For Windows.

Hoje em dia por causa do preço da placa mãe, é comum encontrar pessoas que verificam somente o preço e sequer procuram saber ou conhecem a qualidade das placas embutidas na placa mãe. Pior ainda é encontrar vendedores despreparados que sequer sabem explicar o porque que uma placa de som Sound Blaster 64 é mais cara que uma de modelo genérico...

Certa vez fiz um teste de desempenho em um jogo chamado *Network Rally* do *DOS* com minha máquina Pentium 120MHz (só com a *placa controladora* embutida), 16 MB RAM, placa de som Sound Blaster 16, placa de vídeo Trident 9680 com 1MB *versus* um computador Pentium 200 MMX, 32 MB RAM, placa de vídeo embutida (usando 2 MB de memória compartilhada), fax modem Rockwell embutido, e som CMI 8330 também embutido.

O resultado foi que o jogo rodava perfeito em meu pentium 120MHz e no outro computador com o som pipocando e imagem apresentando paradas. O problema é que em dispositivos de baixa qualidade e baratos, sua carga de processamento é jogada para o processador, resultando em menos potência para executar os programas (veja a próxima seção, **Error! Hyperlink reference not valid.** para maiores detalhes sobre o problema). A memória de vídeo compartilhada quer dizer que parte da memória RAM é usada para memória de vídeo ao invés de uma memória DRAM específica e desenvolvida exclusivamente para aceleração de vídeo. Isto traz mais lentidão pois a memória de vídeo (RAM) também será acessada pelo barramento do computador, envolvendo mais carga para o processador, etc. A técnica de memória compartilhada é exclusiva de placas de vídeo embutidas.

Outro periférico que traz problemas e muita carga para o processador é o fax-modem for Windows, HSP, AMR, micromodem, etc. A maioria destes periféricos se recusam a funcionar em computadores inferiores ao Pentium 150, não trazem seu chip de processamento e o pior: o chip UART. Isto faz com que o periférico, mesmo marcando conexão a 57.600 ou mais tenha um desempenho de até duas vezes menor que um fax-modem inteligente com chip de processamento próprio e UART (sem contar com os controles internos do modem, como os protocolos de correção de erros, e sua extensa interface de programação via comandos). A economia, neste caso, será paga em sua conta telefônica.

Outra vantagem de fax-modens inteligentes é que os modelos atuais vem com *FlashBios* o que significa que podem ser reprogramados facilmente para passar de 33.600 para 57.600 sem trocar a placa, ou aceitarem novas tendências de tecnologia. Para detalhes veja **Error! Hyperlink reference not valid.**

Se você estiver em uma situação destas, certamente os computadores de menor potência e com hardwares inteligentes (que possuem seus próprios chips de controle e processamento) terão um desempenho muito melhor. Mas também existem placas embutidas que tem a mesma qualidade de placas separadas (como alguns modelos de placas mãe que trazem a *Sound Blaster* embutida). O preço pode ser maior, mas você estará pagando por um dispositivo de melhor qualidade e que certamente trará benefícios a você e ao seu sistema.

Consulte um técnico em informática experiente para te indicar uma placa mãe de bom preço e de qualidade. É muito comum encontrar falta de profissionalismo em pessoas que não sabem distinguir as características, funções e vantagens entre uma placa de boa qualidade e um hardware for Windows a não ser o preço mais barato.

6.8 Hardwares específicos ou "For Windows"

Esta seção foi retirada do manual de instalação da Debian GNU/Linux. Uma tendência que perturba é a proliferação de Modems e impressoras específicos para Windows. Em muitos casos estes são especialmente fabricados para operar com o Sistema Operacional Microsoft Windows e costumam ter a legenda WinModem, for Windows, ou Feito especialmente para computadores baseados no Windows.

Geralmente estes dispositivos são feitos retirando os processadores embutidos daquele hardware e o trabalho deles são feitos por drivers do Windows que são executados pelo processador principal do computador. Esta estratégia faz o hardware menos expansível, mas o que é poupado não é passado para o usuário e este hardware pode até mesmo ser mais caro quanto dispositivos equivalentes que possuem inteligência embutida.

Você deve evitar o hardware baseado no Windows por duas razões:

1. O primeiro é que aqueles fabricantes não tornam os recursos disponíveis para criar um driver para Linux. Geralmente, o hardware e a interface de software para o dispositivo é proprietária, e a documentação não é disponível sem o acordo de não revelação, se ele estiver disponível. Isto impede seu uso como software livre, desde que os escritores de software grátis descubram o código fonte destes programas.

Você pode ajudar a reverter esta situação encorajando estes fabricantes a lançarem a documentação e outros recursos necessários para nós desenvolvermos drivers para estes hardwares, mas a melhor estratégia é simplesmente evitar estes tipos de hardwares até que ele esteja listado no HOWTO de hardwares compatíveis com Linux.

2. A segunda razão é que quando estes dispositivos têm os processadores embutidos removidos, o sistema operacional deve fazer o trabalho dos processadores embutidos, freqüentemente em prioridade de tempo real, e assim a CPU não esta disponível para executar programas enquanto ela esta controlando estes dispositivos. Um exemplo típico disso são os Modems for Windows; Além da carga jogada na CPU, o dispositivo não possui o chip UART 16550, que é essencial para uma boa taxa de transferência do modem. O que alguns dispositivos fazem é a emulação deste chip exigindo no mínimo uma CPU Pentium de 166 MHz para operar adequadamente nesta taxa de transmissão. Mesmo assim, devido a falta do chip UART, um modem destes iniciar uma transmissão de arquivo a 57.600, a tendência é sua taxa de transferência ir caindo na medida que um arquivo é transferido (até se estabilizar em 21/25 Kbps).

Assim o usuário típico do Windows não obtém um multi-processamento tão intensivo como um usuário do Linux, o fabricante espera que aquele usuário do Windows simplesmente não note a carga de trabalho que este

hardware põe naquela CPU. No entanto, qualquer sistema operacional de multi-processamento, até mesmo Windows 95 / 98 ou NT, são prejudicados quando fabricantes de periféricos retiram o processador embutido de suas placas e colocam o processamento do hardware na CPU.

Note que hoje já existem muitos drivers para WinModems e outros hardwares for Windows para o Linux. Veja a lista de hardwares compatíveis no HARDWARE-HOWTO ou procure o driver no site do fabricante de seu dispositivo. Mesmo assim a dica é evitar hardwares for Windows e comprar hardwares inteligentes onde cada um faz sua função sem carregar a CPU.

6.9 Dispositivos específicos para GNU/Linux

Esta seção foi retirada do manual de instalação da Debian GNU/Linux. Existem diversos vendedores, que vendem sistemas com a Debian ou outra distribuição do GNU/Linux pré-instaladas. Você pode pagar mais para ter este privilégio, mas compra um nível traqüilidade, tendo certeza que seu hardware é bem compatível com GNU/Linux. Praticamente todas as placas que possuem processadores próprios funcionam sem nenhum problema no Linux (algumas placas da Turtle Beach e mwave tem suporte de som limitado).

Se não estiver comprando um computador com GNU/Linux instalado, ou até mesmo um computador usado, é importante verificar se o hardware existente é suportado pelo kernel do GNU/Linux. Verifique se seu hardware é listado no *Hardware Compatibility HOWTO*, na documentação do código fonte do kernel no diretório Documentation/sound ou consulte um técnico de GNU/Linux experiente.

Informe seu vendedor (se conhecer) saber que o que está comprando é para um sistema GNU/Linux. Desta forma isto servirá de experiência para que ele poderá recomendar o mesmo dispositivo a outras pessoas que procuram bons dispositivos para sistemas GNU/Linux. Apóie vendedores de hardwares amigos do GNU/Linux.

7 IMPRESSÃO

Este capítulo descreve como imprimir em seu sistema GNU/Linux e as formas de impressão via spool, rede, gráfica, etc.

Antes de seguir os passos descritos neste capítulo, tenha certeza que seu kernel foi compilado com o suporte a impressora paralela ativado, caso contrário até mesmo a impressão direta para a porta de impressora falhará. Verifique se as opções “General Setup ---> < > Parallel Port Support”, “General Setup ---> < > PC-Style hardware” e “Character Devices ----> < > Parallel Printer Support” estão presentes no kernel.

Para detalhes veja a seção Recompilando o Kernel

7.1 Portas de impressora

Uma porta de impressora é o local do sistema usado para se comunicar com a impressora. Em sistemas GNU/Linux, a porta de impressora é identificada como lp0, lp1, lp2 no diretório /dev, correspondendo respectivamente a LPT1, LPT2 e LPT3 no DOS e Windows. Recomendo que o suporte a porta paralela esteja compilado como módulo no kernel.

```
General setup    ---->
      <M>  Parallel port support
        <M> PC-Style hardware
Character Devices ---->
      <M> Parallel Printer Support
```

7.2 Imprimindo diretamente para a porta de impressora

Isto é feito direcionando a saída ou o texto com > diretamente para a porta de impressora no diretório /dev.

Supondo que você quer imprimir o texto contido do arquivo trabalho.txt e a porta de impressora em seu sistema é /dev/lp0, você pode usar os seguintes comandos:

- `cat trabalho.txt > /dev/lp0` - Direciona a saída do comando cat para a impressora.
- `cat <trabalho.txt > /dev/lp0` - Faz a mesma coisa que o acima.
- `cat -n trabalho.txt > /dev/lp0` - Numera as linhas durante a impressão.

- `head -n 30 trabalho.txt > /dev/lp0` - Imprime as 30 linhas iniciais do arquivo.

- `cat trabalho.txt | tee /dev/lp0` - Mostra o conteúdo do cat na tela e envia também para a impressora.

Os métodos acima servem somente para imprimir em modo texto (letras, números e caracteres semi-gráficos).

7.3 Imprimindo via spool

A impressão via spool tem por objetivo liberar logo o programa do serviço de impressão deixando um outro programa específico tomar conta. Este programa é chamado de *daemon de impressão*, normalmente é o lpr ou o lprng (recomendado) em sistemas GNU/Linux.

Logo após receber o arquivo que será impresso, o programa de spool gera um arquivo temporário (normalmente localizado em /var/spool/lpd) que será colocado em fila para a impressão (um trabalho será impresso após o outro, em seqüência). O arquivo temporário gerado pelo programa de spool é apagado logo após concluir a impressão.

Antes de se imprimir qualquer coisa usando os daemons de impressão, é preciso configurar os parâmetros de sua impressora no arquivo /etc/printcap. Um arquivo /etc/printcap para uma impressora local padrão se parece com o seguinte:

```
lp|Impressora compatível com Linux
:lp=/dev/lp0
:sd=/var/spool/lpd/lp
:af=/var/log/lp-acct
:lf=/var/log/lp-errs
:pl#66
:pw#80
:pc#150
:mx#0
:sh
```

É possível também compartilhar a impressora para a impressão em sistemas remotos, isto será visto em uma seção separada neste guia.

Usando os exemplos anteriores da seção Imprimindo diretamente para uma porta de impressora, vamos acelerar as coisas:

- `cat trabalho.txt | lpr` - Direciona a saida do comando cat para o programa de spool lpr.

- `cat <trabalho.txt | lpr` - Faz a mesma coisa que o acima.

- `cat -n trabalho.txt | lpr` - Numera as linhas durante a impressão.

- `head -n 30 trabalho.txt | lpr` - Imprime as 30 linhas iniciais do arquivo.

A fila de impressão pode ser controlada com os comandos:

- `lpq` - Mostra os trabalhos de impressão atuais
- `lprm` - Remove um trabalho de impressão

Ou usando o programa de administração lpc para gerenciar a fila de impressão (veja a página de manual do lpc ou digite ? ao iniciar o programa para detalhes).

OBS1: Se a impressora não imprimir ou não for possível compartilhar a porta de impressora paralela com outros dispositivos (tal como o *plip*), verifique se o módulo *parport_pc* foi carregado e com os valores de irq e I/O corretos (por exemplo, `modprobe` parport_pc io=0x378 irq=7). Muitas vezes sua porta paralela pode funcionar sem problemas durante a impressão, mas se ao utilizar plip ocorrerem erros, a causa pode ser essa. Na distribuição Debian, use o programa modconf para configurar os valores permanentemente para o módulo parport_pc.

OBS2: Se tiver mais de uma impressora instalada na máquina, será necessário especificar a opção "-P impressora" para especificar qual impressora deseja imprimir/controlar.

7.4 Impressão em modo gráfico

A impressão em modo gráfico requer que conheça a marca e modelo de sua impressora e os métodos usados para imprimir seus documentos. Este guia abordará somente a segunda recomendação :-)

7.4.1 Ghost Script

O método mais usado pelos aplicativos do GNU/Linux para a impressão de gráficos é o *Ghost Script*. O Ghost Script (chamado de *gs*) é um interpretador do formato *PostScript* (arquivos .ps) e pode enviar o resultado de processamento tanto para a tela como impressora. Ele está disponível para diversas plataformas e sistema operacionais além do GNU/Linux, inclusive o DOS, Windows, OS/2, etc.

O formato .ps esta se tornando uma padronização para a impressão de gráficos em GNU/Linux devido a boa qualidade da impressão, liberdade de configuração, gerenciamento de impressão feito pelo *gs* e por ser um formato universal, compatíveis com outros sistemas operacionais.

