4a. Parte:

Ajustes & Configurações

✔ *Copyright (c) 2002-2008 –Ednei Pacheco de Melo.*

Permission is granted to copy, distribute and/or modify this document under the terms of the *GNU Free Documentation License*, version *1.1* or any later version published by the *Free Software Foundation*; a copy of the license is included in the section entitled "*GNU Free Documentation License*".

Índice

Visão geral 6

I. Vídeo – placa, monitor e modo gráfico 7

Introdução *7*

Preparativos iniciais *7*

Detecção do chipset para o vídeo 7

Processo básico de configuração *8*

xorgconfig 8

Instruções básicas *8*

Mouse *8*

Teclado *10*

Monitor *11*

Placa de vídeo *14*

Memória da placa de vídeo *16*

Descrição do periférico *17*

Resoluções e profundidade de cor *17*

Finalizando *19*

Outras ferramentas 20

As aceleradoras gráficas 3D *21*

O servidor gráfico e as aceleradoras 21

Preparativos necessários 22

GLX / DRI *22*

Suporte ao barramento AGP *23*

Drivers proprietários 23

ATI *23*

nVidia *24*

Sobre o X.org *24*

Sobre os drivers VESA *25*

Conclusão *25*

II. Audio – placa de som 27

Introdução *27*

Preparativos iniciais *27*

Detecção do chipset para o som 27

Checando o carregamento dos módulos 28

Definindo as permissões de acesso 28

Os drivers ALSA *28*

Configurando a placa de som 29

Ajustes finais (mixer) 30

Os devices 31

Problemas mais freqüentes 31

Sobre os antigos drivers OSS *32*

Conclusão *32*

III. Modem – placa de fax-modem 34

Introdução ***34***

Preparativos iniciais ***34***

Detecção do chipset para o fax-modem 34

O código-fonte do kernel 35

Atalho e permissões de acesso 36

Processo genérico de configuração ***36***

Softmodens 36

Hardmodens 37

Preparativos finais ***38***

Carregamento de módulos para a discagem 38

Configuração e inicialização do discador 38

Conflitos de IRQs e portas seriais ***44***

Configuração padrão das portas seriais 44

Setserial 45

Sobre os softmodens ***46***

Conclusão ***47***

IV. Unidades de armazenamento 48

Introdução ***48***

Discos rígidos ***48***

HDParm 48

Obtendo informações sobre o disco rígido *49*

Mão na massa *50*

Otimizações finais *56*

Parâmetros opcionais *57*

Observações importantes *57*

Interfaces gráficas *58*

Agradecimentos *58*

SMART 59

smartctl *59*

smartd *60*

Gravadores de CD/DVDs ***61***

Reconhecimento do sistema 61

Permissões de acesso 61

Memórias eletrônicas ***62***

Gerenciamento de unidades e partições ***63***

Os arquivos de configuração 63

/etc/fstab *63*

/etc/mtab *65*

Conclusão ***66***

V. Teclado e mouse 67

Introdução ***67***

O teclado ***67***

Configurações básicas 67

Modo gráfico....67
Modo texto....68
Intervenções mais freqüentes....68
Habilitando as famosas "teclas do Windows"....68
Aumentando a velocidade....69
***O mouse*....69**
O GPM....69
Configurações básicas....70
Modo gráfico....70
Modo texto....71
Intervenções mais freqüentes....72
Mouses ópticos....72
... e com scroll lock (com rodinhas)....72
Com conexão USB....72
***Conclusão*....73**
VI. O kernel Linux....74
***Introdução*....74**
***Considerações básicas*....74**
As particularidades do kernel....74
O sistema de controle de versões....74
A localização de seus componentes....75
A equipe de desenvolvimento....77
***Os módulos do kernel*....77**
A manutenção....77
lsmod....77
modprobe....79
insmod....79
depmod....80
modinfo....80
rmmod....80
O manuseio de módulos pré-compilados....81
A localização dos arquivos....81
Observações finais....82
***O processo de compilação*....82**
Sobre o processo de compilação....82
O principal motivo....82
Entre outros aspectos interessantes....83
Os preparativos iniciais....83
Cópia de segurança dos dados do sistema....84
A elaboração de pendrives de inicialização....84
Cópia de segurança da configuração anterior....85
Iniciando os procedimentos....85
Atendendo os requerimentos básicos....85
Limpando as definições prévias....85
Inicializando as interfaces de interação....86
Sobre a habilitação de parâmetros....88
As classes de atributos....88
Realizando a compilação....89
Salvando as alterações realizadas....89
Realizando a checagem das pendências....89

Construindo a imagem compactada do kernel.....*90*
Realizando a compilação dos módulos.....*90*
Os preparativos finais.....91
Recomendações gerais.....91
Conclusão*.....*92

VISÃO GERAL

Enfim, chegamos a uma das etapas mais importantes e vitais para nossa satisfação na administração e uso de sistemas *GNU/Linux*: a realização de ajustes e configurações dos periféricos e parâmetros do sistema.

A configuração geral de um computador providos de sistemas *GNU/Linux* é uma das etapas mais delicadas, a qual tem sido motivo de decepção e insatisfação de diversos usuários. Diferente do *Windows* e de acordo com a distribuição em uso – em especial, o *Slackware* –, necessitaremos realizar a configuração manual de diversos programas e periféricos, o que o torna um verdadeiro obstáculo para os iniciantes. Muitos usuários por não conhecerem profundamente o seu funcionamento interno, desistem da sua desinstalação, deixando de experimentar uma excelente opção de sistema operacional. Para agravar ainda mais a situação, existem poucos bons materiais técnicos e didáticos específicos, que cobrem com perfeição diversos aspectos – mesmo apesar de serem inúmeros...

No caso do *Slackware*, mesmo sendo a distribuição mais "*genérica*" existente, muitos de seus processos são diferenciados das outras demais, em "*virtude*" das personalizações destas últimas. Pelo fato desta não possuir determinados utilitários gráficos ou ferramentas de automatização, o processo se torna um pouco complexo e detalhado, pois o usuário necessitará obter certos conhecimentos para que possa proceder com as intervenções e dependendo da complexidade, lidar com processos mais elaborados, tais como a compilação de *drivers*, edição manual da configuração, recompilações do *kernel*, etc. Em contrapartida, uma vez bem configurado, o *Slackware* tende a ter um perfeito funcionamento:

> *"Por sinal, o Slackware é uma das distribuições que se sai melhor neste aspecto: o sistema pode ser mais difícil de configurar, mas pelo menos tudo funciona como deveria". -- [Livro "Entendendo e Dominando o Linux" - 5a. Edição, por Carlos E. Morimoto].*

Isto também deve-se ao fato da distribuição utilizar o mínimo de processos de automatização, onde a ocorrência de erros são mínimas:

> *"Ao contrário de várias outras distribuições, processos de autodetecção que podem 'travar' a máquina não são aplicados, assim não se corre o risco de parar a instalação por culpa de uma placa de vídeo não suportada, ou de uma placa de som mal humorada. Aliás, a detecção da placa de rede (...) também pode ser deixada para depois, facilitando a vida de quem faz as instalações em uma máquina e depois vai passar para outra." -- ["Por quê usar o Slackware", por Piter Punk].*

Em virtude do grande número de periféricos e dispositivos existentes, será praticamente impossível descrever os processos de configuração para cada um deles; porém, veremos as principais características, funcionalidades e requerimentos necessários para realização de procedimentos. &;-D

I. Vídeo – placa, monitor e modo gráfico

Introdução

A configuração da placa de vídeo e conseqüentemente do modo gráfico, além dos demais parâmetros, é um processo mais tranqüilo, graças aos excelentes utilitários disponíveis. Em especial, destaca-se o servidor integrado que provê o subsistema gráfico de vídeo, o *X.org*.

