
GNU/LINUX

Apostila

27 de novembro de 2003

Copyright (c) 2003 Edson Lima Monteiro. É dada permissão para copiar, distribuir e/ou modificar este documento sob os termos da Licença de Documentação Livre GNU, Versão 1.1 ou qualquer versão posterior publicada pela Free Software Foundation, com as Seções Invariantes sendo todos os capítulos e anexos, com os Textos da Capa da Frente sendo "Instalando o GNU/Linux" e "Apostila", e sem Textos da Quarta-Capa. Uma cópia da licença pode ser obtida em <http://www.gnu.org/licenses/fdl.html>.

Sumário

Introdução	1
1 Comandos básicos	2
1.1 Shell	2
1.2 Comandos do Shell	3
2 Editor vi	4
2.1 Dicas	4
3 Usuários	6
3.1 Usuários e Grupos	6
3.2 Arquivos e suas permissões	6
4 FileSystem	9
4.1 /	9
4.2 /root	10
4.3 /etc	10
4.4 /etc/init.d	10
4.5 /var	10
4.6 /home	10
4.7 /proc	10
4.8 /bin /usr/bin /usr/local/bin /sbin	10
5 Programação Shell	11
5.1 Shell Scripts	11
5.2 Script de Firewall	12
6 Compilando o Kernel	14
6.1 Preparando o sistema	14
6.2 Compilando	15
Bibliografia	21

Lista de Figuras

4.1	Estrutura de Diretórios.	9
6.1	Menu Principal da Compilação do Kernel.	15
6.2	Escolha do processador.	16
6.3	Escolha do processador em detalhe.	17
6.4	Opções de Rede (Network Options).	17
6.5	Opções de Rede - Netfilter - Firewall (Network Options).	18
6.6	Netfilter em detalhe.	18
6.7	Habilitando o Suporte a Rede 10/100 Mbits.	18
6.8	Habilitando alguns tipos de placas de rede.	19
6.9	Escolha dos tipos de Filesystems suportados.	19
6.10	Habilitando a placa de Som	19
6.11	Como seria a configuração em modo texto. Menu principal	20

Lista de Tabelas

1.1	Comandos básicos de shell.	3
2.1	Comandos do editor vi.	5
3.1	Permissões escritas em modo alfabético e número.	7

Introdução

Este texto pretende ser uma introdução aos tópicos principais que iremos abordar num curso de instalação.

Utilização em outros cursos

Esta apostila pode ser usada por outros instrutores como um **roteiro** para o seu curso. O material pode ser adaptado para um curso com outro formato já que código fonte lhe dá esta oportunidade!

Novas Versões

As atualizações poderão ser encontradas em <http://www.linorg.usp.br/docs/curso-instalacao-linux/> . Assim que receber uma versão impressa, verifique no site se é a versão mais nova.

Lista de Discussão

Existe uma lista de discussão para os alunos do curso. Para inscrever-se envie um e-mail para **cursolinuxusp-subscribe@yahoogroups.com**.

A idéia é criar uma comunidade de usuários do GNU/Linux que participaram deste curso.

Licença

Esta apostila é publicada sob a licença LGPL. Você pode obter uma cópia no seguinte site: <http://www.gnu.org/copyleft/lesser.html> .

Agradecimentos

Agradeço a paciência e compreensão de todos os meus alunos. A ajuda deles está sendo muito importante na criação desta apostila.

Edson Lima Monteiro

1

Comandos básicos

Para podermos trabalhar com um sistema GNU/Linux, principalmente na ausência de uma interface gráfica, utilizamos um shell. O shell é um dos meios de obtermos do sistema o que queremos e, para isso, temos de saber alguns comandos.

De nada adiantaria começarmos pela instalação se, ao final, não soubéssemos o que fazer quando entrarmos como usuários no sistema. Por isso, esta sessão é muito importante para começarmos a construir uma base para um bom trabalho.

1.1 Shell

Podemos dividir nossa forma de interagirmos com um sistema computacional em duas formas: gráfica ou através de comandos. A interface gráfica pode ser chamada também de GUI (Graphical User Interface). A interface de comandos pode se chamada de CLI (Comand Line Interface).