Para imprimir um documento via Ghost Script, você precisará do pacote gs, gsfonts (para a distribuição Debian e distribuições baseadas, ou outros de acordo com sua distribuição Linux) e suas dependências. A distribuição Debian

vem com vários exemplos Pos Script no diretório /usr/share/doc/gs/example que são úteis para o aprendizado e testes com o Ghost Script.

Hora da diversão:

- Copie os arquivos tiger.ps.gz e alphabet.ps.gz do diretório /usr/share/doc/gs/examples (sistemas Debian) para /tmp e descompacte-os com os comandos
  ```
  # gzip -d tiger.ps.gz
  # gzip -d alphabet.ps.gz.
  ```

- O Ghost Script requer um monitor EGA, VGA ou superior para a visualização dos seus arquivos (não tenho certeza se ele funciona com monitores CGA ou Hércules Monocromático).
  Para visualizar os arquivos na tela digite:
  ```
  # gs tiger.ps
  # gs alphabet.os
  ```
  Para sair do Ghost Script pressione CTRL+C. Neste ponto você deve ter visto um desenho de um tigre e (talvez) letras do alfabeto. Se o comando gs alphabet.ps mostrou somente uma tela em branco, você se esqueceu de instalar as fontes do Ghost Script (estão localizadas no pacote gsfonts na distribuição Debian).

- Para imprimir o arquivo alphabet.ps use o comando:
  ```
  # gs -q -dSAFER -dNOPAUSE -sDEVICE=epson -r240x72 -sPAPERSIZE=legal -sOutputFile=/dev/lp0 alphabet.ps
  ```
  O arquivo alphabet.ps deve ser impresso. Caso aparecerem mensagens como Error: /invalidfont in findfont no lugar das letras, você se esqueceu de instalar ou configurar as fontes do Ghost Script. Instale o pacote de fontes (gsfonts na Debian) ou verifique a documentação sobre como configurar as fontes.
  Cada uma das opções acima descrevem o seguinte:

  - -q, -dQUIET - Não mostra mensagens de inicialização do Ghost Script.

  - -dSAFER - É uma opção para ambientes seguros, pois desativa a operação de mudança de nome e deleção de arquivo e permite somente a abertura dos arquivos no modo somente leitura.

  - -dNOPAUSE - Desativa a pausa no final de cada página processada.

  - -sDEVICE=dispositivo - Dispositivo que receberá a saida do Ghost Script. Neste local pode ser especificada a marca o modelo de sua impressora ou um formato de arquivo diferente (como pcxmono, bmp256) para que o arquivo .ps seja convertido para o formato designado.

    Para detalhes sobre os dispositivos disponíveis em seu Ghost Script, digite gs --help|less ou veja a página de manual.

Normalmente os nomes de impressoras e modelos são concatenados, por exemplo, bjc600 para a impressora *Canon BJC 600*, epson para impressoras padrão epson, stcolor para *Epson Stylus color*, etc.

O Hardware-HOWTO contém referências sobre hardware suportados pelo GNU/Linux, tal como impressoras e sua leitura pode ser útil.

- -r<ResH>x<ResV> - Define a resolução de impressão (em dpi) Horizontal e Vertical. Os valores dependem de sua impressora.

- -sPAPERSIZE=tamanho - Tamanho do papel. Podem ser usados a4, legal, letter, etc. Veja a página de manual do gs para ver os outros tipos suportados e suas medidas.

- -sOutputFile=dispositivo - Dispositivo que receberá a saida de processamento do gs. Você pode especificar

  - arquivo.epson - Nome do arquivo que receberá todo o resultado do processamento. O arquivo.epson terá toda a impressão codificada no formato entendido por impressoras epson e poderá ser impresso com o comando `cat arquivo.epson >/dev/lp0`.
    Uma curiosidade útil: É possível imprimir este arquivo em outros sistemas operacionais, tal como o DOS digitando: `copy /b arquivo.eps prn` (lembre-se que o DOS tem um limite de 8 letras no nome do arquivo e 3 na extensão. Você deve estar compreendendo a flexibilidade que o GNU/Linux e suas ferramentas permitem, isso é só o começo.

  - impressao%d.epson - Nome do arquivo que receberá o resultado do processamento. Cada página será gravada em arquivos separados como impressao1.epson, impressao2.epson, etc.
    Os arquivos podem ser impressos usando os mesmos métodos acima.

  - /dev/lp0 para uma impressora em /dev/lp0

  - para redirecionar a saída de processamento do gs para a saída padrão. É útil para usar o gs com pipes |.

  - \|lpr - Envia a saída do Ghost Script para o daemon de impressão. O objetivo é deixar a impressão mais rápida.

Se você é curioso, ou não está satisfeito com as opções mostradas acima, veja a página de manual do gs.

7.5 Magic Filter

O *Magic Filter* é um filtro de impressão inteligente. Ele funciona acionado pelo spool de impressão (mais especificamente o arquivo /etc/printcap) e permite identificar e imprimir arquivos de diversos tipos diretamente através do comando lpr arquivo.

É um ótimo programa e **ALTAMENTE RECOMENDADO** se você deseja apenas clicar no botão imprimir e deixar os programas fazerem o resto :-) A intenção do programa é justamente automatizar os trabalhos de impressão e spool.

A maioria dos programas para ambiente gráfico X11, incluindo o Netscape, Word Perfect, Gimp e Star Office trabalham nativamente com o magicfilter.

7.5.1 Instalação e configuração d o Magic Filter

O Magic Filter é encontrado no pacote magicfilter da distribuição Debian e baseadas.

Sua configuração pode ser feita com o programa magicfilterconfig que torna o processo de configuração rápido e fácil para quem não conhece a sintaxe do arquivo /etc/printcap ou não tem muitas exigências sobre a configuração detalhada da impressora.

Para testar o programa de configuração, mova o /etc/printcap para outro arquivo e chame o magicfilterconfig:

```
# mv /etc/printcap /etc/printcap.old
# magicfilterconfig
```

Após instalar o magicfilter reinicie o daemon de impressão (se estiver usando a Debian, entre no diretório /etc/init.d e como usuário root digite ./lpd restart).

```
# /etc/init.d/lpd restart
```

Para testar o funcionamento do magicfilter, digite lpr alphabet.ps e lpr tiger.ps, os arquivos serão enviados para o magicfilter que identificará o arquivo como *Pos Script*, executará o Ghost Script e retornará o resultado do processamento para o daemon de impressão. O resultado será visto na impressora.

Se tiver problemas, verifique se a configuração feita com o magicfilterconfig está correta. Caso precise re-configurar o magicfilger, digite magicfilterconfig --force (lembre-se que a opção --force substitui qualquer configuração personalizada que tenha adicionado ao arquivo /etc/printcap).

7.5.2 Outros detalhes técnicos sobre o Magic Filter

Durante a configuração do magicfilter, a seguinte linha é adicionada ao arquivo /etc/printcap:

```
:if=/etc/magicfilter/epson9-filter
```

A *epson* é citada apenas como exemplo. A linha que começa com :if no magicfilter identifica um arquivo de filtro de impressão.

O arquivo /etc/magicfilter/epcon9-filter é criado usando o formato do magicfilter, e não é difícil entender seu conteúdo e fazer algumas modificações:

```
#! /usr/sbin/magicfilter
#
# Magic filter setup file for 9-pin Epson (or compatible) printers
#
# This file is in the public domain.
#
# This file has been automatically adapted to your system.
#
# wild guess: native control codes start with ESC
0  \033  cat
# PostScript
0 %!        Filter      /usr/bin/gs -q -dSAFER -dNOPAUSE -r120x72 -
sDEVICE=epson -sOutputFile=- - -c quit
0 \004%!    filter      /usr/bin/gs -q -dSAFER -dNOPAUSE -r120x72 -
sDEVICE=epson -sOutputFile=- - -c quit
# PDF
0 %PDF      fpipe /usr/bin/gs -q -dSAFER -dNOPAUSE -r120x72 -
sDEVICE=epson -sOutputFile=- $FILE -c quit
# TeX DVI
0 \367\002  fpipe /usr/bin/dvips -X 120  -Y 72  -R -q –f
# compress'd data
0 \037\235  pipe  /bin/gzip  -cdq
# packed, gzipped, frozen and SCO LZH data
0 \037\036  pipe  /bin/gzip  -cdq
0 \037\213  pipe  /bin/gzip  -cdq
0 \037\236  pipe  /bin/gzip  -cdq
0 \037\240  pipe  /bin/gzip  -cdq
0 BZh pipe  /usr/bin/bzip2  -cdq
# troff documents
0 .\?\?\040 fpipe `/usr/bin/grog  -Tps $FILE`
0 .\\\"     fpipe `/usr/bin/grog  -Tps $FILE`
0 '\\\"     fpipe `/usr/bin/grog  -Tps $FILE`
0 '.\\\"    fpipe `/usr/bin/grog  -Tps $FILE`
0  \\\"     fpipe `/usr/bin/grog  -Tps $FILE`
```

Você deve ter notado que para cada tipo de arquivo existe o respectivo programa que é executado, basta você modificar as opções usadas nos programas neste arquivo (como faria na linha de comando) para afetar o comportamento da impressão.

Por exemplo, modificando a resolução para -r240x72 no processamento de arquivos PostScript (gs), a impressora passará a usar esta resolução.

8 LIMITANDO O USO DE ESPAÇO EM DISCO (QUOTAS)

O sistema de quotas é usado para limitar o espaço em disco disponível a usuários/grupo. O uso de partições independentes para o diretório /home e outros montados separadamente não é muito eficaz porque muitos usuários serão prejudicados se a partição for totalmente ocupada e alguns possuem requerimentos de uso maior do que outros.

O suporte a *Quotas* deve estar compilado no kernel (seção *FileSystems*) e o sistema de arquivos deverá ser do tipo *ext2* ou *XFS* para funcionar.

```
Filesystems --->
      [  ] Quota support
```

8.1 Instalando o sistema de quotas

Abaixo o passo a passo para a instalação de quotas em seu sistema:

1. Recompile seu kernel com suporte a quota. Habilite a opção "Quota support" na seção "FileSystems" na configuração de recursos do seu kernel.

2. Instale o pacote quota no sistema (apt-get install quota, ou usando o dselect).

3. Habilite a quota para os sistemas de arquivos que deseja restringir no arquivo /etc/fstab:

   ```
   /dev/hda1    /boot ext2    defaults                  1  1
   /dev/hda3    /     ext2    defaults,usrquota         1  2
   /dev/hda4    /usr  ext2    defaults,grpquota         1  3
   /dev/hda5    /pub  ext2    defaults,usrquota,grpquota 1  4
   ```

   No exemplo acima, o sistema de arquivos /dev/hda1 não terá suporte a quota, /dev/hda3 terá suporte a quotas de usuários (*usrquota*), /dev/hda4 terá suporte a quotas de grupos (*grpquota*) e /dev/hda5 terá suporte a ambos. Por padrão é assumido que os arquivos de controle de quota estão localizados no ponto de montagem da partição com os nomes quota.user e quota.group.
   Em nosso exemplo, vamos criar quotas de usuário e grupos no /dev/hda3, como abaixo. Edite o fstab e confira a existência dos parâmetros usrquota e grpquota.

   ```
   /dev/hda3    /     ext2    defaults,errors=remount -
   ro,usrquota,grpquota          0  1
   ```

4. Agora será necessário criar os arquivos quota.user e quota.group no ponto de montagem de cada partição *ext2* acima que utilizará o

recurso de quotas. O arquivo quota.user controla as quotas de usuários e quota.group controla as quotas de grupos.

5. Crie um arquivo vazio quota.user em / (terá suporte somente a quota de usuários, veja a opção de montagem no /etc/fstab):
   ```
   # touch /quota.user
   ```

6. Crie um arquivo vazio quota.group em /usr (terá suporte somente a quota de grupos):
   ```
   # touch /quota.group
   ```
   Por motivos de segurança, as permissões dos arquivos de controle de quota quota.user e quota.group devem ser leitura/gravação ao usuário root e sem permissões para grupo/outros usuários: chmod 0600 /quota.user /quota.group.

7. Entre em modo monousuário init 1, desmonte os sistemas de arquivos que utilizarão a quota e monte-os novamente (isto serve para ativar as opções de quota). Alternativamente, execute umount -a (para desmontar todos os sistemas de arquivos) e mount -a para remontar todos.
   Se você ativou as quotas para o sistema de arquivos / (como em nosso exemplo) será necessário reiniciar o sistema.

8. O próximo passo é scanear o disco para criar os dados para as partições com suporte a quota (ativadas no /etc/fstab):
   ```
   # quotacheck –augv
   ```
   O parâmetro *-a* diz para checar todas as partições com suporte a quota no arquivo /etc/mtab, *-u* para checar quotas de usuários, *-g* para checar grupos e *-v* para mostrar o progresso da checagem da partição.
   Na primeira execução é mostrado uma mensagem de erro de arquivo quota.user/quota.group corrompido, mas isto é normal porque o arquivo anterior tem tamanho zero. Estes nomes também servem para o quotacheck "auto-detectar" a versão do sistema de quota usada no sistema de arquivos.
   OBS: Certamente será necessário "forçar" a remontagem como somente leitura do sistema de arquivos / com a opção *-m* para o quotacheck criar as configurações de quota nesta partição.