Neste capítulo, iremos conhecê-lo mais detalhadamente.

Preparativos iniciais

Detecção do chipset para o vídeo

Para obtermos as informações sobre as referências técnicas da placa ou do *chipset* de vídeo *onboard* em uso, deveremos utilizar o comando *lspci*, combinado-o com a palavra-chave *VGA*:

```
# lspci -v | grep -i VGA
01:00.0 VGA compatible controller: VIA Technologies, Inc. VT8378 [S3
UniChrome] Integrated Video (rev 01) (prog-if 00 [VGA])
# _
```

Sem maiores dificuldades, vemos que as informações acima referem-se as especificações do vídeo *onboard VT8378* (*S3 UniChrome*, dotado de aceleração *3D*), localizado no *chipset* ponte norte *KM400*, que por sua vez é fabricado pela *Via Tecnologies* e utilizado em diversas placas-mãe.

Para obterem informações mais detalhadas, poderemos utilizar...

```
# lspci -v
```

... e buscar o trecho referente ao sistema de vídeo...

```
...
01:00.0 VGA compatible controller: VIA Technologies, Inc. VT8378 [S3
UniChrome] Integrated Video (rev 01) (prog-if 00 [VGA])
        Subsystem: ABIT Computer Corp. Unknown device 140c
        Flags: bus master, 66MHz, medium devsel, latency 32, IRQ 11
        Memory at e0000000 (32-bit, prefetchable) [size=64M]
        Memory at e4000000 (32-bit, non-prefetchable) [size=16M]
        Expansion ROM at <unassigned> [disabled] [size=64K]
        Capabilities: [60] Power Management version 2
        Capabilities: [70] AGP version 2.0
...
```

Claramente observamos a referência do fabricante da placa-mãe (*Abit Computer Corp.*) e o tipo de barramento suportado (*AGP 2.0*), além de

outras informações menos preponderantes.

Processo básico de configuração

XORGCONFIG

Simples, rápido e extremamente eficiente, o *xorgconfig* é um utilitário desenvolvido pelo projeto *X.org* que auxilia na realização das configurações gerais do modo gráfico do sistema. Ao ser executado, o utilitário constrói o arquivo de configuração baseado em perguntas realizadas por ele e respondidas pelo usuário.

Segue abaixo a descrição passo-a-passo do processo básico de configuração do modo gráfico. Conforme informado, basta digitar na linha de comando...

```
$ xorgconfig
```

Instruções básicas

Após inicializar o utilitário, serão exibidas as seguintes informações:

```
This program will create a basic xorg.conf file, based on menu selections
you make.  It will ask for a pathname when it is ready to write the file.

The xorg.conf file usually resides in /etc/X11 or /usr/etc/X11.  If
no xorg.conf file is present there, Xorg will probe the system to
autoconfigure itself.  You can run Xorg -configure to generate a xorg.conf
file based on the results of autoconfiguration, or let this program
produce a base xorg.conf file for your configuration, and fine-tune it.
A sample xorg.conf file is also supplied with Xorg; it is configured
for a standard VGA card and monitor with 640x480 resolution.

There are also many chipset and card-specific options and settings available,
but this program does not know about these. On some configurations some of
these settings must be specified. Refer to the X driver man pages and the
chipset-specific READMEs in /usr/lib/X11/doc for further details.

Before continuing with this program, make sure you know what video card
you have, and preferably also the chipset it uses and the amount of video
memory on your video card, as well as the specifications of your monitor.

Press enter to continue, or ctrl-c to abort.
```

Conforme as instruções do utilitário, teclamos <ENTER> e seguimos adiante para a próxima etapa.

Mouse

```
First specify a mouse protocol type. Choose one from the following list:

 1.  Auto
```

```
 2.  SysMouse
 3.  MouseSystems
 4.  PS/2
 5.  Microsoft
 6.  Busmouse
 7.  IMPS/2
 8.  ExplorerPS/2
 9.  GlidePointPS/2
10.  MouseManPlusPS/2
11.  NetMousePS/2
12.  NetScrollPS/2
13.  ThinkingMousePS/2
14.  AceCad

The recommended protocol is Auto. If you have a very old mouse
or don't want OS support or auto detection, and you have a two-button
or three-button serial mouse, it is most likely of type Microsoft.

Enter a protocol number:
```

Nesta seção devemos informar ao utilitário o protocolo do *mouse* utilizado neste computador. Em nosso caso estamos utilizamos um *mouse PS/2* de *3* botões com rodinhas (*scroll*), conforme as instruções da embalagem, é compatível com o *mouse* da *Microsoft*. Escolheremos então *7* + <ENTER>.

Enter a protocol number: 7

```
If your mouse has only two buttons, it is recommended that you enable
Emulate3Buttons.

Please answer the following question with either 'y' or 'n'.
Do you want to enable Emulate3Buttons? _
```

O utilitário questiona ao usuário se este *mouse* possui dois botões. Ele pergunta se deseja emular o *3o.* Botão, para caso o *mouse* não possua. A funcionalidade do *3o.* botão é habilitada quando pressionamos os dois botões do *mouse* juntos, muito útil em diversas aplicações. Como possuímos este *mouse*, esta emulação é desnecessária, então apenas digitaremos *n* + <ENTER> e continuaremos com o processo de configuração.

```
Do you want to enable Emulate3Buttons? N

Now give the full device name that the mouse is connected to, for example
/dev/tty00. Just pressing enter will use the default, /dev/input/mice.

Mouse device:_
```

Informe em qual o dispositivo *device* correspondente se encontra conectado o *mouse*. Por opção padrão, utilizaremos o *device /dev/input/mice* indicado pelo utilitário, bastando apenas pressionar <ENTER>.

Teclado

```
Please select one of the following keyboard types that is the better
description of your keyboard. If nothing really matches,
choose "Generic 104-key PC"

  1  Generic 101-key PC
  2  Generic 102-key (Intl) PC
  3  Generic 104-key PC
  4  Generic 105-key (Intl) PC
  5  Dell 101-key PC
  6  Dell Latitude series laptop
  7  Everex STEPnote
  8  Keytronic FlexPro
  9  Microsoft Natural
 10  Northgate OmniKey 101
 11  Winbook Model XP5
 12  Japanese 106-key
 13  PC-98xx Series
 14  A4Tech KB-21
 15  A4Tech KBS-8
 16  A4Tech Wireless Desktop RFKB-23

Enter a number to choose the keyboard.

Press enter for the next page
_
```

Agora o utilitário solicita-nos informar qual o mapa do teclado utilizado neste computador. Não confunda com o mapa do teclado carregado durante a inicialização do sistema. Em nosso caso, pressionaremos <ENTER>...

```
Please select one of the following keyboard types that is the better
description of your keyboard. If nothing really matches,
choose "Generic 104-key PC"

 17  Brazilian ABNT2
 18  Acer AirKey V
 19  ACPI Standard
 20  Azona RF2300 wireless Internet Keyboard
 21  Advance Scorpius KI
 22  Brother Internet Keyboard
 23  BTC 5113RF Multimedia
 24  BTC 5126T
 25  BTC 9000
 26  BTC 9000A
 27  BTC 9001AH
 28  BTC 5090
 29  BTC 9019U
 30  Cherry Blue Line CyBo@rd
 31  Cherry CyMotion Master XPress
 32  Cherry Blue Line CyBo@rd (alternate option)

Enter a number to choose the keyboard.
```

```
Press enter for the next page
_
```

... para visualizar a listagem seguinte. Aí sim, visualizaremos a nossa opção (teclado *Brazilian ABNT2*) e digitaremos *17* + <ENTER>.