O shell é um CLI, isto é, uma interface de comandos entre o usuário e o sistema operacional. Esta interação se dá através dos comandos que o shell reconhece como válidos e é capaz de interpretar ou executar. O shell nada mais é do que um dos programas do seu sistema GNU/Linux.

Os shells mais conhecidos são: sh, csh, tcsh, ksh e bash. O linux utiliza o bash como shell padrão. Os demais podem ser instalados.

Emuladores de Terminais

Um emulador de terminal é um programa que, no modo gráfico (GUI), abre uma janela e disponibiliza um shell. Assim, mesmo no modo gráfico, podemos utilizar uma interface de comandos. No GNU/Linux temos alguns emuladores: xterm, rxvt, konsole.

Terminais Virtuais

Podemos usar uma interface de comandos em um dos seis terminais virtuais que o GNU/Linux disponibiliza. Esses terminais podem ser acessados usando a combinação de teclas Ctrl+Alt+Fn, onde

n pode ir de 1 a 6. A partir de n igual a 7 teremos os terminais gráficos.

1.2 Comandos do Shell

Os principais comandos que devemos conhecer para iniciarmos no Linux são:

Primeiros comandos	
man nome-do-comando	Descobrir o que faz um comando
cp arquivo1 arquivo2	copiar um arquivo1 como ou sobre arquivo2
cp arquivo1 DIRETORIO	copiar um arquivo1 para um diretório
mv arquivo1 arquivo2 / diretorio	move um arquivo ou muda o nome
rm arquivo1	remover o arquivo1
cd diretório de destino	mudar de diretório
touch arquivo1	cria o um arquivo vazio de nome arquivo1
cat arquivo1	mostra na tela o conteúdo do arquivo1
ls	lista os arquivo do diretório em que se está
ls -l	lista os arquivos mostrando suas permissões
ls -a	lista os arquivos escondidos
ls -la	lista todos arquivos e mostra suas opções
grep	procura por expressões regulares
ps opções do comando	lista os processos do sistema
chmod arquivo	mudar as permissões de um arquivo ou diretório
cat arquivo	Exibe o conteúdo do arquivo na tela
more arquivo	Exibe o conteúdo de um arquivo paginando
less arquivo	Exibe o conteúdo de um arquivo paginando
gunzip arquivo.gz	Descompacta arquivos
bunzip2 arquivo.bz2	Descompacta arquivos
ln	Usado para criar links
tar	Usado para empacotar arquivos e/ou diretórios
mkdir	Usado para criar diretórios

Tabela 1.1: Comandos básicos de shell.

2

Editor vi

Para podermos executar as modificações dos arquivos de configuração utilizaremos o editor **vi**. É necessário aprender o **vi** porque muitas vezes não dispomos de uma interface gráfica, principalmente se estivermos em um servidor.

O **vi** possui dois modos de operação: modo de comando e modo de edição. Quando estivermos no modo de edição poderemos digitar o texto como em qualquer outro editor. Quando estivermos no modo de comando teremos que saber os principais comandos para podermos gravar o arquivo, apagar linhas ou apagar caracteres. A principal dificuldade do iniciante é decorar os comandos do **vi**.

Tenha em mente que os comandos do **vi** só funcionam no modo de comando. Ao iniciar a digitação de um novo arquivo, ou editando um já existente, você inicia no modo de comando.

A tabela 2.1 nos dá uma pequena lista de quais comandos utilizar.