9. Agora resta ativar o suporte as quotas de disco em todas as partições (*-a*) com recurso de quota especificado (no /etc/mtab):
   ```
   # quotaon –augv
   ```
   As opções possuem o mesmo significado do comando quotacheck. O utilitário quotaoff serve para desativar quotas de usuários e usa as mesmas opções do quotaon. Estes três utilitários somente podem ser usados pelo usuário root. As opções de quota podem ser especificadas independente para cada sistema de arquivos:

   - Ativa o suporte a quota em /pub (somente grupos de usuários no momento).
     ```
     # quotaon -gv /pub
     ```

- Ativa as quotas de usuários em /pub
  ```
  # quotaon -uv /pub
  ```

- Desativa as quotas de grupos em /pub (deixando somente a de usuários ativa)
  ```
  # quotaoff -gv /pub
  ```

A atualização de quotas durante a gravação/exclusão de arquivos é feita automaticamente. O utilitário quotacheck deverá ser executado sempre que o sistema de quotas for desativado (por não haver atualização automática dos dados de uso de disco) ou quando ocorrerem falhas de disco.

Na distribuição Debian o quotacheck é disparado sempre que necessário após as situações de checagem de disco. As quotas de todas as partições também são ativadas automaticamente pelo script /etc/init.d/quota e /etc/init.d/quotarpc.

Em sistemas que utilizam NFS e possuem sistemas de arquivos exportados em /etc/exports, o daemon rpc.rquotad deverá ser carregado. Sua função é fornecer os detalhes de quota dos sistemas de arquivos locais exportados para as máquinas clientes.

8.2 Editando quotas de usuários/grupos

O programa edquota é usado pelo root para editar as quotas de usuários/grupos. Por padrão, todos os usuários/grupos do sistema não possuem quotas. Sua sintaxe é a seguinte:

`edquota [`*`opções`*`] [`*`usuário/grupo`*`]`

As opções podem ser:

- -u - Edita a quota do usuário especificado (esta é a padrão).
- -g - Edita a quota de grupo especificado.
- -r - Permite editar a quota de sistemas de arquivos remotos através do daemon rpc.rquotad.
- -t - Permite modificar o valor de tolerância dos limites que ultrapassam *soft* até que sejam bloqueados. Durante o tempo de tolerância, serão enviados somente avisos sobre a quota ultrapassada sem bloquear totalmente a gravação de arquivos (até que o limite *hard* seja atingido ou o tempo de tolerância seja ultrapassado).

Quando a quota *soft* do usuário/grupo é estourada, a mensagem "warning: user disk quota excedeed" será exibida. Quando a quota *hard* é ultrapassada, a gravação atual é interrompida e a mensagem "write failed, user disk limit reatched" é mostrada ao usuário. Nenhuma nova gravação que

ultrapasse a quota *hard* é permitida Por exemplo, para modificar a quota do usuário gleydson: edquota gleydson

```
Disk quotas for user gleydson (uid 1000):
  Filesystem  blocks  soft    hard    inodes   soft    hard
  /dev/hda5   504944  500100  600000  10868    15000    20000
```

O editor de textos usado poderá ser modificado através da variável $EDITOR. Abaixo a explicação destes campos:

- Filesystem - Sistema de arquivos que terá a quota do usuário/grupo editada. As restrições se aplicam individualmente de acordo com o sistema de arquivos.

- blocks - Número máximo de blocos (especificado em Kbytes) que o usuário possui atualmente. O usuário gleydson está usando atualmente 504944 Kbytes.

  - soft - Restrição mínima de espaço em disco usado. Atualmente 500100 Kb.

  - hard - Limite máximo aceitável de uso em disco para o usuário/grupo sendo editado. 600000 Kb atualmente. O sistema de quotas nunca deixará este limite ser ultrapassado.

- inodes - Número máximo de arquivos que o usuário possui atualmente na partição especificada. O usuário gleydson possui atualmente 10868 arquivos na partição /pub.

  - soft - Restrição mínima de número de arquivos que o usuário/grupo possui no disco. Atualmente em 15.000.

  - hard - Restrição máxima de número de arquivos que o usuário/grupo possui no disco. Atualmente em 20.000.

Para desativar as restrições coloque "0" no campo *soft* ou *hard*. Quando o limite *soft* é atingido, o usuário é alertado por ter ultrapassado sua quota com a mensagem "warning: user quota excedeed" (quota do usuário excedida). O programa setquota é uma programa não-interativo para edição de quotas para ser usado diretamente na linha de comando ou em shell scripts.

Após ultrapassar o limite *soft*, começa a contagem do tempo para que este passe a valer como limite *hard* (o máximo aceitável e que nunca poderá ser ultrapassado). O comando edquota -t serve para modificar estes valores na partição especificada:

```
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
Filesystem             Block grace period     Inode grace period
/dev/hda5                   2days                  7days
```

Abaixo a explicação destes campos:

- Filesystem - Sistema de arquivos que terá o período de tolerância modificado.

- Block grade period - Tempo máximo de tolerância para usuários/grupos que ultrapassaram sua quota *soft* de espaço em disco antes de passar a valer como *hard*. No exemplo, o usuário tem 2 dias para excluir possíveis arquivos ou contactar o administrador para redimensionar o tamanho de quota. O valor padrão é 7 dias.

- Inode grade period - Tempo máximo de tolerância para usuários/grupos que ultrapassaram sua quota *soft* de número de arquivos gravados antes de passar a valer como *hard*. No exemplo, o usuário tem 7 dias para excluir possíveis arquivos ou contactar o administrador para analisar seu tamanho de quota. O valor padrão é 7 dias.

OBS1: - O comando quotacheck deverá ser executado na partição sempre que novas restrições/limites forem editados com o edquota. Isto atualiza os arquivos quota.user e quota.group. Lembre-se de desativar o sistema de quotas (quotaoff -ugv /partição) antes de executar este comando (para liberar totalmente a partição, quotacheck remonta a partição somente para leitura quando é executado). Por este motivo é recomendável fazer isso em modo monousuário.

OBS2: Quando o limite *soft* (suave) é excedido, o sistema começará a lhe mostrar mensagens alertando a passagem do limite (para lhe dar tempo de eliminar arquivos ou não ser pego desprevenido com o bloqueio de gravação) porque o limite *hard* (rígido) nunca poderá ser ultrapassado.

OBS3: - O tempo de tolerância restante ao usuário/grupo quando a quota é ultrapassada poder ser visualizada com o comando quota.

OBS4: - Quando o usuário exclui seus arquivos e volta a ficar abaixo dos limites *soft* da quota, o tempo de tolerância é resetado aos valores padrões (especificados por edquota -t.

OBS5: - As quotas de espaço em disco podem ser definidas automaticamente para os novos usuários adicionados ao sistema colocando o espaço em disco na variável QUOTAUSER=numero do arquivo /etc/adduser.conf. Isto será equivalente a digitar o comando edquota -q QUOTA novo_usuário.

Exercício:

1.Adicione um usuário (fulano) através do comando adduser

```
# adduser
```

2.Defina uma quota de blocks soft de 70 KB e hard de 100 KB; e uma quota de inodes soft de 10 e hard de 30 para esse usuário com o comando edquota,

```
# edquota fulano
```

```
/dev/hda3: blocks in use: 4, limits (soft = 70, hard = 100)
           inodes in use: 1,  limits (soft = 10, hard = 30)
```

3.Como root, desative as quotas, com o comando quotaoff

```
# quotaoff -avug
```

4.Edite os limites de tempo para os block e file grace period, com o comando edquota

```
# edquota -t

/dev/hda3: block grace period: 10 minutes, file grace period: 5
minutes
```

OBS: o default é 7 dias (7 days)

5.Cheque as quotas novamente para garantir que o novo grace period seja ativado

```
# quotacheck -avug
```

6.Reative as quotas:

```
# quotaon -avug
```

7.Entre como usuário “fulano” usando o comando su

```
# su – fulano
```

8.Crie mais de 10 arquivos, um depois do outro, como abaixo:

```
# > teste1
# > teste2
...
# > teste10
```

Você vai ver a mensagem de warning sobre o excesso de quota quando ultrapassar o número de arquivos permitidos. Nesse momento, chame:

```
# quota fulano
```

E verá que ultrapassou o soft limit do número de arquivos (inodes). Além disso, verá o tempo restante no “grace period”.

9.Para testar a quota de máximo de blocos escritos, escolha algum arquivo grande para copiar (por exemplo do /sbin) para o seu diretório. Após copiar o arquivo, verifique a quota com o comando “quota fulano”. Tente copiar mais alguns arquivos grandes até chegar à quota hard, quando você verá a mensagem que a quota excedeu e você não pode mais gravar no disco.

8.3 Verificando a quota disponível ao usuário

O comando quota mostra os limites de usuários/grupos e a tolerância restante antes do limite *soft* se tornar rígido. Abaixo alguns exemplos descritivos deste comando:

```
# quota
Disk quotas for user gleydson (uid 1234):
Filesystem blocks  quota   limit   grace  files  quota   limit   grace
/dev/hda5  504944* 500100  600000  00:05  10868  0       0
```

Os campos tem o seguinte significado:

- Filesystem - Sistema de arquivos.
- blocks - Número de blocos usados atualmente na partição (em Kb). O "*" indica que o limite foi ultrapassado. Atualmente em 504944.
    - quota - Limite suave (*soft*) de espaço na partição que o usuário/grupo possui. Atualmente 500100. O valor 0 indica que o usuário/grupo não possui restrições.
    - limit - Limite máximo (*hard*) de espaço na partição que o usuário/grupo possui. Atualmente em 600000. O valor 0 indica que o usuário/grupo não possui restrições.
    - grace - Tolerância antes que o limite *soft* passe a valer como *hard* quando o espaço em disco é ultrapassado. Este usuário tem 5 minutos restantes para que isto ocorra. Quando o valor *soft* volta a ficar abaixo da quota, a tolerância é resetada.

      O parâmetro "none" indica que o tempo de tolerância expirou (caso existam limitações de quota que foram ultrapassadas) ou que o usuário/grupo não possui restrições. Veja se existe um "*" no campo blocks.
- files - Número máximo de arquivos que usuário/grupo possui atualmente na partição. Um "*' indica que o limite foi ultrapassado. Atualmente em 10868.
    - quota - Limite suave (*soft*) de número de arquivos na partição que o usuário/grupo possui. Atualmente ilimitado.
    - limit - Limite máximo (*hard*) de número de arquivos na partição que o usuário/grupo possui. Atualmente ilimitado.
    - grace - Tolerância antes que o limite *soft* passe a valer como *hard* para o número de arquivos ultrapassados. Como não existe quota para número de arquivos, não existe tolerância. A tolerância é resetada aos valores padrões quando o valor *soft* volta a ficar abaixo da quota.

As quotas de outros usuários/grupos podem ser visualizadas especificando as opções *-u* (padrão) e *-g* na linha de comando respectivamente. A opção *-v* permite visualizar quotas em sistemas de arquivos não alocados e *-q* mostra somente uma mensagem dizendo se o usuário está ou não dentro de sua quota:

```
# quota -u usuario
# quota -uq usuario
# quota -g users
```

Por motivos de segurança, você não poderá visualizar as quotas de outros usuários e grupos a que não pertence (exceto para o usuário root).

8.4 Verificando a quota de todos os usuários/grupos do sistema

Quando precisamos verificar o uso de quotas de todos os usuários/grupos do sistema o quota se torna incômodo e pouco prático. O comando repquota está disponível ao administrador para facilitar esta tarefa. Sua listagem é organizada por partições listando dados adicionais como grace time e aceita as mesmas opções dos utilitários quotaon e quotaoff. Primeiro são listados as restrições de usuários e depois de grupos para a partição. (tolerância) As opções aceitas por este utilitário tem o mesmo significado das opções do quotaon e quotaoff:

```
# repquota -aug

*** Report for user quotas on device /dev/hda3
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --   29160       0       0   none    9970     0     0   none
daemon    --      64       0       0              22     0     0
man       --     944       0       0              65     0     0
mail      --    4960       0       0             823     0     0
news      --       4       0       0               1     0     0
gleydson  --   31032       0       0            6956     0     0
testuser  --      16       0       0               4     0     0
anotheruser --    16       0       0               4     0     0
nobody    --    2344       0       0               2     0     0

*** Report for user quotas on device /dev/hda5
Block grace time: 2days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --   16052       0       0   none    6443     0     0   none
gleydson  +-    4944  500100  600000   none   10868     0     0

*** Report for group quotas on device /dev/hda5
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
Group           used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
```

```
root      --   20308       0       0   none     636     0     0   none
src       --   11404       0       0            660     0     0
users     --    1756       0       0           6561     0     0
gleydson  --    3452       0       0           9307     0     0
```

Um sinal de "+-" no segundo campo indica quota ultrapassada ou no espaço em disco, "-+' em número de arquivos e "++" em ambos. Como vimos acima, o este comando também lista o número de arquivos e bytes pertencentes a cada usuário na partição (mesmo não sendo monitorado pelas restrições de quota), isto ajuda a monitorar ações suspeitas com a excedência de espaço em disco de determinados usuários/grupos do sistema. Um exemplo é alguém que esteja fora da quota e abusando de seu usuário/grupo para uso excessivo de espaço em disco sem seu conhecimento.

OBS: Este utilitário pode ser executado por qualquer usuário no sistema e mostrar o uso de quotas de usuários/grupos que não deveria ter acesso. É recomendado deve ter permissões de leitura/gravação somente para o usuário root e sem permissões para grupo/outros usuários.