À seguir, deveremos definir o país (*country*):

```
  1  U.S. English
  2  Andorra
  3  Afghanistan
  4  Arabic
  5  Albania
  6  Armenia
  7  Azerbaijan
  8  Belarus
  9  Belgium
 10  Bangladesh
 11  India
 12  Bosnia and Herzegovina
 13  Brazil
 14  Bulgaria
 15  Myanmar
 16  Canada
 17  Congo, Democratic Republic of the
 18  Croatia

Enter a number to choose the country.
Press enter for the next page
_
```

Neste caso basta apenas digitar o valor referente ao nosso país, no caso *Brasil*. sem muito mistério, digitem *13* + <ENTER>.

```
Please enter a variant name for 'br' layout. Or just press enter
for default variant
_
```

A questão acima mencionada é recomendada para a utilização de um teclado variante do padrão oficial. Como no *Brasil* utilizamos o padrão *ABNT-2*, bastam apenas que teclem <ENTER>.

```
Please answer the following question with either 'y' or 'n'.
```

Do you want to select additional XKB options (group switcher,

group indicator, etc.)?

Esta opção será para criar um novo grupo com permissões inerentes a configuração do teclado. Apenas digitem *n* + <ENTER>.

Monitor

O *xorgconfig* agora nos alerta sobre a seriedade dos próximos parâmetros que iremos definir para a correta configuração do nosso *monitor*.

```
Now we want to set the specifications of the monitor. The two critical
```

```
parameters are the vertical refresh rate, which is the rate at which the
the whole screen is refreshed, and most importantly the horizontal sync rate
which is the rate at which scanlines are displayed.

The valid range for horizontal sync and vertical sync should be documented
in the manual of your monitor. If in doubt, check the monitor database
/usr/X11R6/lib/X11/doc/Monitors to see if your monitor is there.

Press enter to continue, or ctrl-c to abort.
```

Esta é a fase mais crítica da configuração, pois caso as freqüência seja informada incorretamente, haverá o risco de danos ao *monitor*. deveremos ter certeza de que estamos utilizando todos os parâmetros corretos, caso contrário poderemos perder um caro componente do computador.

Para dar seqüência ao processo, apenas pressionem <ENTER>.

```
You must indicate the horizontal sync range of your monitor. You can either
select one of the predefined ranges below that correspond to industry-
standard monitor types, or give a specific range.

It is VERY IMPORTANT that you do not specify a monitor type with a horizontal
sync range that is beyond the capabilities of your monitor. If in doubt,
choose a conservative setting.

    hsync in kHz; monitor type with characteristic modes
 1  31.5; Standard VGA, 640x480 @ 60 Hz
 2  31.5 - 35.1; Super VGA, 800x600 @ 56 Hz
 3  31.5, 35.5; 8514 Compatible, 1024x768 @ 87 Hz interlaced (no 800x600)
 4  31.5, 35.15, 35.5; Super VGA, 1024x768 @ 87 Hz interlaced, 800x600 @ 56 Hz
 5  31.5 - 37.9; Extended Super VGA, 800x600 @ 60 Hz, 640x480 @ 72 Hz
 6  31.5 - 48.5; Non-Interlaced SVGA, 1024x768 @ 60 Hz, 800x600 @ 72 Hz
 7  31.5 - 57.0; High Frequency SVGA, 1024x768 @ 70 Hz
 8  31.5 - 64.3; Monitor that can do 1280x1024 @ 60 Hz
 9  31.5 - 79.0; Monitor that can do 1280x1024 @ 74 Hz
10  31.5 - 82.0; Monitor that can do 1280x1024 @ 76 Hz
11  Enter your own horizontal sync range

Enter your choice (1-11): _
```

Aqui devemos consultar o manual do *monitor* de vídeo e procurar suas definições de freqüência da varredura horizontal. Geralmente essas configurações estão situadas em capítulos ou seções com o título *Especificações Técnicas*. Caso tenhamos a infelicidade de não possuir a disposição o manual do equipamento, basta virar o *monitor* de costas: certamente que atrás dele deverá constar suas especificações técnicas.

Se as freqüências do utilitário não se enquadrarem as especificações do equipamento em uso, existe a opção *11* do *xorgconfig* que permitirá lançar manualmente a freqüência correta. Por utilizarmos um *monitor* de *15"* fabricado pela *LG*, modelo *Flatron EZ T530S*, que possui a freqüência horizontal de *30* a *56 Hz*, deveremos então digitar *11* + <ENTER>.

```
Enter your choice (1-11): 11
```

```
Please enter the horizontal sync range of your monitor, in the format used
in the table of monitor types above. You can either specify one or more
continuous ranges (e.g. 15-25, 30-50), or one or more fixed sync frequencies.

Horizontal sync range: _
```

Agora é só informar a freqüência de acordo com o formato solicitado pelo utilitário. No caso do *monitor* acima referido, será *30-56* + <ENTER>.

```
Horizontal sync range: 30-56

You must indicate the vertical sync range of your monitor. You can either
select one of the predefined ranges below that correspond to industry-
standard monitor types, or give a specific range. For interlaced modes,
the number that counts is the high one (e.g. 87 Hz rather than 43 Hz).

 1  50-70
 2  50-90
 3  50-100
 4  40-150
 5  Enter your own vertical sync range

Enter your choice: _
```

O mesmo vale para a freqüência de varredura vertical. Também existe uma opção que nos permite lançar a freqüência vertical do nosso *monitor* - a *5*. Neste caso, digitem *5* + <ENTER> se as freqüências do utilitário não se enquadrarem com as freqüências disponibilizadas.

```
Enter your choice: 5

Vertical sync range: _
```

Da mesma forma que na freqüência horizontal, informar ao utilitário a freqüência vertical do *monitor* de acordo com o formato solicitado (o mesmo formato da freqüência horizontal). Em nosso caso, o *monitor* utiliza os valores de *50* a *120 Hz*. Informaremos estes valores e teclem <ENTER>:

```
Vertical sync range: 50-120

You must now enter a few identification/description strings, namely an
identifier, a vendor name, and a model name. Just pressing enter will fill
in default names.

The strings are free-form, spaces are allowed.
Enter an identifier for your monitor definition: _
```

Agora é só informar a nomenclatura de identificação do *monitor*. Se desejarmos, poderemos apenas pressionar <ENTER> e continuar com o processo de configuração, já que estas definições são opcionais. Em nosso caso digitamos o texto *LG Flatron EZ T530S*.