2.1 Dicas

No seu diretório "home" crie o arquivo `.virc` ou `.vimrc` com o seguinte conteúdo:

```
:set paste
```

Memorize os principais comandos do vi	
Para ...	Comando
Começar a digitar na posição do cursor	i
Começar a digitar uma posição à direita do cursor	a
Começar a digitar uma linha abaixo da linha do cursor	o
Sair do modo de edição e ir para o modo de comandos	ESC
A partir daqui pressupõe-se que você já digitou ESC	
Apagar um caracter por vez a partir da posição do cursor	x
Apagar uma linha inteira	dd
Copiar uma linha inteira	yy
Colar	p
Sair do arquivo sem salvar as modificações	:q
Gravar as modificações	:w
Gravar as modificações e sair	:wq
Desfazer uma modificações (UNDO)	u
Para mover o cursor para cima	k
Para mover o cursor para cima	Seta para cima
Para mover o cursor para baixo	j
Para mover o cursor para baixo	Seta para baixo
Para mover o cursor para a esquerda	h
Para mover o cursor para a esquerda	Seta para esquerda
Para mover o cursor para a direita	l
Para mover o cursor para a direita	Seta para a direita

Tabela 2.1: Comandos do editor vi.

3

Usuários

3.1 Usuários e Grupos

Para usarmos um sistema GNU/Linux devemos entrar no sistema como um usuário. Mas precisamos saber que existem categorias de usuários. Eles são distinguidos pelo nível de suas permissões dentro do sistema.

O usuário root é o super-usuário porque tem todos os privilégios. Tem acesso total a todos os arquivos do sistema. Em contraste, temos uma conta de usuário sem privilégios especiais para trabalharmos. É com essa conta que faremos quase tudo o que precisamos no sistema, utilizando o root apenas em tarefas administrativas, isto é, instalação, remoção ou configuração de programas.

Além disso, cada usuário no sistema tem um grupo. O grupo existe para que diferentes pessoas consigam compartilhar seus arquivos entre si, evitando que, pessoas de outros grupos, tenham acesso a eles.

Quando um usuário é acrescentado ao sistema também é criado um grupo com seu nome. Isto pode ser alterado, tornando-o membro do grupo "users", por exemplo, desde a sua criação no sistema. Quando formos criar um usuário de modo que ele já pertença a um grupo específico, este deve existir antes da criação do usuário.

Arquivos: /etc/passwd, /etc/shadow , /etc/group , /etc/skel/.bashrc /etc/profile

Comandos: vipw, vigr, useradd, groupadd, adduser, deluser e userdel

3.2 Arquivos e suas permissões

Ao trabalharmos com os arquivos, sejam eles criados por nós, ou criados pelo sistema ao instalar um pacote, precisamos saber a quem pertence e quais suas permissões.

As permissões de que falamos são: leitura (read), escrita (write) e execução (execution). Além disso, tem acesso ao arquivo o seu dono, um grupo de usuários ou todos os usuários do sistema.

Vamos analisar as permissões do arquivo que contém a mensagem que aparece assim que um usuário entra no sistema (no modo texto).

```
boni@debian:~$ ls -l /etc/motd
-rw-r--r--    1 root    root          370 Ago  1 14:53 /etc/motd
```

A primeira trinca (rwX) refere-se às permissões do dono do arquivo, o usuário root, que pode ler, escrever e executar. A segunda trinca (r-x) permite ao grupo root ler e executar. A terceira trinca (r-x) permite a todos os outros usuários do sistema ler e executar.

Podemos mudar as permissões de um arquivo, desde que tenhamos permissão para isso. Podemos usar o sistema numérico ou o alfabético para fazê-lo. Nesta apostila vamos aprender o numérico.

A cada permissão é associado um valor de acordo com a tabela 3.1 :

Ler (read)	r	4
Escrever (write)	w	2
Executar (execute)	x	1

Tabela 3.1: Permissões escritas em modo alfabético e numérico.

A permissão é construída somando-se os valores numéricos correspondentes à permissão que queremos dar ao arquivo.