9 SCRIPTS DE SHELL

Digamos que você use uma série de comandos freqüentemente e gostaria de economizar tempo agrupando todos juntos num único "comando". Por exemplo, nos três comandos

```
[aluno@gauss ~]$  cat  chapter1 chapter2 chapter3  > book

[aluno@gauss ~]$  wc  -l bool

[aluno@gauss ~]$  lp  book
```

O primeiro (cat) concatena os arquivos chapter1, chapter2 e chapter3 e coloca o resultado no arquivo "book". O segundo comando mostra a contagem do número de linhas em "book", e o terceiro comando (lp) imprime o arquivo "book".

Ao invés de digitar todos os três comandos, você pode agrupá-los em um **script shell**. O script shell usado para rodar todos esses três comandos, pode ser parecido com:

```
#!/bin/sh

# Um script shell para criar e imprimir o livro

cat chapter1 chapter2 chapter3 > book

wc -l book

lp book
```

Scripts de shell são arquivos texto comuns. Você pode criá-los num editor, como o Emacs ou vi.

Vamos dar uma olhada neste script shell. A primeira linha, "#!/bin/sh", identifica o arquivo como um script de shell e diz ao shell como executar o script. Ela instrui o shell a passar o script para o /bin/sh para execução, onde /bin/sh é o próprio programa do shell. Por que isto é importante? Na maioria dos sistemas Linux, /bin/sh é um shell tipo Bourne shell, como o bash. Forçando o script de shell a rodar com /bin/sh, você assegura que o script rodará com shell com sintaxe Bourne (ao invés do C shell). Isso vai fazer com que seu script rode com a sintaxe do Bourne shell mesmo que você use o tcsh (ou outro shell C).

A segunda linha é um comentário. Comentários começam com o caracter "#", e continuam até o fim da linha. Comentários são ignorados pelo shell - eles são normalmente usados para identificar um script shell para o programador e fazer com que o script seja mais fácil de entender.

O resto das linhas no script são somente comandos, como se você os tivesse digitado no shell diretamente, Em efeito, o shell lê cada linha do script e roda essa linha como se você a tivesse digitado no prompt. Digamos que você

tenha gravado o script acima com o nome de makebook. Para rodá-lo, use o seguinte comando:

```
[aluno@gauss ~]$ bash makebook
```

Permissões são importantes para scripts shell. Se você cria um script shell, assegure-se que tenha permissão de execução no script para poder rodá-lo. Quando você cria arquivos texto, as permissões padrão usualmente não incluem permissão de execução, e você deve especificá-las explicitamente. Brevemente, se o script foi salvo no arquivo chamado "makebook", você pode usar o seguinte comando para dar a você mesmo permissão de execução para o script de shell "makebook":

```
[aluno@gauss ~]$  chmod u+x makebook
```

Agora você pode usar o seguinte comando para rodar todos os comandos no script:

```
[aluno@gauss ~]$ ./makebook
```

Vejamos um outro exemplo: um script que mostra o nome do usuário logado e as horas:

```
#!/bin/sh

echo “O nome do usuário é `whoami`”

echo “Agora são `date +%H:%M`”
```

O resultado da execução do script acima é o seguinte:

```
[aluno@gauss ~]$ ./quemsou

O nome do usuário é aluno

Agora são 18:45
```

9.1 Passagem de Parâmetros para o Scipt

Uma característica interessante dos scripts é que podemos pssar parâmetros pela linha de comando. Esses parâmetros podem ser posteriormente usados pelo programa.

Vamos ver como isso funciona na prática. Primeiro, vamos criar um arquivo que vai nos servir de “banco de dados”:

```
Joao        555-1111    joao@alfamidia.com.br
Jose        555-2222    jose@linux.org
Maria       555-3333    maria@kernel.org
Pedro       555-4444    pedro@alfamidia.com.br
Luis        555-5555    luis@redhat.com
```

Podemos criar este arquivo com o vi, e salvá-lo com o nome de agenda.dat.

Agora vamos fazer um script para procurar um nome no banco de dados:

```
#!/bin/sh

grep –i $1 agenda.dat
```

Agora podemos executar o script com o comando:

```
[aluno@gauss ~]./agenda Pedro

Pedro       555-4444    pedro@alfamidia.com.br
```

Pode-se perceber que a variável $1, utilizada dentro do script, é na verdade o primeiro parâmetro passado. O mesmo vale para $2, que é o segundo parâmetro, e assim por diante. A variável $0 contém o nome do script, e a variável $@ contém todos os parâmetros.

9.2 Leitura do teclado

Podemos alterar o exemplo anterior para que o valor seja lido do teclado ao invés de ser passado com parâmetro. Para isso vamos usar o comando `read`.

```
#!/bin/sh

echo –n “Nome a procurar: “

read NOME

grep –i $NOME agenda.dat
```

Note que a variável não precisa ser declarada anteriormente.

9.3 Construções Condicionais

Construções condicionais fazem com que determinado código seja executado dependendo do resultado de uma expressão.

9.3.1 Expressões Inteiras

Expressões inteiras utilizam os operadores de testes para dados inteiros.

Tabela 9-1Expressões Inteiras

Expressão	Verdadeiro se:

x –eq y	x == y
x –ne y	x != y
x -ge y	x >= y
x -gt y	x > y
x -le y	x <= y
x -lt y	x < y

9.3.2 Expressões de Strings

Expressões de strings servem para fazer vários testes com strings.

Tabela 9-2 Expressões de Strings

Expressão	**Verdadeiro se:**
`str`	`str` não é nula
`-z str`	`str` tem tamanho zero
`-n str`	`str` não tem tamanho zero
`str1 = str2`	`str1` é igual a `str2`
`str1 != str2`	`Str1` é diferente de str2

Expressões de Arquivo

Expressões de arquivo servem para fazer vários tipos de testes com arquivos.

Tabela 9-3 Expressões de Arquivos

Expressão	Verdadeiro se:
-e arq	arq existe
-f arq	arq é um arquivo normal
-d arq	arq é um diretório
-r arq	arq pode ser lido
-w arq	arq pode ser gravado
arq1 –nt arq2	arq1 é mais novo que arq2
arq1 –ot arq2	arq1 é mais velho que arq2

9.3.3 Expressões Lógicas

Expressões lógicas servem para combinar o resultado de outras expresões.

Tabela 9-4 Expressões Lógicas

Expressão	Verdadeiro se:
`!exp`	NOT `exp`
`exp1 && exp2`	`exp1` AND `exp2`
`exp1 –a exp2`	`exp1` AND `exp2`
`exp1 || exp2`	`exp1` OR `exp2`
`exp1 –o exp2`	`exp1` OR `exp2`

9.3.4 Declaração `if`

A declaração `if` executa determinadas linhas, dependendo se uma expressão for verdadeira ou falsa. O formato da declaração é o seguinte:

```
if exp; then
    comandos;
elif exp; then
    comandos;
elif exp; then
```

```
    comandos;
else
    comandos;
fi
```

Note que a declaração `else` é opcional, e a declaração `elif` pode aparecer qualquer número de vezes (inclusive zero).

9.3.5 Declaração `case`

A declaração `case` executa comandos dependendo do valor de uma variável.

```
case $var in
    const1 | const2) comando; comando;;
    const3 | const4) comando; comando;;
    *) comando; comando;;
esac
```

Note que podem haver várias constantes em cada opção. A opção *) é escolhida se nenhuma das anteriores for verdadeira.

9.4 Controle de Laços

9.4.1 Declaração `while`

A declaração while é usada para repetir comandos até que uma expressão seja avaliada como falsa.

```
while exp
do
    comandos
done
```

9.4.2 Declaração `for`

A declaração for é usada basicamente para percorrer listas.

```
for var in [ word1 word2 word3 ]
do
```

```
    comando;
done
```

10 MANUTENÇÃO DO SISTEMA

Este capítulo descreve como fazer a manutenção de seu sistema de arquivos e os programas de manutenção automática que são executados periodicamente pelo sistema.

10.1 Checagem dos sistemas de arquivos

A checagem do sistema de arquivos permite verificar se toda a estrutura para armazenamento de arquivos, diretórios, permissões, conectividade e superfície do disco estão funcionando corretamente. Caso algum problema exista, ele poderá ser corrigido com o uso da ferramenta de checagem apropriada. As ferramentas de checagem de sistemas de arquivos costumam ter seu nome iniciado por fsck e terminados com o nome do sistema de arquivos que verifica, separados por um ponto:

- `fsck.ext2` - Verifica o sistema de arquivos EXT2. Pode também ser encontrado com o nome `e2fsck`.
- `fsck.minix` - Verifica o sistema de arquivos Minix.
- `fsck.msdos` - Verifica o sistema de arquivos Msdos. Pode também ser encontrado com o nome `dosfsck`.

Para verificar um sistema de arquivos é necessário que ele esteja desmontado caso contrário poderá ocorrer danos em sua estrutura. Para verificar o sistema de arquivos raiz (que não pode ser desmontado enquanto o sistema estiver sendo executado) você precisará inicializar através de um disquete e executar o `fsck.ext2`.

10.1.1 `fsck.ext2`

Este utilitário permite verificar erros em sistemas de arquivos EXT2 (*Linux Native*).

`fsck.ext2 [`*`opções`*`] [`*`dispositivo`*`]`

Onde:

- `dispositivo`: É o local que contém o sistema de arquivos EXT2 que será verificado (partições, disquetes, arquivos).
- `opções`:
  - **-d** : Debug - Mostra detalhes de processamento do fsck.ext2.

- **-c** : Faz o fsck.ext2 verificar se existem agrupamentos danificados na unidade de disco durante a checagem.

- **-f** : Força a checagem mesmo se o sistema de arquivos aparenta estar em bom estado. Por padrão, um sistema de arquivos que aparentar estar em bom estado não são verificados.

- **-F** : Grava os dados do cache no disco antes de iniciar.

- **-l [arquivo]** : Inclui os blocos listados no [arquivo] como blocos defeituosos no sistema de arquivos. O formato deste arquivo é o mesmo gerado pelo programa badblocks.

- **-L [arquivo]** : Faz o mesmo que a opção -l, só que a lista de blocos defeituosos do dispositivo é completamente limpa e depois a lista do [arquivo] é adicionada.

- **-n** : Faz uma verificação de somente leitura no sistema de arquivos. Com esta opção é possível verificar o sistema de arquivos montado. Será assumido não para todas as perguntas e nenhuma modificação será feita no sistema de arquivos.

- **-p** : Corrige automaticamente o sistema de arquivos sem perguntar. É recomendável fazer isto manualmente para entender o que aconteceu, em caso de problemas com o sistema de arquivos.

- **-v** : Ativa o modo verbose (mais mensagens são mostradas durante a execução do programa).

- **-y** : Assume sim para todas as questões.

Obs: Caso a opção -c seja usada junto com -n, -l ou -L, o sistema de arquivos será verificado e permitirá somente a atualização dos setores danificados não alterando qualquer outra área.

Caso sejam encontrados arquivos problemáticos e estes não possam ser recuperados, o fsck.ext2 perguntará se deseja salvá-los no diretório lost+found. Este diretório é encontrado em todas as partições *ext2*.

Após sua execução são mostrados detalhes sobre o sistema de arquivos verificado como quantidade de blocos livres/ocupados e taxa de fragmentação.

Exemplos:

```
# fsck.ext2 /dev/hda2

# fsck.ext2 -f /dev/hda2

# fsck.ext2 -vrf /dev/hda1.
```

Exercício:

Como root, entre no modo monousuário:

```
# init 1
```

Desmonte o sistema de arquivos que vai checar (em nosso exemplo, o /dev/hda3)

```
# umount /dev/hda3
```

Chame o fsck.ext2 com a opção -f, para forçar a checagem

```
# fsck.ext2 -f /dev/hda3
```

Pressione Ctrl-Alt-Del para reiniciar o sistema

10.1.2 badblocks

Procura blocos defeituosos em um dispositivo.

`badblocks [`*`opções`*`] `*`dispositivo blocos [bloco inicial]`*

Onde:

- opções
  - -b [tamanho] : Especifica o [tamanho] do bloco do dispositivo em bytes
  - -o [arquivo] : Gera uma lista dos blocos defeituosos do disco no [arquivo]. Este lista pode ser usada com o programa fsck.ext2 junto com a opção -l.
  - -s : Mostra o número de blocos checados durante a execução do badblocks.
  - -v : Modo verbose - São mostrados mais detalhes.
  - -w : Usa o modo leitura/gravação. Usando esta opção o badblocks procura por blocos defeituosos gravando alguns padrões (0xaa, 0x55, 0xff, 0x00) em cada bloco do dispositivo e comparando seu conteúdo. Nunca use a opção -w em um dispositivo que contém arquivos pois eles serão apagados!
- dispositivo - Partição, disquete ou arquivo que contém o sistema de arquivos que será verificado.
- blocos - É o número de blocos do dispositivo. Você pode ver isso com o comando “df”.
- bloco inicial - É o bloco de onde você quer começar a checagem

Exemplo: badblocks -sv /dev/hda3 3200460

10.2 Limpando arquivos de LOGS

Tudo que acontece em sistemas GNU/Linux pode ser registrado em arquivos de log em /var/log, como vimos anteriormente. Eles são muito úteis por diversos motivos, para o diagnóstico de problemas, falhas de dispositivos, checagem da segurança, alerta de eventuais tentativas de invasão, etc.

O problema é quando eles começam a ocupar muito espaço em seu disco. Verifique quantos Megabytes seus arquivos de LOG estão ocupando através dos comandos:

```
# cd /var/log
# du -hc
```

Antes de fazer uma limpeza nos arquivos de LOG, é necessário verificar se eles são desnecessários e só assim zerar os que forem dispensáveis.