Placa de vídeo

```
Now we must configure video card specific settings. At this point you can
choose to make a selection out of a database of video card definitions.
Because there can be variation in Ramdacs and clock generators even
between cards of the same model, it is not sensible to blindly copy
the settings (e.g. a Device section). For this reason, after you make a
selection, you will still be asked about the components of the card, with
the settings from the chosen database entry presented as a strong hint.

The database entries include information about the chipset, what driver to
run, the Ramdac and ClockChip, and comments that will be included in the
Device section. However, a lot of definitions only hint about what driver
to run (based on the chipset the card uses) and are untested.

If you can't find your card in the database, theres nothing to worry about.
You should only choose a database entry that is exactly the same model as
your card; choosing one that looks similar is just a bad idea (e.g. A
GemStone Snail 64 may be as different from a GemStone Snail 64+ in terms of
hardware as can be).

Do you want to look at the card database? _
```

Agora vamos para a etapa de configuração da placa de vídeo. O utilitário apenas deseja saber se queremos dar uma olhada nos modelos existentes suportados pelo *X.org*. Claro que sim! Se não, como saberemos a especificação da nossa placa de vídeo? Digitem então *y* + <ENTER>.

```
  0  * Generic VESA compatible                          -
  1  * Generic VGA compatible                           -
  2  * Unsupported VGA compatible                       -
  3  ** 3DLabs, TI (generic)              [glint]      -
  4  ** 3Dfx (generic)                    [tdfx]       -
  5  ** ASPEED Technology (generic)       [ast]        -
  6  ** ATI (generic)                     [ati]        -
  7  ** ATI Radeon (generic)              [radeon]     -
  8  ** ATI Rage 128 based (generic)      [r128]       -
  9  ** Alliance Pro Motion (generic)     [apm]        -
 10  ** Ark Logic (generic)               [ark]        -
 11  ** Chips and Technologies (generic)  [chips]      -
 12  ** Cirrus Logic (generic)            [cirrus]     -
 13  ** Cyrix MediaGX (generic)           [cyrix]      -
 14  ** DEC TGA (generic)                 [tga]        -
 15  ** Intel i740 (generic)              [i740]       -
 16  ** Intel i810 (generic)              [i810]       -
 17  ** Linux framebuffer (generic)       [fbdev]      -

Enter a number to choose the corresponding card definition.
Press enter for the next page, q to continue configuration.

_
```

Nesta etapa poderemos observar que existem muitos modelos de placas suportadas pelo servidor gráfico. Basta teclar <ENTER> para ver os demais modelos suportados. Se a placa de vídeo em uso não for tão

antiga, provavelmente encontraremos as especificações de seu modelo e o número correspondente para que possamos informar ao utilitário (na coluna à esquerda). Caso não encontre o modelo de uma placa de vídeo listada, basta utilizar a opção *0* (*VESA*) + <ENTER> para utilizá-la, ou as opções *1* e *2* + <ENTER> caso esta ainda não suporte o padrão *VESA*.

As próximas instruções deste livro se baseiam em um *chipset* de vídeo *3D* integrado pela Via *Technologies*: o *IGP S3 UniChrome*:

```
# lspci -v | grep VGA
01:00.0 VGA compatible controller: VIA Technologies, Inc. VT8378 [S3
UniChrome] Integrated Video (rev 01) (prog-if 00 [VGA])
# _
```

Como ele não consta na listagem principal apresentada pelo utilitário, deveremos pressionar <ENTER> para visualizar as opções seguintes:

```
 18  ** Matrox Graphics (generic)          [mga]       -
 19  ** NVIDIA (generic)                   [nv]        -
 20  ** NeoMagic (generic)                 [neomagic]  -
 21  ** Number Nine I128 (generic)         [i128]      -
 22  ** Rendition (generic)                [rendition] -
 23  ** S3 (not ViRGE or Savage) (generic) [s3]        -
 24  ** S3 Savage (generic)                [savage]    -
 25  ** S3 ViRGE (generic)                 [s3virge]   -
 26  ** SiS (generic)                      [sis]       -
 27  ** Silicon Motion (generic)           [siliconmotion]-
 28  ** Trident (generic)                  [trident]   -
 29  ** Tseng Labs (generic)               [tseng]     -
 30  ** VMWare guest OS (generic)          [vmware]    -
 31  2 the Max MAXColor S3 Trio64V+                    -
 32  2-the-Max MAXColor 6000                           ET6000
 33  3DLabs Oxygen GMX                                 PERMEDIA 2
 34  928Movie                                          S3 928
 35  AGX (generic)                                     AGX-014/15/16

Enter a number to choose the corresponding card definition.
Press enter for the next page, q to continue configuration.

_
```

Observem que existem *3* opções de chipsets *S3*; porém, conforme as informações obtidas no próprio site da *Via*, concluímos não se trata do *S3* diferente do *Savage* ou *ViRGE*. Optaremos então pelo *S3 (not VirGE or Savage)*, ou seja, um *chipset S3* genérico. Teclem *23* + <ENTER>.[1]

```
23
```

1 Um detalhe importante a ser mencionado é que, mesmo com a escolha do *driver* genérico *S3*, a configuração feita manualmente não funcionou; mas ao inicializar o utilitário de detecção *xorgsetup*, este detectou o *chipset* corretamente e o configurou para utilizar o *driver via*, vindo a funcionar perfeitamente (consultamos o arquivo de configuração gerado). Mesmo sabendo que o servidor *X.org* suporta este *chipset* corretamente (inclusive com a aceleração gráfica), não encontramos a menção do *driver via* na listagem das placas suportadas. Por isso, manteremos estas informações apenas para efeito de instrução.

```
Your selected card definition:

Identifier: ** S3 (not ViRGE or Savage) (generic) [s3]
Chipset:    -
Driver:     s3

Press enter to continue, or ctrl-c to abort.
```

Apenas confirmem a configuração escolhida, pressionando <ENTER> para continuar. Caso tenham feito uma escolha errda, acionem a combinação de teclas <CTRL> + *C* para cancelar todo o processo de configuração.

Memória da placa de vídeo

A partir deste ponto, deveremos definir a quantidade de memória disponível físicamente na placa de vídeo, ou compartilhada com a memória principal, caso estejamos utilizando um *chipset* de vídeo integrado na placa-mãe.

```
Now you must give information about your video card. This will be used for
the "Device" section of your video card in xorg.conf.

It is probably a good idea to use the same approximate amount as that detected
by the server you intend to use. If you encounter problems that are due to the
used server not supporting the amount memory you have, specify the maximum
amount supported by the server.

How much video memory do you have on your video card:

 1  256K
 2  512K
 3  1024K
 4  2048K
 5  4096K
 6  8192K
 7  16384K
 8  32768K
 9  65536K
10  131072K
11  262144K
12  Other

Enter your choice: _
```

Praticamente todas as opções se encontram aqui, porém há casos em que a quantidade de memória não se enquadra na listagem apresentada. Um belo exemplo é a antiga aceleradora de vídeo *Monster Fusion*, da *Diamond*, equipada com o *chipset Voodoo2* e *12 MB* de memória *RAM*. Neste caso deverá ser digitada a opção *12* + <ENTER> e o utilitário logo solicitará informar a quantidade de memória em *KB* da placa de vídeo em questão, neste caso o valor *12288*. Veja o pequeno exemplo abaixo:

```
Enter your choice: 12

Amount of video memory in kbytes: _
```

Mas, como estamos lidando com um *chipset* em que foi definido o valor de *16 MB* de memória compartilhada, bastará então utilizar a opção *7*.