Vamos copiar o arquivo para o diretório temporário e mudar suas permissões com o comando **chmod**.

```
boni@dsu61:/tmp$ ls -l /bin/date
-rwxr-xr-x    1 root    root          25820 Jul 26  2001 /bin/date
```

```
boni@debian:~$ cp /bin/date /tmp/
```

```
boni@debian:~$ cd /tmp
```

```
boni@debian:/tmp$ ls -l date
-rwxr-xr-x    1 boni    boni          25820 Ago  3 16:51 date
```

```
boni@debian:/tmp$ ./date
```

```
boni@debian:/tmp$ chmod 777 gzip
```

```
boni@debian:/tmp$ ls -l date
-rwxrwxrwx    1 boni    boni          25820 Ago  3 16:51 date
```

```
boni@debian:/tmp$ ./date
```

```
boni@debian:/tmp$ chmod 444 date
```

```
boni@debian:/tmp$ ls -l date
-r--r--r--    1 boni    boni          25820 Ago  3 16:51 date
```

```
boni@dsu61:/tmp$ ./date
```

```
bash: ./date: Permissão negada
```

```
boni@dsu61:/tmp$ date
```

```
Dom Ago  3 16:54:46 BRT 2003
```


4

FileSystem

Vamos conhecer alguns diretórios do sistema operacional GNU/Debian.

Utilize o *konqueror* para navegar pelos diretórios do sistema. Ele também pode funcionar como Navegador WEB, mas eu prefiro utilizá-lo como "navegador de diretórios".



Figura 4.1: Estrutura de Diretórios.

4.1 /

Este é o diretório raiz. Todos os outros diretórios estão, logicamente, abaixo dele. Isto é, mesmo que existam as partições `/`, `/home/`, `/var` e `/tmp` no disco rígido, por exemplo, estas partições sempre estarão "abaixo" do diretório raiz (`/`).

4.2 /root

É o diretório do usuário root.

4.3 /etc

Neste diretório ficam todos os arquivos de configuração de programas instalados.

4.4 /etc/init.d

Scripts que iniciam serviços durante o boot do sistema estão aqui.

4.5 /var

Os arquivos de *logs* do sistema ficam aqui. Principalmente no diretório /var/log.

4.6 /home

Os diretórios de usuários ficam aqui.

4.7 /proc

Os arquivos deste diretório estão, na verdade, na memória RAM. Alguns arquivos úteis são: cpuinfo, pci, interrupts, ioports e filesystems.

4.8 /bin /usr/bin /usr/local/bin /sbin ...

Os comandos e binários do sistema encontram-se nestes diretórios. Por isso a variável de ambiente PATH precisar contê-los.

5

Programação Shell

5.1 Shell Scripts

Shell scripts nada mais são do que arquivos texto com uma sequência de comandos que devem ser executados na ordem em que aparecem.

O administrador de sistema recorre frequentemente ao uso de shell scripts pela facilidade em escrevê-los. Podemos criar scripts muito simples com apenas algumas linhas de comando mas que, ainda assim, são capazes de fazer coisas importantes.

Precisamos saber quatro elementos básicos para começarmos a aprender a escrever shell scripts:

- O arquivo de shell script deve ser executável, isto é, dever ter permissão de letra x.
- A primeira linha diz quem vai executar os comandos constantes do arquivo.
- O símbolo `#` representa um comentário, isto é, que a linha não deve ser executada. Isto só não vale para a primeira linha do arquivo.
- A primeira linha do arquivo não pode ser vazia.

Como exemplo, vamos escrever o shell script `hello.sh`¹.

```
#!/bin/bash

#primeiro comando
echo "Oi. Este é meu primeiro shell script! "
```



```
#segundo comando
date
```

Depois de editar o arquivo no `vi`, faça:

```
$ chmod 750 hello.sh

$ ./hello.sh
Oi. Este é meu primeiro shell script!
Dom Ago  3 17:15:11 BRT 2003
```

¹As extensões dos arquivos servem para termos uma idéia do tipo de arquivo. Não precisamos chamar nosso script de `hello.exe` para executá-lo.

Agora, vamos aprender a usar variáveis em nosso shell script.

Copie o `hello.sh` para `hello2.sh` e edite-o com o `vi`.

```
# cp hello.sh hello2.sh

-----
O hello2.sh deve ficar assim:

#!/bin/bash

# primeira variavel
UNIDADE=CCE

# segunda variavel
NOME="Edson Lima Monteiro"

# primeiro comando
echo "Oi. Este é meu primeiro shell script! "

# segundo comando
echo "Eu sou $NOME e trabalho no $UNIDADE "

# terceiro comando
date

Saia do vi.

$ ./hello2.sh
Oi. Este é meu primeiro shell script!
Eu sou Edson Lima Monteiro e trabalho no CCE
Dom Ago  3 17:23:05 BRT 2003
```

Agora que conhecemos o conceito de variáveis em shell scripts, vamos criar a variável **EDITOR=vi** no arquivo **.bashrc**. Ela será importante quando formos utilizar o comando **cron**.