Não é recomendável apagar um arquivo de log pois ele pode ser criado com permissões de acesso indevidas (algumas distribuições fazem isso). Você pode usar o comando: “echo -n >arquivo” ou o seguinte shell script para zerar todos os arquivos de LOG de uma só vez (as linhas iniciante com # são comentários):

```
#!/bin/sh
cd /var/log
for l in `ls -p|grep '/'`; do
  echo -n >$l &>/dev/null
  echo Zerando arquivo $l...
done
echo Limpeza dos arquivos de log concluída!
```

Copie o conteúdo acima em um arquivo com a extensão .sh, dê permissão de execução com o chmod e o execute como usuário root. É necessário executar este script para zerar arquivos de log em subdiretórios de /var/log, caso sejam usados em seu sistema.

Algumas distribuições, como a Debian GNU/Linux, fazem o arquivamento automático de arquivos de LOGs em arquivos .gz através de scripts disparados automaticamente pelo cron. ATENÇÃO: LEMBRE-SE QUE O SCRIPT ACIMA APAGARÁ TODOS OS ARQUIVOS DE LOGs DO SEU SISTEMA SEM POSSIBILIDADE DE RECUPERAÇÃO. TENHA ABSOLUTA CERTEZA DO QUE NÃO PRECISARÁ DELES QUANDO EXECUTAR O SCRIPT ACIMA!

10.3 Tarefas automáticas de manutenção do sistema

Os arquivos responsáveis pela manutenção automática do sistema se encontram em arquivos individuais localizados nos diretórios /etc/cron.daily,

/etc/cron.weekly e /etc/cron.montly. A quantidade de arquivos depende da quantidade de pacotes instalado em seu sistema, porque alguns programam tarefas nestes diretórios e não é possível descrever todas, para detalhes sobre o que cada arquivo faz veja o cabeçalho e o código de cada arquivo.

Estes arquivos são executados pelo cron através do arquivo /etc/crontab. Você pode programar quantas tarefas desejar, para detalhes veja a próxima seção (cron). Alguns programas mantêm arquivos do cron individuais em /var/spool/cron/crontabs que executam comandos periodicamente.

10.4 cron

O cron é um daemon que permite o agendamento da execução de um comando/programa para um determinado dia/mes/ano/hora. É muito usado em tarefas de arquivamento de logs, checagem da integridade do sistema e execução de programas/comandos em horários determinados.

As tarefas são definidas no arquivo /etc/crontab e por arquivos individuais de usuários em /var/spool/cron/crontabs/[usuário] (criados através do programa crontab). Adicionalmente a distribuição Debian utiliza os arquivos no diretório /etc/cron.d como uma extensão para o /etc/crontab.

Para agendar uma nova tarefa, basta editar o arquivo /etc/crontab com qualquer editor de texto (como o ae e o vi) e definir o mes/dia/hora que a tarefa será executada. Não é necessário reiniciar o daemon do cron porque ele verifica seus arquivos a cada minuto. Veja a seção “O formato de um arquivo crontab” para entender o formato de arquivo cron usado no agendamento de tarefas.

10.4.1 O formato de um arquivo crontab

O arquivo /etc/crontab tem o seguinte formato:

```
52 18  1  *  *  root  run-parts --report /etc/cron.montly
 |  |  |  |  |    |    |
 |  |  |  |  |    |    \_Comando que será executado
 |  |  |  |  |    \_ UID que executará o comando
 |  |  |  |  \_ Dia da semana (0-7)
 |  |  |  \_ Mes (0-11)
 |  |  \_ Dia do Mês (1-31)
 |  \_ Hora (0-23)
 \_ Minuto (0-59)
```

Onde:

- Minuto: Valor entre 0 e 59
- Hora: Valor entre 0 e 23

- Dia do Mês: Valor entre 1 e 31
- Mês: Valor entre 0 e 11 (Janeiro=0, Dezembro=11)
- Dia da Semana: Valor entre 0 e 7 (Domingo a Sábado). Note que tanto 0 e 7 equivalem a Domingo.
- Usuário: O usuário especificado será usado para executar o comando (o usuário deverá existir).
- Comando: Comando que será executado. Podem ser usados parâmetros normais usados na linha de comando.

Os campos do arquivo são separados por um ou mais espaços ou tabulações. Um asterisco * pode ser usado nos campos de data e hora para especificar todo o intervalo disponível. O hífen - serve para especificar períodos de execução (incluindo a o número inicial/final). A vírgula serve para especificar lista de números. Passos podem ser especificados através de uma /. Veja os exemplos no final desta seção.

O arquivo gerado em /var/spool/cron/crontabs/[usuário] pelo crontab tem o mesmo formato do /etc/crontab exceto por não possuir o campo usuário (UID), pois o nome do arquivo já identifica o usuário no sistema.

Para editar um arquivo de usuário em /var/spool/cron/crontabs ao invés de editar o /etc/crontab use crontab -e, para listar as tarefas daquele usuário crontab -l e para apagar o arquivo de tarefas do usuário crontab -r (adicionalmente você pode remover somente uma tarefa através do crontab -e e apagando a linha correspondente).

OBS: Não esqueça de incluir uma linha em branco no final do arquivo, caso contrário o último comando não será executado.

Exercício:

1.Crie uma conta de usuário (vamos usar fulano, como exemplo). Abra um terminal com essa conta (tty1) e outro (tty2) com a conta root.

2.Com a conta root, apague qualquer arquivo que esteja em /var/spool/cron/crontabs

```
# rm /var/spool/cron/crontabs/*
```

3.Alterne para o outro terminal (tty1) e edite o arquivo de tarefas do usuário, como abaixo:

```
$ crontab -e

# cria o arquivo /tmp/teste1 com a palavra OLA às 15: 56 todos os dias
# coloque apenas alguns minutos após a hora atual para ver o efeito.
# você tem que rodar o comando “date” para ver a hora atual.
56 15 * * * echo “OLA” > /tmp/teste1
```

4.Salve o conteúdo do arquivo de tarefas e verifique o diretório /var/spool/cron/crontabs

```
$ ls -l /var/spool/cron/crontabs
```

Você verá que foi criado um arquivo com o nome do usuário.

5.Se a hora que você especificou no crontab passou, verifique se o arquivo /tmp/teste1 foi criado com sucesso

```
$ cat /tmp/teste1
```

6.No logon do usuário (fulano – tty1), rode o crontab -l

```
$ crontab -l
```
(aqui será mostrado o conteúdo do seu arquivo de tarefas)

7.Com o mesmo usuário, apague o arquivo de tarefas, através do crontab -r

```
$ crontab -r
```

8.Verifique novamente o conteúdo do arquivo de tarefas

```
$ crontab -l
```
no crontab for fulano (observe a mensagem de erro)

O cron define o valor de algumas variáveis automaticamente durante sua execução; a variável SHELL é definida como /bin/sh, PATH como /usr/bin:/bin, LOGNAME, MAILTO e HOME são definidas através do arquivo /etc/passwd. Os valores padrões destas variáveis podem ser substituídos especificando um novo valor nos arquivos do cron.

Exemplo de um arquivo /etc/crontab:

```
SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin

# Executa o comando sync todo o dia as 10:00
00 10  * * *  root sync

# Executa o comando updatedb toda segunda-feira as 06:00.
00 06  * * 1  root updatedb

# Executa o comando runq todos os dias e a toda a hora
# em 10, 20 e 40 minutos.
10,20,40 *  * * *  root runq

# Executa o comando fetchmail de 10 em 10 minutos todos os dias
*/10 *  * * *  root fetchmail

# Envia um e-mail as 0:15 todo o dia 25/12 para john desejando
# um feliz natal.
15 0  25 12 * root echo "Feliz Natal"|smail john

# Executa o comando poff automaticamente as 5:30 de
# segunda-feira a sábado.
30 5  * * 1-6   root  poff
```

11 GERENCIAMENTO DE CONTAS E CUIDADOS PARA A PROTEÇÃO DE SENHAS

Este capítulo traz explicações e comandos úteis para o gerenciamento de contas e proteção de senhas de usuários em sistemas Linux. Também explica os principais métodos usados para quebra de senha usando diversos métodos como engenharia social, brute force, etc., bem como dicas de como escolher boas senhas para você e seus usuários e métodos automatizados de checagem de senhas vulneráveis.

Estes métodos são explicados para que você entenda, se previna destes tipos de ataques além de entender a importância de políticas de proteção de senhas.

11.1 Introdução

A criação de uma conta em uma máquina Linux pode expor seu sistema (ou toda sua rede) a crackers simplesmente com a falta de treinamento e políticas de segurança. Uma invasor com um simples acesso a uma conta de usuário pode conseguir acesso a áreas que contém dados importantes expondo seu sistema a ataques ou roubo de dados.

Um firewall não pode fazer muito em uma situação dessas, um acesso através de uma conta de sistema válida é difícil de ser auditado e descoberto, a não ser que o usuário monitore seus acesso via lastlog e o administrador conheça os hábitos de seus usuários para notar o uso estranho de contas em determinados dias/horários. Evitar situações como esta depende mais de conscientização e treinamento tanto do administrador como dos usuários das contas para não expor o sistema a um ataque direto. Este capítulo do guia explicará as situações mais comuns e alguns exemplos de como tais ataques acontecem.

ATENÇÃO: - Os dados aqui disponibilizados são puramente para fins didáticos e compreensão de como tais situações funcionam para se criar mecanismos de defesa personalizados de acordo com o que deseja proteger.

11.2 Criação, monitoração e segurança de contas

Para adicionar uma conta de usuário ao sistema é simples, basta um comando adduser [usuário] e alguns poucos segundos para responder as questões do programa. Quando criamos contas para outros usuários temos 2 alternativas: deixarmos a senha em branco ou escolher uma senha que será passada ao usuário para que ele possa fazer a troca mais tarde. A primeira alternativa é muito perigosa, pois uma pessoa com acesso a /etc/passwd poderá facilmente descobrir sua lista de usuários (principalmente em uma grande

empresa quando conhecemos as políticas de criação de novas contas). Um funcionário notaria a presença do novato e poderia aproveitar esta oportunidade para tentar incriminar este usando a conta recém criada ou tentar outras coisas para obter benefício próprio através do descuido de outros.

O segundo método de senha inicial é um pouco mais seguro e de preferência a senha deve ser escolhida pelo usuário para que pessoas que conhecem o estilo de senhas iniciais escolhidas pelo administrador não possam deduzir a nova senha criada. É comum vermos senhas como "novo1234", "123456", "abcdef", "a1b3c3", ou "nome do usuário" como senhas iniciais, pois é fácil de lembrar. Senhas deste tipo são as primeiras a ser tentadas por crackers e programas específicos para este fim. Mas se o o usuário esquecer de trocar sua senha provisória?

O programa chage e passwd possui recursos que permitem definir limites mínimos e máximo do tempo para troca de senha de acesso, número máximo de dias após expirar o tempo de troca da senha em que a conta será permanentemente desabilitada (até que o administrador a reative) e o período mínimo entre troca de senhas. Alguns exemplos:

```
passwd -x 10 -w 3 teste
```

A senha do usuário teste expirará após 10 dias (-x 10) e ele será avisado com 3 dias de antecedência (-w 3) para trocar sua senha. Após o período máximo o usuário será obrigado a trocar a senha.

Quando o usuário efetuar o login receberá a seguinte mensagem: *Warning: your password will expire in 3 days.*

```
passwd -x 10 -w 3 -i 2 teste
```

A senha do usuário teste expirará após 10 dias (-x 10) e ele será avisado com 3 dias de antecedência (-w 3) para trocar sua senha, após a expiração da senha, o usuário tem 2 dias antes da conta ser desativada (-i 2). Se o período expirar e o usuário tentar um novo login será mostrada a mensagem:

Your account has expired: Please contact your system administrator

Para reativar a conta acima, remova totalmente o bloqueio da conta do usuário teste com:

```
# passwd -x 0 teste
```

ou

```
# passwd -x 99999 -w 7 -i 0 teste
```

ou especifique um período de dias maior em adição àqueles especificados para que ele possa trocar a senha.

Por exemplo, caso tenha passado 3 dias desde que a conta acima expirou e deseje dar mais 2 dias para o usuário trocar a conta: passwd -x 17 -i 0 teste A conta será reativada por mais 2 dias dando a oportunidade do usuário

trocar a senha. Preste atenção neste exemplo para entender bem a situação e prazos.

```
# passwd -x 90 -n 60 -w 15 -i 0 teste
```

A senha do usuário teste expirará após 90 dias (-x 90), ele será avisado para trocar sua senha com 15 dias antes do prazo final (-w 15) e a conta será imediatamente desativada caso o prazo máximo para troca da senha expire (-i 0). O usuário também não poderá trocar sua senha durante os primeiros 60 dias desde a última troca de senha (-n 60).

Em sistemas onde precisa adicionar restrições a muitos usuários na criação da conta, é recomendável seguir os métodos descritos em na seção “Defininfo valores padrão para restrições”.

OBS1: Em sistemas com senhas ocultas ativadas as restrições acima serão especificadas no arquivo /etc/shadow, isto garante que só o usuário root tenha acesso aos detalhes fornecidos neste arquivo.

OBS2: A *-d* do passwd serve para remover a senha do usuário especificado ou seja somente será necessário fornecer o nome de usuário para ter acesso ao sistema.

OBS3: Leve em consideração que o uso do recursos de senhas de grupo é um risco de segurança, pois a mesma senha será compartilhada entre diversas pessoas.