Descrição do periférico

Nesta seção, o utilitário apenas exibe o modelo definido pela seleção manual e, da mesma forma que foi feito na vez do *monitor*, é solicitada uma nomenclatura para a identificação da placa de vídeo.

```
Enter your choice: 7

You must now enter a few identification/description strings, namely an
identifier, a vendor name, and a model name. Just pressing enter will fill
in default names (possibly from a card definition).

Your card definition is ** S3 (not ViRGE or Savage) (generic) [s3].

The strings are free-form, spaces are allowed.
Enter an identifier for your video card definition: _
```

A especificação desta identificação é opcional, podendo ser livremente definida ou não. Em nosso caso, digitaremos *Via IGP S3 UniChrome* + <ENTER>; mas, poderíamos perfeitamente ter teclado apenas <ENTER>.

Resoluções e profundidade de cor

Agora iremos definir quais as resoluções e profundidade de cor que serão apresentadas pelo sistema quando executarmos o ambiente gráfico *X*.

```
For each depth, a list of modes (resolutions) is defined. The default
resolution that the server will start-up with will be the first listed
mode that can be supported by the monitor and card.
Currently it is set to:

"1280x1024" "1024x768" "800x600" "640x480" for 8-bit
"1280x1024" "1024x768" "800x600" "640x480" for 16-bit
"1280x1024" "1024x768" "800x600" "640x480" for 24-bit

Modes that cannot be supported due to monitor or clock constraints will
be automatically skipped by the server.

 1  Change the modes for 8-bit (256 colors)
 2  Change the modes for 16-bit (32K/64K colors)
 3  Change the modes for 24-bit (24-bit color)
 4  The modes are OK, continue.

Enter your choice: _
```

Observem que foram habilitadas várias resoluções de vídeo para outras várias profundidades de cores. Assim, dependendo da profundidade de

cor desejada, poderemos definir quais as resoluções que serão adequadas. Em nosso caso, iremos definir as resoluções que serão utilizadas na profundidade de cor de *24 bits*. Para isto, digitem *3* + <ENTER>.

```
Enter your choice: 3

Select modes from the following list:

 1  "640x400"
 2  "640x480"
 3  "800x600"
 4  "1024x768"
 5  "1280x1024"
 6  "320x200"
 7  "320x240"
 8  "400x300"
 9  "1152x864"
 a  "1600x1200"
 b  "1800x1400"
 c  "512x384"
 d  "1400x1050"

Please type the digits corresponding to the modes that you want to select.
For example, 432 selects "1024x768" "800x600" "640x480", with a
default mode of 1024x768.

Which modes? _
```

Como podem ver, o utilitário nos oferece uma série de opções para definirmos a(s) resoluções que desejamos ter nesta profundidade de cor.

Podemos escolher tanto uma única definição, digitando o valor correspondente + <ENTER> ou ainda definirmos múltiplas resoluções, utilizando uma combinação de valores com os números correspondentes, conforme o próprio utilitário exemplificou com a opção *432*.[2]

Em nosso caso utilizaremos apenas a opção *1024x768*, bastando digitar *4* + <ENTER>. Para qqueles usuários que se sentem desconfortáveis com o uso desta resolução, podem optar por *3* + <ENTER>.

```
Which modes? 4

You can have a virtual screen (desktop), which is screen area that is larger
than the physical screen and which is panned by moving the mouse to the edge
of the screen. If you don't want virtual desktop at a certain resolution,
you cannot have modes listed that are larger. Each color depth can have a
differently-sized virtual screen

Please answer the following question with either 'y' or 'n'.
Do you want a virtual screen that is larger than the physical screen? _
```

2 A opção *432* seleciona as resoluções *1024x768*, *800x600* e *640x480*, onde a *1a.* definição (valor *4*) define a resolução de *1024x768* como padrão assim que o modo gráfico é inicializado.

O utilitário pergunta se o usuário deseja ter uma tela virtual mais larga que a tela padrão. Este é um recurso interessante quando se trabalha com editorações gráficas ou tratamento de imagem, onde poderemos deslocar o *mouse* para o canto e a tela se arrastar, podendo mostrar uma "*área extra*" de vídeo para que possa trabalhar. Mas para uso comum, infelizmente não ajuda muito. Se nos permitem o conselho, teclem *n* + <ENTER>.

```
For each depth, a list of modes (resolutions) is defined. The default
resolution that the server will start-up with will be the first listed
mode that can be supported by the monitor and card.
Currently it is set to:

"1280x1024" "1024x768" "800x600" "640x480" for 8-bit
"1280x1024" "1024x768" "800x600" "640x480" for 16-bit
"1024x768" for 24-bit

Modes that cannot be supported due to monitor or clock constraints will
be automatically skipped by the server.

 1  Change the modes for 8-bit (256 colors)
 2  Change the modes for 16-bit (32K/64K colors)
 3  Change the modes for 24-bit (24-bit color)
 4  The modes are OK, continue.

Enter your choice: _
```

Enfim, o utilitário novamente nos mostra a lista de profundidade de cores para que possamos definir as demais profundidades & resoluções. Note agora que a profundidade de cor de *24 bits* possui apenas uma única definição de resolução, graças a escolha que realizamos ao configurá-la.

Caso queiram escolher outras resoluções as demais profundidade de cores, digitem a opção que lhe corresponde + <ENTER>. Senão, confirmem as escolhas feitas digitando a opção *4* + <ENTER>.

```
Please specify which color depth you want to use by default:

  1  1 bit (monochrome)
  2  4 bits (16 colors)
  3  8 bits (256 colors)
  4  16 bits (65536 colors)
  5  24 bits (16 million colors)

Enter a number to choose the default depth.
```

Agora iremos definir a profundidade de cores padrão do sistema, ou seja, a quantidade de cores simultâneas o qual o servidor *X* poderá exibir ao executarmos o ambiente gráfico. Escolheremos a opção *5* + <ENTER>, pois foi a única que definimos um único padrão de resolução.

Finalizando

Enfim, para encerrar este processo, o utilitário questionará se

desejaremos gravar as alterações em um arquivo texto onde o *X.org* armazena todo os parâmetros de configuração. Antes disso o utilitário nos faz um aviso, informando que este arquivo de configuração será gerado no diretório padrão do usuário ao invés de sobrescrever as definições já existentes.

```
I am going to write the xorg.conf file now. Make sure you don't accidently
overwrite a previously configured one.

Do you want it written to the current directory as 'xorg.conf'? _
```

Simplesmente confirmem esta opção, teclando *y* + <ENTER>.

```
File has been written. Take a look at it before starting an X server. Note
that
the xorg.conf file must be in one of the directories searched by the server
(e.g. /etc/X11) in order to be used. Within the server press
ctrl, alt and '+' simultaneously to cycle video resolutions. Pressing ctrl,
alt and backspace simultaneously immediately exits the server (use if
the monitor doesn't sync for a particular mode).

For further configuration, refer to the xorg.conf(5x) manual page.

$ _
```

Em nosso caso, será gerado um arquivo xorg.conf em */home/darkstar*, o qual deveremos nos autenticar como superusuário e copiar este arquivo com as novas definições para a sua localização padrão: em */etc/X11*.

```
$ su
Password: [SENHA]
# copy xorg.conf /etc/X11
# exit
exit
$ _
```

Caso já possuam alguma configuração anterior funcional, é recomendável que se faça uma cópia de segurança da antiga configuração. Se por ventura vier a ocorrer algum imprevisto, bastará apenas sobrescrever o arquivo de configuração problemático pela cópia de segurança funcional.

Enfim, agora é só iniciar o ambiente gráfico, digitando simplesmente *X* + <ENTER> e verificar se não ocorre algum problema durante a execução. Se estiver o ponteiro em forma de cruz circulando na tela à disposição do *mouse*, é sinal de que está tudo (aparentemente) bem.