É importante lembrar que podemos usar os comandos **for**, **if**, **while** em shell scripts. O uso dessas expressões está fora do escopo do nosso curso.

5.2 Script de Firewall

Agora vamos criar um shell script que irá fazer a proteção de nosso sistema GNU/Linux. É um exemplo para as portas que poderemos abrir durante o curso. Ele não irá garantir a segurança de outro sistema que esteja em produção!

Com os elementos básicos que aprendemos na seção anterior, somos capazes de criar scripts maiores e mais importantes. O que falta é aprender os comandos que queremos colocar dentro dos scripts.

Nosso script está usando o comando **iptables** que será explicado durante a aula.

```
#!/bin/sh
#
```

```
IPTA="/sbin/iptables"

#Limpar as regras existentes
/sbin/iptables -F

#LOCAL HOST
LH=192.168.1.2

#####
#impressora
$IPTA -A INPUT -p tcp -s 0/0 -d $LH/32 --destination-port 515 -j DROP
#####

#####
# ssh
$IPTA -A INPUT -p tcp -s 192.168.1.3 -d $LH/32 --destination-port 22 -j ACCEPT
$IPTA -A INPUT -p tcp -s 0/0 -d $LH/32 --destination-port 22 -j DROP
#####

#####
# ftp
$IPTA -A INPUT -p tcp -s 192.168.1.4 -d $LH/32 --destination-port 21 -j ACCEPT
$IPTA -A INPUT -p tcp -s 0/0 -d $LH/32 --destination-port 21 -j DROP
#####

#####
# apache
$IPTA -A INPUT -p tcp -s $LH/32 -d $LH/32 --destination-port 80 -j ACCEPT
$IPTA -A INPUT -p tcp -s 0/0 -d $LH/32 --destination-port 80 -j DROP
#####

#####
# X WINDOW
$IPTA -A INPUT -p tcp -s 0/0 -d $LH/32 --destination-port 6000 -j DROP
$IPTA -A INPUT -p udp -s 0/0 -d $LH/32 --destination-port 6000 -j DROP
$IPTA -A INPUT -p tcp -s 0/0 -d $LH/32 --destination-port 6001 -j DROP
$IPTA -A INPUT -p udp -s 0/0 -d $LH/32 --destination-port 6001 -j DROP
#####
```

6

Compilando o Kernel

O kernel é a parte do sistema operacional que sequer vemos ao utilizarmos o sistema. Ele nada mais é do que um programa que aloca/disponibiliza o hardware da máquina para a utilização por outros programas. É um programa que possibilita que todo o resto funcione em harmonia utilizando o mesmo hardware.

O kernel é controlado por Linus Torvalds e, a atual versão estável, está sob o comando do brasileiro Marcelo Tosati, funcionário da Cyclades.

A numeração do kernel tem um padrão, onde o número intermediário é que nos diz se estamos com uma versão estável ou de teste. Números pares são para as versão estáveis e, ímpares, para as versões de teste. Atualmente a versão estável é a 2.4.22 e essa que vamos compilar.

Este procedimento é válido apenas até série 2.4 do kernel. Para a nova série 2.6.x os passos mudam.

6.1 Preparando o sistema

Entre novamente no ambiente gráfico como usuário root, abra um terminal e siga os passos abaixo. Eles só serão feitos a primeira vez em que você for compilar o kernel!