OBS4: O programa useradd combina as funções do adduser e passwd para garantir que a conta seja criada com as restrições apropriadas. O único inconveniente é que o useradd quebra o *Debian Policy* e precisa de todos todos os parâmetros para a criação correta da conta (como o diretório home, senha criptografada, e UID numérico). Seu uso é indicado em shell scripts que cuidam automaticamente da tarefa de adicionar usuários ao sistema.

Exercício

1.Como root, faça com que a senha do usuário fulano expire em 5 dias e peça um aviso de 2 dias antes de expirar

```
# passwd -x 5 -w 2 fulano
```

2.Em outro terminal, entre como fulano

```
login: fulano
Passwd:
Warning: your password will expire in 1 day
```

Observe a mensagem de aviso.

11.2.1 Definindo valores padrão para restrição

Isto é muito útil quando precisa criar diversos usuários com as mesmas restrições de contas e tornará o gerenciamento do sistema muito mais prático (tudo em Unix é feito para ser mais prático, só devemos saber onde mexer). O arquivo /etc/default/useradd contém valores padrões que serão usados pelo useradd e adduser para definir valores de restrições de contas. Estes valores são gerados usando a opção *-D* em combinação com as seguintes opções do useradd:

- -b [home] - Especificar o diretório home de usuário. O padrão é /home.
- -e [data] - Data padrão de expiração de contas, especificada no formato AnoMesDia. Por exemplo, 20010920.
- -f [dias] - Número máximo de dias que a conta permanece válida após a data de expiração até ser desativada.
- -g [gid/grupo] - ID do grupo ou nome do grupo que o usuário pertencerá inicialmente.
- -s [shell] - Shell do usuário. O padrão é /bin/bash.

OBS: Note que nem todas as opções acima terão efeito com o adduser (principalmente as opções *-f*, *-g* e *-s* que são especificadas no seu arquivo de configuração /etc/adduser.conf).

O arquivo /etc/default/useradd só existe se algum parâmetro foi modificado através do “useradd -D”.

11.2.2 Senhas fáceis de adivinhar e escolha de boas senhas

A senha lhe identifica como o verdadeiro dono de uma conta em um sistema para garantir acesso a seus recursos. A senha de um sistema é tão importante quanto uma senha de sua conta bancária, caso caia em mãos erradas as conseqüências poderão ser catastróficas, todo cuidado é pouco na hora de escolher uma senha.

Senhas fáceis de adivinhar são o primeiro motivo de sucesso de crackers no acesso a sistemas de computadores, o administrador pode forçar o usuário a fazer trocas periódicas de senhas através dos recursos citados em **Error! Hyperlink reference not valid.**, mas quem vai garantir que ele esteja escolhendo boas senhas para que ninguém as descubra com facilidade? Abaixo uma lista de senhas ruins (que deverá evitar a todo custo usá-las) e boas:

11.2.2.1 Senhas Ruins

- O uso da palavra senha como senha! Isto parece idiota mais existe...
- Senhas com o mesmo nome do login (joao/joao).
- Compostas por letras ou números em seqüencia crescente ou decrescente (abcdef, 123456, 654321, abc123, etc, etc). Este tipo de senha pode ser adivinhada por dedução e são uma das primeiras combinações que crackers usam para acertar senhas.
- palavras relacionadas com o gosto pessoal. Por exemplo "escort", "vectra", "subaru" se a pessoa é amante de carros.
- Nome da esposa, filhos, familiares, animal de estimação, time de futebol, ídolo da TV/filmes ou qualquer coisa relacionada a familiares ou indiretamente ao usuário.
- Idade, data de aniversário, data de casamento, número de identidade, título de eleitor, placa de carro ou qualquer coisa que seja característica do usuário.
- Palavras existentes. Um ataque de dicionário poderá descobrir facilmente sua senha.
- Senhas com menos de 8 letras
- Senhas apenas em minúsculas ou MAIÚSCULAS.

11.2.2.2 Senhas Boas

- Uma boa senha nunca deverá ser lida mas fácil de lembrar. Por exemplo pense em uma frase importante para você "meu sistema operacional preferido é o Linux" e pegue a primeira letra de cada palavra: "msopeol". PRONTO esta escolhida uma boa senha que é fácil de se lembrar e difícil de ser quebrada por ataques de dicionário!
- Uma boa senha deve conter números e letras. A senha acima poderia ser modificada para "msopeol1"
- Conter letras maiúsculas e minúsculas. "msopeoL1".
- Conter 8 caracteres sempre que possível. Isto aumenta bastante o número de combinações necessárias para se quebrar uma senha em um ataque brute force (veja adiante). Mesmo que a senha escolhida não chegue a 8 caracteres mínimos, você poderá combiná-la com números.

Com as dicas acima, a possibilidade de alguém conseguir quebrar uma senha criptografada em seu sistema usando os ataques descritos adiante é praticamente nula! Para os paranóicos de plantão, o utilitário makepasswd pode criar uma senha com caracteres completamente aleatórios:

```
# makepasswd --chars 8
# 4y0sBdwM
```

Este comando retorna uma string com 8 caracteres (--chars 8) "4y0sBdwM". Se você entendeu boa parte deste guia tenho certeza que 1 ou 2 dias de treino e se acostuma com uma senha como esta ;-)

OBS: NUNCA dê pistas sobre sua senha! Para você isto pode ser um desafio lançado a outras pessoas quase impossível de ser resolvido, mas não se esqueça que muita gente é especializada neste tipo de dedução.

11.2.3 Atualização de senhas de múltiplas contas

O programa chpasswd é usado para tal operação. Deve ser especificado um arquivo que contém os campos usuário:senha por linha. Caso as senhas estejam encriptadas deverá ser especificada a opção *-e* ao programa.

```
chpasswd -e /localadmin/contas/contas.db
```

O comando acima atualiza a senha de todos os usuários especificados no arquivo contas.db de uma só vez.

11.2.4 A senha do usuário root

Esta seção foi retirada do Manual de Instalação da Debian.

A conta root é também chamada de *super usuário*, este é um login que não possui restrições de segurança. A conta root somente deve ser usada para fazer a administração do sistema, e usada o menor tempo possível.

Qualquer senha que criar deverá conter de 6 a 8 caracteres, e também poderá conter letras maiúsculas e minúsculas, e também caracteres de pontuação. Tenha um cuidado especial quando escolher sua senha root, porque ela é a conta mais poderosa. Evite palavras de dicionário ou o uso de quaisquer outros dados pessoais que podem ser adivinhados.

Se qualquer um lhe pedir senha root, seja extremamente cuidadoso. Você normalmente nunca deve distribuir sua conta root, a não ser que esteja administrando um computador com mais de um administrador do sistema.

Utilize uma conta de usuário normal ao invés da conta root para operar seu sistema. Porque não usar a conta root? Bem, uma razão para evitar usar privilégios root é por causa da facilidade de se cometer danos irreparáveis como root. Outra razão é que você pode ser enganado e rodar um programa *Cavalo de Tróia* -- que é um programa que obtém poderes do *super usuário* para comprometer a segurança do seu sistema sem que você saiba.

11.3 Tipos de ataques mais comuns para se conseguir uma senha.

11.3.1 Dedução

O cracker se aproveita da ingenuidade de usuários que deixam senhas em branco, usam senhas simples como o próprio nome, "abcdef", "asdfg", "123456", e outros tipos de senhas comuns para tentar obter acesso ao sistema. Senhas deduzidas são geralmente senhas muito simples e muito usadas. Uma situação comum para a escolha de uma senha deste tipo é o medo de esquecer a senha (quando não se consegue pensar em algo mais difícil e ao mesmo tempo que seja fácil de lembrar) e quando o usuário é pego desprevenido e não se sabe o que usar como senha (como na assinatura de um provedor Internet, muito comum essa situação).

Geralmente é muito rápido e muito eficaz dependendo das habilidades do atacante dispõe.

11.3.2 Engenharia Social

Ataques por engenharia social são feitos através de pesquisa de dados pessoais e outras características relacionadas ao usuário (time de futebol, data de nascimento dele, da esposa, filhos, nome da atriz predileta, etc) e usando estes dados coletados para auxiliar na descoberta da senha. Este ataque requer uma pesquisa sobre os hábitos, gostos, etc. Mas existem outros tipos de ataque baseados em engenharia social, inclusive com o cracker passando-se pelo usuário. Para diminuir as possibilidades deste tipo de ataque entenda e siga os procedimentos da parte "Senhas Boas" na seção **Error! Hyperlink reference not valid.** e continue lendo esta seção.

Outro detalhe importante para diminuir as possibilidades de um ataque deste tipo bem sucedido é permitir somente o acesso do serviço de finger a redes confiáveis (locais onde uns conhecem os outros). Os detalhes fornecidos pelo finger podem ser suficientes para garantir sucesso deste tipo de ataque:

```
#finger joao
Login: joao                          Name: Joao P.  M.
Directory: /home/joao                Shell: /bin/bash
Office: Sala 400 Andar 2, 123-4567  Home: 123-7654
Last login Fri Aug 25 21:20 (AMT) on tty3
No mail.
Grupo de cadastramento.
```

As últimas linhas da saída do finger são os dados contidos nos arquivos .plan e .projects do diretório de usuário. O cracker com base nos dados fornecidos acima pelo finger poderia inventar uma situação em que necessitaria de troca de senha por algum motivo. Abaixo uma situação onde o cracker sabe que não existe identificador de chamadas na empresa e conhece as fragilidades:

- Ø Cracker: Disca para o CPD...

- Ø Vítima: CPD?

- Ø Cracker: Oi, eu sou o João do grupo de cadastramento aqui do segundo andar, estou tentando entrar no sistema mas por algum motivo ele não aceita minha senha (fazendo-se de ignorante no assunto).

- Ø Vítima: Por favor Sr. verifique se o Caps Lock do seu teclado está ativado, letras em maiúsculas/minúsculas fazem diferença em nossos sistemas.

- Ø Cracker: Ok vou checar (espera um tempo). Não, está tudo Ok, você poderia agilizar isto de alguma maneira? Preciso lançar algumas fichas no sistema.

- Ø Vítima: Posso modificar sua senha para um nome qualquer, depois você poderá trocar por si próprio.

- Ø Cracker: Ok, por mim tudo bem.

- Ø Vítima: Humm, modifiquei para "cad1234", basta você usá-la e terá acesso ao sistema. Após isso execute o utilitário passwd para trocá-la para algo que desejar.

- Ø Cracker: Ok, muito obrigado. Tenha um bom dia.

Este é um exemplo simples de ataque por engenharia social. Dependendo do objetivo, este tipo de ataque pode levar semanas e às vezes requer contatos com diversas empresas criando diversas situações para obter detalhes necessários para atingir o objetivo.

As políticas de segurança de senhas minimizam riscos deste tipo. Como este é um caso que o requisitante é um funcionário próximo do departamento de informática, o mais adequado seria o administrador se deslocar ao setor (ou enviar um técnico do setor treinado para tal situação) para saber se quem diz ser quem é está realmente no local enfrentando aquela situação. O contato com o responsável do setor (conhecido do técnico) também pode ser uma alternativa antes de entregar uma senha a um desconhecido.

Para casos externos (principalmente para empresas que mantém determinados serviços em funcionamento em nosso servidor, como servidores de páginas), o procedimento correto seria passar uma nova senha por e-mail (de preferência criptografado com pgp) ao invés de telefone. Isto garantirá que a senha não caia nas mãos erradas.

OBS1: Qualquer detalhe sobre a política de criação de senhas, trocas de senhas, etc. poderá ter muito valor para um cracker obter acesso ao seu sistema.

OBS2: Dificulte as maneiras para se obter acesso root ao sistema via conta de usuário comum. É de extrema importância utilizar conexões de dados criptografadas quando for necessário acesso externo ao seu sistema.

OBS3: Nunca use uma mesma senha para fazer tudo (banco, acessar seu sistema, conectar-se ao seu provedor, senha de root). Você estará em sérios apuros caso alguém tenha acesso a esta senha. É difícil lembrar de várias senhas, mas você pode aditar uma senha e criar modificações a partir dela para utilização em outros locais, por exemplo: "wekpdm" => "Bwekpdm1" => "3wekpdmS", etc.

11.3.3 Ataques por dicionário

De posse do arquivo de senhas /etc/passwd, o cracker utiliza um arquivo que contém diversas palavras que serão tentadas como senha. Este trabalho é feito automaticamente por ferramentas dedicadas a este tipo de tarefa e pode levar dias dependendo da lista de senhas do cracker e quantidades de usuários existentes no arquivo de senha.

Note que o uso de criptografia *md5* e *senhas ocultas* dificulta bastante ao arquivo de senhas e o sucesso de um ataque bem sucedido.

11.3.4 Força Bruta

De posse do arquivo de senhas /etc/passwd o cracker utiliza uma ferramenta que tenta diversas combinações de letras seqüencialmente na tentativa de descobrir uma senha. Este ataque geralmente é usado como último recurso após um ataque por dicionário, e leva muito tempo para descobrir uma senha.

Dependendo se uma senha conter caracteres aleatórios, combinação de letras maiúsculas/minúsculas, números, a senha será praticamente impossível de ser descoberta. Note que o uso de criptografia *md5* e *senhas ocultas* aumenta bastante a proteção das senhas.