Outras ferramentas

Além do *xorgconfig*, temos também disponível os utilitários *xorgcfg* e *xorgsetup*. O primeiro provê uma interface gráfica para a realização dos ajustes, embora precária em recursos visuais; já o segundo provê um sistema de detecção para o *hardware* em uso:

```
$ xorgsetup
```

... ou...

```
$ xorgcfg
```

Ambos são simples e fáceis de serem utilizados; por isto, não iremos descrever instruções gerais para o seu uso.

As aceleradoras gráficas 3D

Em seus primeiros tempos, a necessidade de se processar recursos gráficos era pouca ou quase nenhuma para as aplicações que existiam para as placas de vídeo. Porém, com a evolução da informática, sofisticados recursos de computação gráfica tem evoluido e tornado-se necessários para a implementação de muitos programas, em especial os jogos em *3a.* dimensão, onde os recursos de processamento eram ineficientes. Foi desenvolvido então um *chipset* especial que visa suprir as deficiências em processamento gráfico dos computadores modernos, instalados em uma placa de vídeo. À partir deste momento, nasceu então o que chamamos de aceleradora de vídeo, uma placa de vídeo preparada especialmente para a demanda de processamento gráfico, tendo ao seu dispor um processador gráfico – também conhecido por *GPU*[3] – desenvolvido exclusivamente para o processamento avançados recursos de computação gráfica.

O servidor gráfico e as aceleradoras

À partir da versão *4.1.0* do antigo *Xfree86*, o servidor gráfico passou a suportar nativamente os recursos de aceleração gráfica disponíveis em diversas placas *3D* do mercado ("*herdado*" também pelo *X.org*). Com isto, os sistemas *GNU/Linux* passaram a ter suporte a diversas aplicações que requerem o uso de aceleração via *hardware*, como os modeladores gráficos, ferramentas de arte gráfica, jogos e muitos outros. Porém, muitos dos *drivers* inclusos, apesar de terem o código aberto e apresentarem bom desempenho, encontram-se ainda em um patamar bem inferior aos *drivers* de código fechado dos principais fabricantes, e mesmo tendo a opção de uso destes, ainda serão necessários a instalação dos *drivers* proprietários.

As aceleradoras de vídeo da série *Voodoo*, fabricadas pela então extinta *3DFx*, foram as primeiras a terem suporte em sistemas *GNU/Linux* graças a liberação das especificações de sua *API Glide*, desenvolvida pela *3DFx*, o qual a empresa a forneceu para o desenvolvimento de *drivers* nativos sob os termos de licenciamento *NDA - Non-disclosure agreement* –, que entre suas exigências, tais informações não deveriam ser repassadas a terceiros. Mas, por volta do ano *2000*, este grande fabricante se encontrava no vermelho e fechar as portas para ela era inevitável. Mas antes, foi adquirida pela *nVidia*, uma outra empresa que, na época, era a

3 *GPU – Graphics Processor Units*, significa "*Unidade de processamento gráfico*".

sua principal concorrente.

Também não deixando nada a desejar, a *nVidia* foi a primeira empresa a disponibilizar o código-fonte dos *drivers* para as aceleradoras gráficas da família *TNT*, os quais se encontram disponíveis em sua página oficial. Atualmente, ela desenvolve apenas *drivers* proprietários para as atuais *GeForce*, embora possuam um excelente desempenho.

Já a *Matrox* foi uma das primeiras empresas a disponibilizar especificações de seus *chipsets* gráficos para que os desenvolvedores de código-aberto pudessem desenvolver *drivers* livres para estes periféricos. Graças a isto, bastam apenas rodar os utilitários de configuração do modo gráfico para habilitar seus *drivers*, como se fosse simplesmente uma placa de vídeo comum e logo em seguida realizar alguns pequenos ajustes para habilitar os recursos de aceleração gráfica.

Por último, a *ATI* é a empresa que possui menos *"tradição"* em suas iniciativas, embora também disponibilize *drivers* proprietários para as suas aceleradoras *Radeon*. Porém, a qualidade destes *drivers* são bastante discutidas em comparação a concorrente principal *nVidia*, ainda que sejam superiores aos drivers livres existentes do projeto *X.org*.

Preparativos necessários

GLX / DRI

- ✔ <http://www.sgi.com/software/opensource/glx/>.
- ✔ <http://dri.sourceforge.net/>.

A *GLX* e a *DRI* são extensões do servidor gráfico *X.org* para facilitar o processamento de aplicações que utilizam a *OpenGL* para gerar gráficos tridimensionais. Além da configuração da placa de vídeo, será necessário habilitar estes módulos para a utilização dos recursos gráficos disponíveis nas aceleradoras. Para isto edite o arquivo */etc/X11/xorg.conf* e localizem o seguinte trecho da seção *Module*:

```
# This loads the GLX module
#    Load       "glx"
# This loads the DRI module
#    Load       "dri"
```

Descomente as linhas referentes aos módulos *GLX* e *DRI*.

```
# This loads the GLX module
    Load       "glx"
# This loads the DRI module
    Load       "dri"
```

Salve as alterações e reinicie o modo gráfico.

Suporte ao barramento AGP

Para que as aceleradoras gráficas utilizem o recurso de armazenamento de texturas na memória, é necessário que esteja ativado o módulo *agpgart*. Para isto, verifique inicialmente se o mesmo está carregado.

```
# lsmod | grep agp
via_agp                 8192  1
agpgart                26928  1 via_agp
# _
```

Se não for exibida uma mensagem similar a esta, provavelmente o módulo não foi detectado pelo sistema e isto se deve ao fato de que o *kernel 2.6* não o suportar por padrão. Ainda assim, é possível utilizar perfeitamente os recursos de aceleração gráfica da placa ou *chipset* de vídeo, embora o armazenamento de texturas permaneça desabilitado.

Drivers proprietários

Em virtude do grande crescimento da utilização de sistemas *GNU/Linux* em aplicações gráficas, as principais fabricantes de *GPU* passaram a desenvolver os *drivers* de seus produtos para este sistema operacional. Mas infelizmente o código-fonte destes *drivers* encontram-se fechados, pois os desenvolvedores temem que detalhes técnicos importantes da arquitetura de suas placas de vídeo sejam revelados aos seus principais concorrentes.

Dentre os principais fabricantes, destacam-se a *ATI* e a *nVidia*.

ATI

- ✔ <http://www.ati.com/>.
- ✔ <http://r300.sourceforge.net/>.

A grande canadense *ATI* é atualmente a maior concorrente da poderosa *nVidia*. Inicialmente suas placas aceleradoras de vídeo não eram reconhecidas pela performance de seus processadores gráficos, e sim pela qualidade de seus produtos para a captura de vídeo – a famosa linha *ATI All-in-Wonder*. Posteriormente, passaram a investir na qualidade de seus *chipsets* e atingiram o ápice de produzirem as aceleradoras mais possantes atuais durante um certo tempo.

Apesar disto, sua supremacia no mercado ainda não foi confirmada devido a popularidade das *GeForce* e seu baixo custo de aquisição, além da retomada do antigo posto de a fabricante das *GPUs* mais possantes do mercado.

As aceleradores gráficas fabricadas da família *Radeon* possuem bons *drivers* disponíveis para os sistemas *GNU/Linux*. Visitem a página oficial do fabricante e acessem a seção *Drivers* para obter os *drivers* da aceleradora gráfica, caso já não as tenha.