Antes iniciarmos a compilação, vamos garantir que temos todos os pacotes necessários para o mesmo. Em outras distribuições, esses pacotes são classificados em "pacotes de desenvolvimento" ou Kernel. Vamos instalar alguns deles:

```
# apt-get install gcc bin86 make automake autoconf wget

# apt-get install tk8.3 cpio bc libc6-dev bzip2 libncurses5-dev

# cd /usr/src

# wget http://www.linorg.usp.br/kernel/v2.4/linux-2.4.22.tar.bz2

# bunzip2 linux-2.4.22.tar.bz2

# tar -xvf linux-2.4.22.tar

# ln -s /usr/src/linux-2.4.22 linux

# cd /usr/include

# mv asm asm.old

# mv scsi scsi.old
```

```
# mv linux linux.old
# ln -s /usr/src/linux/include/asm-i386 asm
# ln -s /usr/src/linux/include/scsi scsi
# ln -s /usr/src/linux/include/linux linux
```

6.2 Compilando

Esta parte será feita todas as vezes que desejar compilar um novo kernel da série 2.4.x.

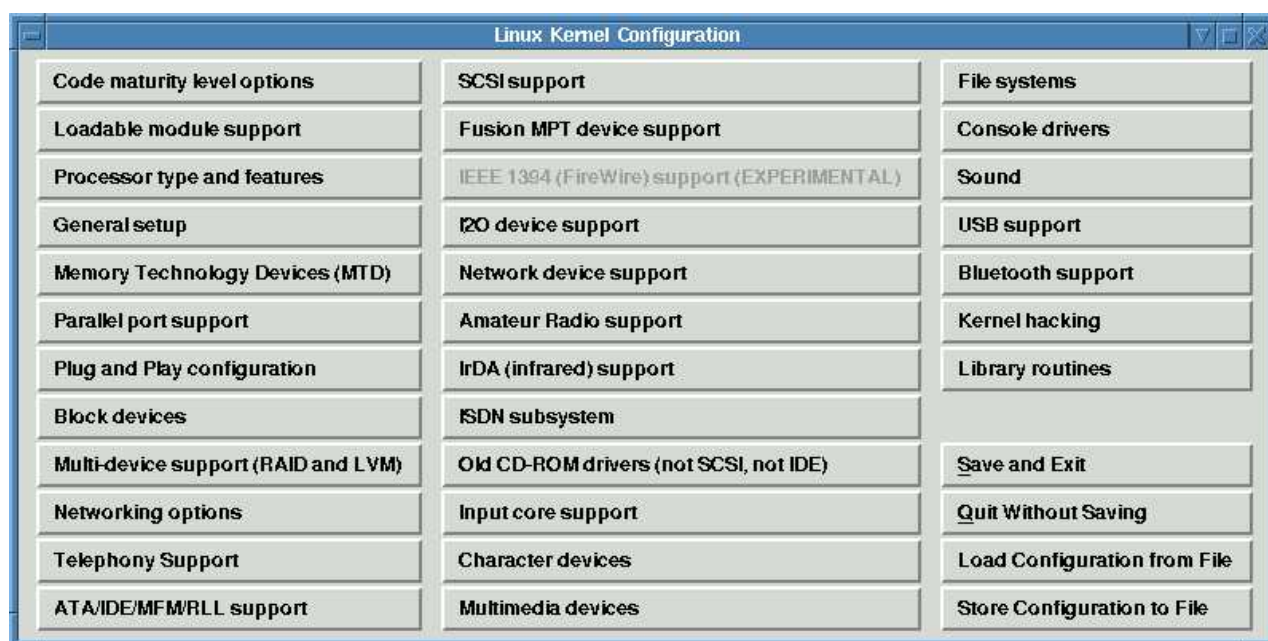


Figura 6.1: Menu Principal da Compilação do Kernel.