11.3.5 Monitoração de toques do teclado

Este ataque é muito comum em sistemas DOS e Windows, um programa é instalado sem o conhecimento do usuário que grava todos os toques do teclado em um arquivo escondido pelo cracker. Após certo tempo o cracker obtém acesso ao arquivo e aos dados que ele contém. Este tipo de ataque é muito perigoso e pode capturar senhas não só do sistema como números de cartão de crédito digitados (caso o usuário tenha feito compras on-line), conta bancária+senha e tudo mais que for digitado pelo teclado.

11.3.6 Login falso

Esta é uma forma rápida de se conseguir acesso a um sistema. É criada uma tela de login idêntica a original do sistema, só que ao digitar nome e senha, estes são gravados em um arquivo (que será mais tarde recuperado pelo cracker para obter acesso ao sistema) e uma mensagem de erro será exibida pelo sistema.

Naturalmente o usuário pensará que digitou o nome/senha incorretamente e fará uma nova tentativa, a segunda ocorrerá com sucesso (fazendo este pensar que errou *mesmo* a senha).

Sua atenção é muito importante para evitar este tipo de ataque, caso desconfie de algo errado, entre no sistema e dê um find / -type f -cmin -3 para localizar os arquivos modificados nos últimos 3 minutos e localizar possíveis bancos de dados de senhas.

Outra alternativa é realmente digitar uma senha inválida intencionalmente (e diferente da correta) e na segunda tentativa lançar a senha válida (normalmente sistemas deste tipo bem elaborados chamam o verdadeiro sistema de login na segunda tentativa).

11.4 Melhorando a segurança das senhas armazenadas em seu sistema

11.4.1 Shadow Passwords

Senhas Ocultas (shadow passwords) aumentam consideravelmente a senha do seu sistema pois as senhas serão armazenadas em um arquivo separado: /etc/shadow para senhas de usuários e /etc/gshadow para senhas de grupos. Estes dois arquivos poderão ser acessados somente pelo usuário root. O armazenamento de senhas no arquivo /etc/passwd e /etc/groups não é seguro, estes arquivos devem ser lidos por todos os usuários porque muitos programas mapeiam a UID do usuário com seu nome e vice versa.

O utilitário shadowconfig é usado para ativar/desativar o suporte a senhas ocultas (de usuários e grupos) em seu sistema. Adicionamente os utilitários pwconv/grpconv podem ser usados separadamente para ativar o suporte a senhas ocultas de usuários/grupos e pwunconv/grpunconv para desativar este suporte.

Exercício:

1.Como root, dê um cat no arquivo /etc/passwd

```
# cat /etc/passwd
```

Você vai notar que os campos estão separados por ":" e que o segundo campo de cada entrada (cada linha) é um "x", o que significa que as senhas estão no arquivo /etc/shadow

2.Desative o suporte a senhas shadow com o comando shadowconfig

```
# shadowconfig off
```

3.Dê novamente um cat no /etc/passwd

```
# cat /etc/passwd
```

Note que o segundo campo agora contém caracteres estranhos, ou "*". Um "*" significa uma senha bloqueada. Os caracteres estranhos, como "plEKAPg5eWBNti" representam a senha criptografada pelo sistema.

4.Reative o suporte a senhas shadow:

```
# shadowconfig on
```

5.Inspecione agora o arquivo /etc/shadow

```
# cat /etc/shadow
```

Note que os campos de senha estão neste arquivo agora. O arquivo shadow somente pode ser lido pelo root.

ATENÇÃO: Caso você inclua usuários em grupos manualmente no arquivo /etc/passwd, também precisará fazer isto no arquivo /etc/shadow para que não tenha problemas. Esta tarefa é feita automaticamente com o comando adduser usuário grupo. Os programas vipw e vigr também podem ser usados com a opção *-s* para editar os arquivos /etc/shadow e /etc/gshadow respectivamente.

11.4.2 Senhas MD5

O sistema de criptografia usado pelas senhas MD5 é mais seguro que o padrão Crypto e permitem o uso de senhas maiores do que 8 caracteres.

O uso de senhas MD5 é recomendado para aumentar o nível de proteção da senha. Não use caso estiver executando um serviço de NIS. **OBS:** Caso utilize senhas MD5 em um sistema com PAM, inclua a palavra md5 na linha de configuração do método de autenticação password do módulo pam_unix.so:

```
password required pam_unix.so md5
```

12 O X WINDOW

O sistema X Window é um ambiente de trabalho gráfico disponível para o GNU/Linux e para muitos outros sistemas Unix (e não Unix). Com efeito, o X é mais que um simples ambiente gráfico. Graças ao uso do protocolo X, o sistema X Window é "network transparent" (transparente à rede) e é capaz de executar aplicações distribuídas (cliente-servidor). Isto significa, a uma primeira aproximação, que é possível lançar um programa em um host "client" e visualizar a saída gráfica em um outro host "server" de modo totalmente transparente (transparente significa, nesse caso, que não é necessário modificar a aplicação para obter este resultado). O sistema X Window é produto do "X Consortium" e a edição corrente é a X11R6. A versão do X usada pelo Debian é a XFree86, uma implementação open source livremente redistribuível do sistema X Window.

Quando começamos a usar o X encontramos alguns termos cujo significado pode parecer inicialmente obscuro. Os elementos de base do X são:

- Hardware de vídeo suportado pelo XFree86.

- Um *servidor X* interfaciando-se com o hardware. O servidor X fornece um modo padrão para abrir janelas, efetuar operações gráficas (por exemplo, utilizar fontes para a visualização do texto) e para ler os movimentos do mouse, as entradas do teclado ou de outros periféricos. X é transparente à rede e, portanto, é possível acionar cliente X em uma máquina e o servidor de X correspondente (isto é, o monitor com o hardware de vídeo) em outra.

- Um *window manager* (gerenciador de janelas) que utiliza o servidor X. O window manager é basicamente um tipo especial de cliente X a que é permitido manipular o posicionamento e o aspecto das janelas. Pode-se "decorar" as janelas com "widgets" padrões, que geralmente permitem deslocar, redimensionar, reduzir a ícone, etc., as janelas. Um gerenciador de janelas pode ser dotado também de outras funcionalidades (fechar as janelas, encerrar os programas, mostrar o menu dos programas, etc.) ou de efeitos especiais (sombreamento, decorações, etc.) e assim por diante. O XFree86 já dispõe de um gerenciador de janelas simples chamado twm. Não é muito sofisticado mas muitos o acham mais que suficiente. Exemplos de Window Managers: FVWM, FVWM95, TWM, SAWFISH

- Um *desktop manager* ou gerenciador das telas de trabalho (opcional). KDE e GNOME são dois exemplos de desktop managers. Trata-se de conjuntos de programas (suítes) mais ou menos integrados e coordenados, projetados para fornecer ao usuário um conjunto de aplicações de base dotadas de uma interface comum. O gerenciador da área de trabalho (desktop manager) pode permitir o acesso ao sistema de arquivos utilizando a metáfora do desktop (mesa de trabalho), pode dispor de um browser de navegação no

help, terminais que substituem o clássico xterm, aplicativos de gerenciamento de áudio, ambiente de desenvolvimento, editor e assim por diante. Os desktop managers oferecem características que podem facilitar a aproximação ao sistema de usuários provenientes de outros ambientes, como os Macintosh ou uma das tantas versões do MS-Windows.

- Todos os outros aplicativos (clientes X de terceiros) instalados no sistema. Estes "falam" com o servidor X e com o gerenciador de janelas (geralmente o desktop manager não está envolvido de modo particular naquilo que os aplicativos executam).

Nota: aqui já deveria estar claro que os desktops como GNOME e KDE têm seu próprio window manager, embora também possam usar um externo compatível.

Normalmente se utiliza apenas um window manager por vez em cada servidor X (mas é possível ativar mais de um servidor X em um mesmo computador).

12.1 Configuração e Inicialização

Para Instalar o X Window, você pode usar a ferramenta dselect e selecionar o pacote *task-x-window-system "X Window System (complete)"*. Além desse, aconselho que você instale:

- *xserver-svga* – para poder usar uma resolução melhor do que 640x480 com 16 cores

- outros gerenciadores de janelas, como o FVWM, FVWM95, SAWFISH (ou SAWMILL) ou WMAKER. Assim vai poder experimentar qual se adapta melhor a sua forma de trabalho

- um gerenciador de telas de trabalho, como o GNOME (*task-gnome-desktop* – GNOME basic desktop)

Depois de instalar o X, você deve configurar o arquivo /etc/X11/XF86Config de acordo com o seu hardware. Para ter uma idéia da complexidade desse arquivo, veja o exemplo em /usr/share/doc/xserver-common/examples/XF86Config.eg. Dê um more nesse arquivo:

```
# more /usr/share/doc/xserver-common/examples/XF86Config.eg
```

Não vamos iniciar configurando esse arquivo manualmente, mas para que você possa configurar minimamente o X, precisará ter os seguintes dados disponíveis:

- tipo e marca do mouse (serial, PS2, compatível com Microsoft, dois ou três botões).

- Freqüências verticais e horizontais suportadas pelo monitor. Essa informação você consegue no manual do fabricante ou no site Internet do fabricante, ou ainda no serviço de atendimento ao consumidor. O monitor Studioworks 550M, por exemplo, suporta as seguintes freqüências:
  Freqüência Horizontal: 30-54 KHz
  Freqüência Vertical: 50-90 Hz
  Resolução Máxima: 1024x768 @ 60 Hz

- Tipo de teclado. Normalmente estará usando um teclado com padrão ABNT2.

- Marca, modelo e memória da placa de vídeo e memória disponível. Em nosso caso, usaremos a placa Sis530, com 2MB de memória. Você pode identificar corretamente a Marca, modelo e memória de sua placa chamando o utilitário SuperProbe. Exemplo:

```
# SuperProbe

First video: Super-VGA
Chipset: Silicon Integrated Systems 530/620 (PCI Probed)
Memory: 2048 Kbytes
RAMDAC: Sis built-in DAC w/clock
```

Para configurar o arquivo, podemos usar um de dois programas:

- xf86config – um programa em modo texto que gera, ao final, o arquivo XF86Config

- XF86Setup – programa gráfico utilizado para a mesma função.

Em nosso exemplo, utilizaremos o programa xf86config (modo texto), que com certeza irá funcionar em qualquer ambiente.

Para ter uma idéia da configuração, faça uma cópia de segurança do XF86Config atual e chame o xf86config:

```
# cp /etc/X11/XF86Config /etc/X11/XF86Config.copia
# xf86config
```

Ao chamar o xf86config, ele apresenta uma primeira tela que diz que você pode usar o arquivo exemplo para uma configuração inicial ou deixar que o programa gere um XF86Config a partir das informações que você fornecer. Pressione <Enter> para continuar.

A próxima tela pede para você selecionar o tipo de mouse que possui. Se a máquina que você está usando tem um mouse serial, a opção 1 (Microsoft compatible) deve funcionar. Se tem um mouse PS/2, selecione a opção 4.

A seguir, o programa pergunta se você quer habilitar o ChordMiddle (*Do you want to enable ChordMiddle?*) , responda “y”. O *ChordMiddle* é uma opção de alguns mouses Logitech para habilitar o terceiro botão. Provavelmente você deve responder “n” (não) a esta pergunta.

A pergunta seguinte se refere a uma emulação de três botões para o seu mouse (isto se você já não tinha escolhido um mouse de três botões). Alguns gerenciadores de janela usam o terceiro botão (como o sawfish, por exemplo). Você pode responder sim ou não a esta pergunta (y ou n).

A seguir, deverá indicar em que dispositivo seu mouse está conectado. Sugiro usar o /dev/mouse. Apenas terá que fazer um link desse arquivo (/dev/mouse) para o real dispositivo. Se o mouse está na serial 1 (COM1, para quem vem do DOS ou MS-Windows), você deve fazer um link de /dev/mouse para /dev/ttyS0.

A próxima tela do xf86config indica que a próxima etapa é a configuração do teclado. Pressione <Enter>

Selecione agora um teclado compatível com o que estiver usando. Muito provavelmente você estará usando o teclado com padrão brasileiro ABNT2. Entre o número correspondente e pressione <Enter>.

Após, o programa indica que a próxima etapa é a configuração do monitor, e que você deve ter em mãos os valores de freqüência horizontal e freqüência vertical. Pressione <Enter>

Esta etapa pode parecer confusa, mas de posse dos dados fornecidos pelo fabricante fica bem mais fácil decidir qual a melhor opção para o nosso caso. Você tem que selecionar uma opção com faixa de freqüência que esteja dentro da faixa de freqüência suportada pelo seu monitor. Em nosso caso, a melhor opção é:

```
6  31.5 – 48.5; Non-Interlaced SVGA, 1024x768 @ 60 Hz, 800x600 @ 72 Hz
```

A seguinte pergunta se refere a faixa de freqüência vertical suportada pelo monitor. **É muito importante nunca selecionar um valor maior do que o suportado. Isso pode causar danos irreversíveis ao hardware**. Em nosso caso, a opção correta, conforme consta no manual do monitor, é:

```
2   50-90
```

A seguir você deve fornecer dados sobre a identificação do monitor. Essa etapa é opcional, você pode simplesmente pressionar <Enter> às seguintes perguntas:

```
Enter an identifier for your monitor definition:
Enter the vendor name of your monitor:
Enter the model name of your monitor:
```

Na próxima tela o programa irá perguntar se você quer escolher sua placa de vídeo a partir da base de dados das placas suportadas (*Do you want to look at the card database?*). Sugiro que você responda sim, mas esteja certo do modelo e fabricante de placa que possui, de outra forma, a sua configuração pode não funcionar corretamente.