Os *drivers* proprietários da *ATI* são fornecidos em um pacote pré-compilado, bastando apenas instalá-lo via *RPM* e seguir as instruções de um *script* especial de configuração, que aos moldes do *xorgconfig*, realiza uma série de perguntas e cria um arquivo de configuração. Certifiquem-se de que conheçam todas as seções referentes a este processo, pois caso contrário, consultem a seção *Processos básicos de configuração* para uma melhor noção das questões apresentadas.

Uma característica interessante do instalador da *ATI* está na remoção dos *drivers*: o próprio programa se encarrega de restaurar as definições anteriores, deixando praticamente intacto a configuração do servidor gráfico existente antes da instalação dos mesmos.

Nem todas as placas aceleradoras são suportadas pelos *drivers* proprietários da *ATI*, como também existem certos requerimentos a serem considerados (como a versão do servidor gráfico suportada). Por isto, visitem a página oficial do fabricante e, na seção *Drivers*, confirmem se os *drivers* disponibilizados atenderão perfeitamente.

nVidia

- ✔ <http://www.nvidia.com/>.

A principal marca do mercado e seguida de perto pela *ATI*, a *nVidia* atualmente possui os melhores *drivers* disponíveis para os sistemas *GNU/Linux*. As aceleradoras de vídeo da série *GeForce* são atualmente as líderes no mercado de placas aceleradoras de vídeo.

A *nVídia* inovou ao desenvolver um conjunto de pacote de *drivers* chamado *Detonador*, ao invés de um pacote de *drivers* para cada modelo comercializado. Estes *drivers* são atualizados constantemente, e graças à este procedimento, atualmente estes se encontram em um ótimo estágio de desenvolvimento, apresentando ótimo desempenho e grande maturidade. Além disso, estes *drivers* suportam toda a linha da série *GeForce*, diferente da *ATI* que apenas suporta a série *8500* em diante.

Atualmente os *drivers* do servidor *X.org* somente provê suporte *2D* destas aceleradoras gráficas, sendo então estritamente necessários a obtenção dos *drivers* fechados da *nVidia*. Para obter estes *drivers*, deveremos visitar a página oficial do fabricante e procurá-los na seção *Download drivers*.

Tal como a *ATI*, a *nVidia* não dá suporte a todas as aceleradoras gráficas da família *GeForce*, onde geralmente as versões mais antigas acabam deixando de serem suportadas pelos *drivers* proprietários. Para obterem maiores informações sobre estes aspectos, consultem a sua página oficial.

Sobre o X.org

- ✔ <http://www.x.org/>.

- ✔ <http://www.xfree86.org/>.

O *XFree86* é uma implementação livre do *X Windows*[4] para os sistemas baseados em *Unix*. Foi desenvolvido por uma equipe de programadores originalmente liderada por *David Wexelblat*. Apesar do nome, o servidor gráfico roda em inúmeros sistemas operacionais, sejam livres ou não, além de diferentes *hardwares*.[5]

Mas, devido a algumas mudanças dos termos da licença do *XFree86* que tornaram-na incompatível com a *GPL* e conseqüentemente a rejeição por parte das distribuições, entrou em cena o *X.org*, que nada mais é que uma implementação do *XFree86 4.4-RC2*, que por sua vez ainda mantinha as antigas cláusulas da licença antiga. Por ser compatível com a *GPL*, o trabalho desenvolvido pode ser utilizado sem maiores problemas.

Sobre os drivers VESA

Possivelmente muitos usuários deverão estar se perguntando: "*Ué, mas se o X.org suporta uma grande variedade de placas de vídeo, porquê a minha antiga placa modelo X, capacidade Y e marca Z não se encontra aqui?*"

Bem, até a versão *3.3.6*, o servidor gráfico suportava uma quantidade de modelos antigos relativamente grande, porém já estava ficando saturado da grande quantidade de *drivers* existentes e pouco uso. Então a *Conectiva* desenvolveu um conjunto de *drivers VESA* genérico para suprir o uso destas placas de vídeo, pois ainda praticamente todas as placas de vídeo em uso suportam o padrão *VESA*.

A partir da versão *4.0*, o *X.org* deixou de incorporar *drivers* das placas de vídeo mais antigas, substituindo-as pelo driver *VESA* genérico.

Conclusão

Houve um tempo em que uma das maiores limitações dos sistemas *GNU/Linux* era a performance gráfica. Dada as limitações do servidor gráfico em seus primeiros tempos, seu desempenho era modesto na utilização de aplicações gráficas *3D*, porém suficiente para as aplicações básicas de manipulação de janelas do dia-a-dia. Hoje, devido ao grande apoio das empresas e distribuições, o servidor gráfico avançou de tal

---

4 O sistema gráfico *X Windows System* foi concebido em *junho* de *1984* por *Robert W. Scheifler*. O seu objetivo era apenas prover um sistema de janelas que deve rodar de forma transparente em ambientes de rede. Foi originalmente foi escrito no *MIT*, onde posteriormente passou a ser utilizado como o ambiente gráfico *X* padrão dos sistemas *UniX* pelos seus desenvolvedores comerciais.

5 A fase marcante do projeto iniciou-se em meados de *1992*, quando vieram à luz, os sistemas *GNU/Linux* e *BSD*. Pela necessidade de contar com um ambiente gráfico para estes sistemas operacionais, o projeto passou à contar com a colaboração destes, tendo um grande salto em desenvolvimento e maturidade.

ponto que poderemos usufruí-lo praticamente no mesmo nível de performance em comparação aos demais sistemas operacionais. &;-D

II. Audio – placa de som

Introdução

Atualmente a grande maioria das placas e *chipsets* de som são suportadas pelos sistemas *GNU/Linux*, graças aos esforços de diversos programadores. Para melhorar o cenário, a maioria os fabricantes também disponibilizam os *drivers* para os seus produtos, embora não sejam todos.

Este capítulo tem como objetivo apresentar as propriedades relacionadas ao subsistema de áudio; em especial, a arquitetura *ALSA*.

Preparativos iniciais

Detecção do chipset para o som

Para obtermos as informações sobre as especificações técnicas do sistema de áudio, deveremos utilizar o comando *lspci*, combinado-o com a palavras-chave *audio* para filtrar as referências dos dispositivos:

```
# lspci -v | grep -i audio
```

Deverá ser exibido na tela do *monitor* uma mensagem constando as referências técnicas do *hardware* instalado no computador:

```
00:07.0 Multimedia audio controller: Creative Labs SB Live! EMU10k1 (rev 07)
```

Neste exemplo, as informações exibidas pelo utilitário referem-se a uma *Creative Sound Blaster Live! PCI*, com o processador de áudio *Emu10k1*.

```
00:0a 0 Multimedia audio controller: Ensoniq ES1371 [AudioPCI-97] (rev 06)
```

Neste outro exemplo trata-se de uma placa de som *Crystal* com processador de áudio *Sound Blaster 64V PCI*, onde também consta seu módulo correspondente, neste caso *es1371*.

Em muitos casos existirão a necessidade de se obter informações mais detalhadas sobre o periférico em questão. Para isto, digitem...

```
# lspci -v
```

..., onde serão exibidas as informações de diversos periféricos. Localizem somente aqueles trechos referentes ao periférico...

```
00:11.5 Multimedia audio controller: VIA Technologies, Inc. VT8233/A/8235/8237
AC97 Audio Controller (rev 50)
        Subsystem: ABIT Computer Corp. Unknown device 140b
        Flags: medium devsel, IRQ 5
        I/O ports at e400 [size=256]
        Capabilities: [c0] Power Management version 2
```

Neste último caso, trata-se de um dispositivo de áudio *onboard*, onde os

circuitos integrados são provenientes do *chipset VT8235* (ponte sul) e a referência *Abit Computer Corp.* definem o fabricante da placa-mãe.