Verifique se o link linux aponta para o fonte do kernel desejado antes de iniciar.

```
# cd /usr/src
# ls -ld linux
# cd /usr/src/linux
# make mrproper
```

No próximo item, siga as orientações do instrutor para a configuração do processador, file-systems, network devices, network options, sound e etc.

```
# make xconfig
# make dep
```

```
# make bzImage

# make modules

# make modules_install

# cp arch/i386/boot/bzImage /boot/vmlinuz-2.4.22
```

O arquivo `/etc/lilo.conf` original tem um trecho assim:

```
Boot up Linux by default.  default=Linux image=/vmlinuz labo
```

```
# vi /etc/lilo.conf

# lilo

# reboot
```

Se tudo funcionou:

```
# cp /usr/src/linux/System.map /boot/System.map

# reboot
```

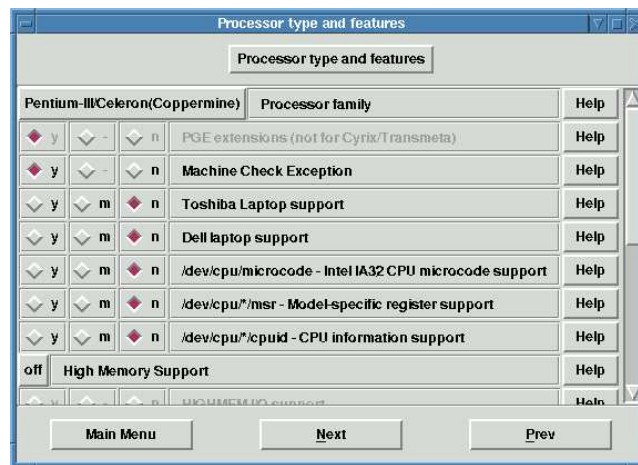


Figura 6.2: Escolha do processador.



Figura 6.3: Escolha do processador em detalhe.

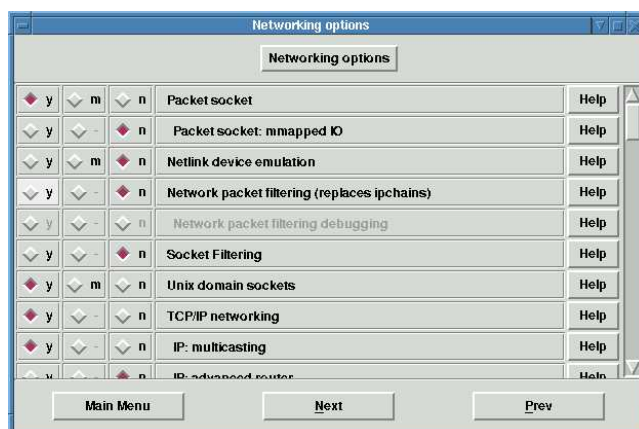


Figura 6.4: Opções de Rede (Network Options).

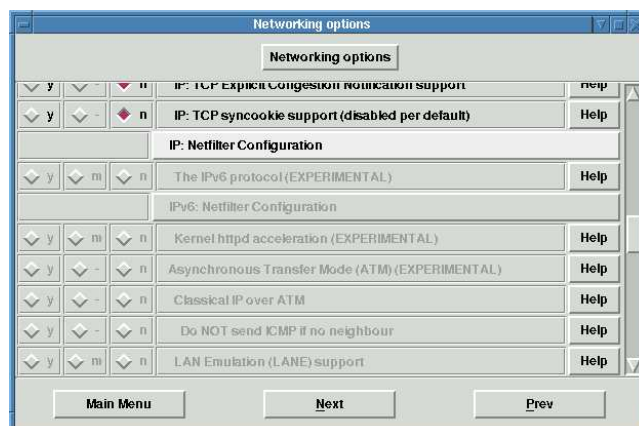


Figura 6.5: Opções de Rede - Netfilter - Firewall (Network Options).



Figura 6.6: Netfilter em detalhe.

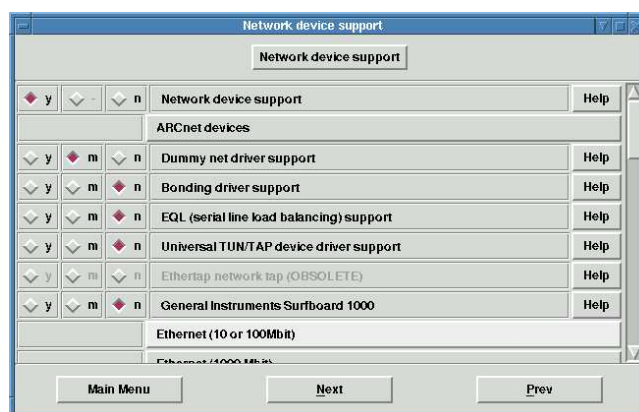


Figura 6.7: Habilitando o Suporte a Rede 10/100 Mbits.