A seguir, é apresentada uma lista de placas suportadas. Você pode colocar um número para escolher uma placa ou ir rolando as páginas com a

tecla <Enter>. Se por acaso passar do nome que queria, continue pressionando <Enter> para chegar ao início da lista novamente. A nossa escolha será:

```
622   SiS 530
```

O programa irá confirmar sua seleção e pedir para que pressione <Enter> para continuar.

As próximas opções apresentadas referem-se ao tipo de servidor que vai rodar (mono, VGA, SVGA, etc). Muito provavelmente, se você instalou o pacote *x-server-svga*, pode escolher o servidor XF86_SVGA. Isso vai permitir que você rode com em resoluções ou número de cores superiores ao pacote VGA, onde você estaria limitado a 16 cores e 640x480. Note que o programa já indica a melhor opção para sua placa de vídeo (opção 5). Isso, no entanto, não quer dizer que essa opção é suportada pela sua instalação atual. Você não pode, por exemplo, escolher o servidor XF86_SVGA se não estiver instalado o pacote *xserver-svga*.

A próxima pergunta é se você quer modificar o arquivo /etc/X11/Xserver. Se você quer que o servidor que escolheu na opção anterior seja usado na próxima vez que executar o X, certamente deve responder sim (y) a esta pergunta (Do you want this program to modify the /etc/X11/Xserver file?).

A próxima informação que você deverá fornecer é a memória da placa de vídeo. Isso é importante para que o X determine quais resoluções e profundidades de cor que você poderá usar. Quanto mais memória sua placa tiver, maior a resolução e o número de cores possíveis. Algumas placas compartilham a memória com o micro. Nesse caso, você pode configurar quanta memória quer para a placa de vídeo pela BIOS. De qualquer forma, você deve saber quanta memória possui para configurar o X. Vamos selecionar 2048K:

```
4   2048K
```

Agora o programa vai pedir algumas informações opcionais para identificação de sua placa de vídeo. Você pode pressionar <Enter> em cada uma para que ele utilize valores default:

```
Enter an identifier for your video card definition:
Enter the vendor name of your video card:
Enter the model (board) name of your video card:
```

A próxima pergunta é se você quer usar um Clockchip e qual. A maioria das placas modernas não necessitam um e, além disso, o X pode detectá-lo automaticamente. Pressione <Enter> indicando que você não quer selecionar um Clockchip.

Na próxima tela, o xf86config pergunta se quer fazer um teste no servidor X (Do you want me to run 'X -probeonly' now?). Responda sim (y) para testarmos a configuração atual. O programa ainda vai dar uma mensagem de aviso e pedir que você confirme pressionando <Enter>. Vá em

frente. Se ocorrer um erro, como “X -probeonly failed”, ou a máquina pendurar ou a tela ficar em branco, certamente você precisa rever suas configurações. Se tudo estiver ok, a tela irá chavear para modo gráfico e retornar para o prompt do programa de configuração. Pressione <Enter> para continuar. Nota: se você já estiver rodando o X Window, o teste irá falhar.

A próxima tela irá permitir que você configure a ordem das resoluções que o servidor X irá usar para iniciar em cada modo de profundidade de cores. A unidade está em bpp (bits por pixel). Para você ter uma idéia, abaixo deixamos uma tabela com o número máximo de cores para cada modo:

```
8bpp  – 256 cores
16bpp – 32.000 ou 64.000 cores
24bpp – True color – modo compactado
32bpp – True color
```

Nesta etapa, você pode aceitar a ordem mostrada, ou alterar a ordem para cada profundidade de cores. Se resolver mudar as configurações, o programa também permitirá que você indique se quer ou não uma tela virtual maior do que a resolução permite. Se você disser que sim, quando o mouse chegar nas bordas da tela você vai ver a tela correr, pois a tela virtual é maior do que a que aparece no monitor. Nota: sugiro que você modifique a ordem dos modos de resolução em cada opção de profundidade. O default é que o programa irá tentar iniciar na menor resolução possível, mas certamente você vai querer usar numa resolução maior que 640x480. Faça suas alterações seguindo as instruções na tela. Você pode, por exemplo, selecionar “4” para mudar a opção dos modos True Color (32bpp) e depois indicar “432”, querendo dizer para tentar primeiramente o modo “1024x768”, depois “800x600” e por último “640x480”. Depois indicar que não quer usar uma tela virtual maior que a tela real (*Do you want a virtual screen that is larger than the physical screen?*). Observe as mudanças na tela apresentada e, se tudo estiver como quer, selecione a opção 5 (The modes are OK, continue.) para continuar.

A última pergunta é se você quer salvar a configuração no arquivo XF86Config. Se você quer que as modificações façam efeito, responda sim (y).

Bom, você chegou ao fim das configurações. Com o comando more, dê uma olhada no arquivo XF86Config gerado. Você poderá reconhecer várias partes de acordo com as escolhas que fez.

```
# more /etc/X11/XF86Config
```

Certifique-se que o X já não esteja rodando e que não tenha um arquivo chamado “.xinirc” no seu diretório HOME e chame o startx.

```
# cd ~
# mv .xinitrc .xinitrc.old
# startx
```

Se você configurou tudo corretamente, o X Window iniciará. Note que ele vem com a configuração padrão na profundidade de 256 cores. Se você não tem uma placa muito muito velha, certamente irá querer usar mais cores

na sua tela. Para sair do X, pode usar o menu (botão da direita, opção Logout), ou usar a combinação de teclas <Ctrl>+<Alt>+<BackSpace>.

Vamos editar o arquivo XF86Config manualmente para alterar o modo inicial de profundidade de cores.

```
# vi /etc/X11/XF86Config
```

Procure, dentro do arquivo, a seção que inicia com o comentário “Screen sections”. Logo em seguida, inicia 'Section “Screen”', abaixo da opção “Device”, insira a linha

```
DefaultColorDepth 16
```

Depois, salve e saia do vi.

Entre no X Window novamente e note a diferença!

```
# startx
```

Nota: se você usar novamente o utilitário xf86config as opções alteradas manualmente serão perdidas.

Dica: dentro do X você pode chavear a resolução através das combinações de teclas <Ctrl>+<Alt>+<+> ou <Ctrl>+<Alt>+<->.

Para maiores informações sobre opções dentro do arquivo XF86Config, refira-se à página de manual:

```
# man XF86Config
```

12.2 Personalizando o X

O gerenciador de janelas nativo do X Window é o twm. Trata-se de um window manager simples mas eficaz, tanto que muita gente não sente a necessidade de procurar outros e continuam a usá-lo. Naturalmente, é possível instalar um outro window manager, assim como personalizar a aparência do twm. Por exemplo, dentro do X, em uma janela xterm, tente dar o seguinte comando:

```
# xsetroot -solid DarkSeaGreen
```

Há vários maneiras de personalizar a configuração (isto é, o aspecto das janelas) do X já na inicialização. Uma das mais simples é criar um arquivo .xinitrc no próprio diretório HOME e alterar o gerenciador de janelas padrão.

```
# cd ~
# vi .xinitrc
```

Experimente colocar uma única linha no seu novo .xinitrc:

```
# exec fvwm95
```

Isso fará com que seu gerenciador de janelas padrão seja o fvwm95. Agora chame o X Window:

```
# startx
```

Note que o ambiente já está bem diferente.

Experimente mudar os gerenciadores de janela entre estes: fvwm, icewm, sawmill (ou sawfish, veja o que está instalado), e wmaker.

Agora experimente chamar iniciar alguns aplicativos através de seu arquivo .xinitrc:

```
# iniciando alguns programas interessantes:
xclock -geometry 50x50-1-1 &
xterm -geometry 80x34-1+1 -bg OldLace &
exec wmaker
```

Observe que as chamadas de programas antes do exec do gerenciador de janelas devem ser feitas em segundo plano (indicado pelo caracter &). O exec irá transferir toda a execução do processo para o gerenciador de janelas, e deve ser a última linha do arquivo. Para detalhes das opções suportadas por cada aplicativo, refira-se às páginas de manual (man xclock, etc).

12.3 GNOME: um Gerenciador de Ambiente

Como já mencionamos anteriormente, um desktop manager é um conjunto de aplicativos integrados e coordenados, projetados para fornecer ao usuário um conjunto de aplicações de base dotadas de uma interface comum.

O GNOME é um exemplo de Desktop Manager. Ele fornece um ambiente de trabalho integrado com o qual os usuários podem configurar seus computadores. Inclui uma barra de ferramentas de onde se pode iniciar os aplicativos e indicar o estado do sistema. Também mostra um ambiente de trabalho onde os aplicativos ficam posicionados e um conjunto de ferramentas para configurar o ambiente. Define também convenções para facilmente integrar ferramentas diferentes, tornando-as consistentes umas com as outras.

A definição do projeto no site oficial (www.gnome.org) diz:

"*O projeto GNOME nasceu como um esforço para criar um ambiente de desktop inteiramente livre para sistemas livres. Desde o início, o principal objetivo do GNOME tem sido fornecer um conjunto de ferramentas e um desktop de fácil utilização.*

Como a maioria dos programas GNU, o GNOME foi projetado para rodar em todas as variantes modernas de sistemas operacionais semelhantes ao UNIX."

Para rodar o ambiente GNOME, basta que você edite o ".xinitrc" no seu diretório HOME, como fizemos anteriormente para mudar o gerenciador de

janelas. Ao invés de chamar um gerenciador de janelas, vamos chamar o GNOME.

Edite o .xinitrc e coloque a seguinte chamada

```
# arquivo .xinitrc.  A linha abaixo chama o ambiente GNOME
exec gnome-session
```

Agora basta você chamar o X Window com o “startx”

```
# startx
```

Pronto, você deve estar rodando o GNOME. Ele cria um diretório chamado “.gnome” em seu diretório HOME e salva ali toda a personalização que você fizer em sua área de trabalho.

Experimente chamar, através do menu da barra de tarefas, o GNOME Control Center. Com essa ferramenta você pode alterar o pano de fundo de seu ambiente, o protetor de telas, e assim por diante. Pode até mesmo alterar o Gerenciador de Janelas utilizado junto com o GNOME. O Gnome vem com um próprio, mas você pode querer usar o WMAKER, por exemplo.

As possibilidades de configuração são bem amplas, e cada usuário pode ter um ambiente completamente diferenciado.

Ao sair da sessão, você pode gravar as alterações que fez no ambiente para que da próxima vez tudo esteja da forma que você deixou.

Se você não precisa de gerenciamento de sessões, pode usar o gnome-wm ao invés do gnome-session no .xinitrc. Nesse caso, as alterações em seu ambiente de trabalho não serão salvas.

12.4 Login gráfico com um Display Manager

Quando se utiliza sempre o X para trabalhar, pode ser mais prático inicializá-lo diretamente com a inicialização do computador e ter um login gráfico. Isso pode ser feito através de um Display Manager. Existem vários, dentre eles: XDM, GDM, KDM e WDM.

A vantagem é que na inicialização da máquina, um login gráfico é apresentado. Alguns display managers podem ser configurados para colocar uma foto de cada usuário, bastando dar um duplo clique sobre o mesmo e digitar a senha para entrar na sessão personalizada do X. O gdm tem esse recurso, por exemplo.

Para instalar um display manager no Debian GNU Linux, basta usar a ferramenta dselect, selecionar o pacote e mandar instalar. A mesma coisa para desinstalar. Observe apenas que somente um Display Manager é permitido. Se você tentar instalar mais de um, o gerenciador de pacotes indicará o conflito e indicará a remoção de um deles.

12.4.1 O GDM

O GDM é o Display Manager do GNOME. Após instalá-lo, você pode alterar algumas configurações através da edição do arquivo /etc/X11gdm/gdm.conf. Algumas opções que você pode querer mudar são:

- Permitir login do usuário root – isso não é recomendado, mas se você é o único usuário de um sistema caseiro, talvez seja interessante
  ```
  [security]
  AllowRoot=1
  ```

- Mostrar imagens dos usuários. Essa opção é interessante para sistemas com poucos usuários. O programa irá procurar pela foto no arquivo /home/*usuario*/.gnome/photo (sem extensão), onde "usuario" é cada usuário do sistema. A foto deve estar em jpeg, xpm, gif ou png.
  ```
  [greeter]
  Browser=1
  ```
  Dica: o diretório ~/.gnome deve ter permissão de execução e leitura para todos para que o usuário possa configurar sua foto. Por motivos de segurança, essa não é a configuração padrão.

- Permitir que o shutdown seja dado diretamente da tela de login, sem entrar no sistema. Essa opção pode ser útil se você quer permitir que alguém que não tenha a senha de root possa desligar o sistema.
  ```
  [greeter]
  SystemMenu=1
  ```

- Mudar a tty onde o Display Manager roda. Por default, o gdm roda na tty7, mas isso pode ser um problema se você já tiver outro programa rodando na mesma tty, como um getty (terminal). Aliás, se você tiver um getty rodando no mesmo terminal, haverá um conflito com o gdm, que não funcionará corretamente. O teclado pode trancar, por exemplo. Assim, você pode configurar para que o gdm rode em outro terminal (ou fazer com que nenhum outro programa rode na tty7)
  ```
  [servers]
  # para rodar na tty9, descomente a linha abaixo
  # 0=/usr/bin/X11/X vt9
  ```

Um bom guia do GDM foi escrito por Martin Petersen, e pode ser encontrado em:

http://mkp.net/~mkp/gdm/gdm.pdf