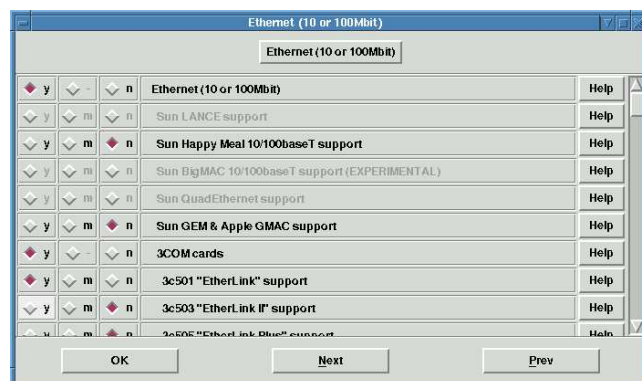


Figura 6.8: Habilitando alguns tipos de placas de rede.

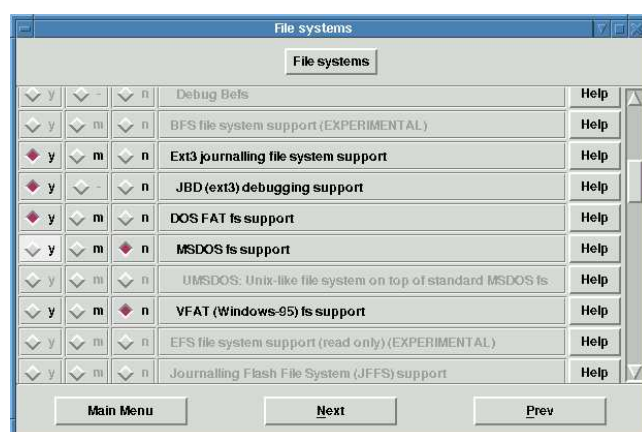


Figura 6.9: Escolha dos tipos de Filesystems suportados.

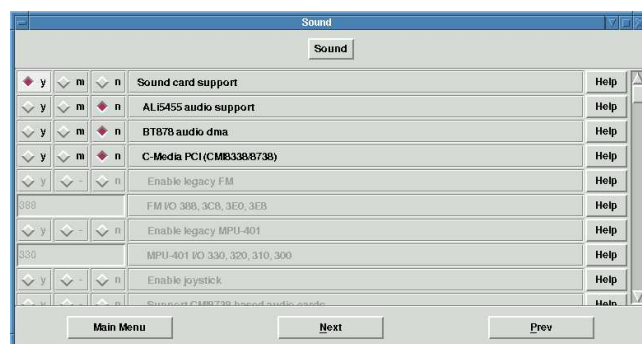


Figura 6.10: Habilitando a placa de Som

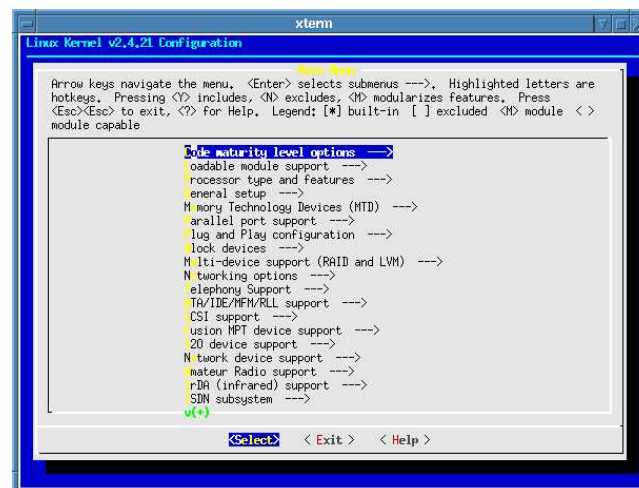


Figura 6.11: Como seria a configuração em modo texto. Menu principal