Checando o carregamento dos módulos

Antes, verifiquem se os módulos foram corretamente carregados:

```
$ lsmod | grep snd
snd_seq_dummy           2692  0
snd_seq_oss            28032  0
snd_seq_midi_event      5888  1 snd_seq_oss
snd_seq                42576  5 snd_seq_dummy,snd_seq_oss,snd_seq_midi_event
snd_pcm_oss            38432  0
snd_mixer_oss          13824  1 snd_pcm_oss
snd_via82xx            22296  0
gameport               10760  1 snd_via82xx
snd_ac97_codec         96292  1 snd_via82xx
ac97_bus                2048  1 snd_ac97_codec
snd_pcm                65160  3 snd_pcm_oss,snd_via82xx,snd_ac97_codec
snd_timer              17540  2 snd_seq,snd_pcm
snd_page_alloc          7432  2 snd_via82xx,snd_pcm
snd_mpu401_uart         6272  1 snd_via82xx
snd_rawmidi            18080  1 snd_mpu401_uart
snd_seq_device          6540  4 snd_seq_dummy,snd_seq_oss,snd_seq,snd_rawmidi
snd                    41956  11
snd_seq_oss,snd_seq,snd_pcm_oss,snd_mixer_oss,snd_via82xx,snd_ac97_codec,snd_p
cm,snd_timer,snd_mpu401_uart,snd_rawmidi,snd_seq_device
soundcore               5728  1 snd
$ _
```

Definindo as permissões de acesso

Para que todos os usuários do sistema tenham acesso aos recursos de áudio existente nestes periféricos, devemos incluí-los no grupo *audio*. Para isto, digitem na linha de comando...

```
# gpasswd -a darkstar audio
Adding user darkstar to group audio
# _
```

Onde *darkstar* representa a nossa conta de usuário para a autenticação.

Não é recomendado realizar intervenções nas permissões de acesso dos dispositivos de áudio. Optem sempre por incluir o usuário no grupo *audio*.

Os drivers ALSA

- ✔ *<http://www.alsa-project.org/>.*

O *drivers* do *Projeto ALSA – Advanced Linux Sound Architecture* – foram desenvolvido com o objetivo de prover uma alternativa de opção para os *drivers OSS – Open Sound System* –, estes utilizados até o *kernel 2.4*.

Dentre suas características, está a sua modularidade, em comparação aos antigos drivers *OSS*. Em virtude da utilização de diversos módulos, os recursos tecnológicos de cada *chipset* diferente são melhor aproveitados, tendo ainda a vantagem de degradar menos o desempenho geral do sistema por este método, além de otimizar o uso da memória ao utilizar um *driver* único para os mesmos recursos de diferentes placas e *chipsets*.

Pelo fato da existência de antigas aplicações que necessitam da utilização de recursos dos antigos *drivers OSS*, os *drivers ALSA* foram projetados para que fossem compatíveis. Assim, não será comprometida a utilização de aplicações que requeiram os *drivers OSS*.

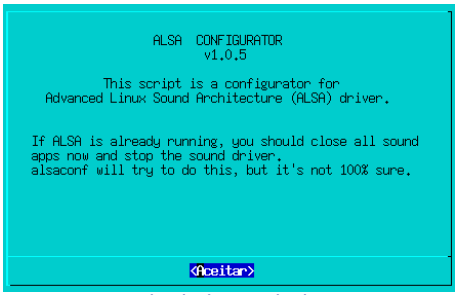
Configurando a placa de som

Conforme enfatizado diversas vezes, o *Slackware* não possui utilitários nativos da distro para a configuração de diversos periféricos do sistema; porém, quanto as placas de som, até antes não havia implementação de nenhum utilitário para esta finalidade, como o *sndconfig* ou qualquer outro utilizado em outras distribuições. Mas a partir da versão *9.0* da distribuição e, com a incorporação dos *drivers ALSA*, este processo passou a ser automatizado graças aos utilitários de configuração inerentes aos *drivers*.

Para iniciar a configuração de uma placa de som, teremos que utilizar uma ferramenta especial que funciona em modo texto: o *Alsaconf*.

```
# alsaconf
```

O utilitário apresentará uma uma tela de boas vidas, descrevendo em seguida algumas de suas características, além de recomendações gerais sobre o processo. Por ser simples e de fácil entendimento, bastará apenas seguirmos as instruções apresentadas por este assistente.

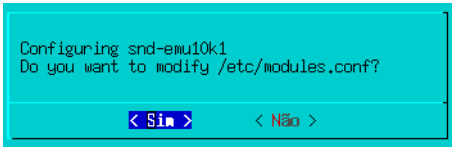


Tela de boas vindas.

O utilitário irá realizar o processo de detecção do periférico instalado no sistema, e ao encontrar as possíveis especificações, será exibido ao administrador os modelos compatíveis suportados.

Especificações do chipset encontrado durante a verificação.

Bastará selecionar o item referente ao periférico e solicitar a alteração do arquivo */etc/modules.conf* para definir as suas propriedades.



Alteração das definições de /etc/modules.conf.

Terminada a configuração, o utilitário realiza as alterações finais para disponibilizar ao sistema os recursos proporcionados pela placa de som.

```
Loading driver...
Loading ALSA mixer settings:  /usr/sbin/alsactl restore
Loading OSS compatibility modules for ALSA.
Setting default volumes...
==========================================================================

 Now ALSA is ready to use.
 For adjustment of volumes, please use alsamixer or gamix.

 Have a lot of fun!

# _
```

Sem grandes mistérios, o periférico e seus recursos estarão disponíveis.

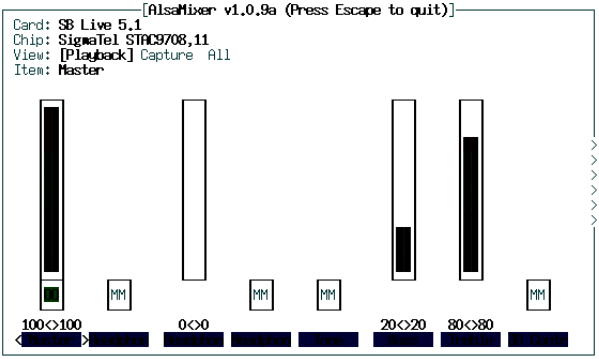
Ajustes finais (mixer)

Uma observação importante é que, mesmo corretamente detectada e configurada, a placa de áudio não emitirá qualquer som pois os canais de áudio são pré-configurados como mudo e o volume no mínimo. Para realizar os ajustes necessários, executem o configurador *alsamixer*, evocando-o na linha de comando com...

```
# alsamixer
```

... onde será apresentado uma interface texto simples, prática e intuitiva, bastando utilizar as teclas <SETA_ACIMA> e <SETA_ABAIXO> para ajustar as proporções de áudio, <SETA_ESQUERDA> e <SETA_DIREITA> para navegar entre os ítens passíveis de configuração.

Para encerrar e gravar as alterações realizadas, basta apenas teclar <ESC>.



Interface de ajuste do AlsaMixer (em preto & branco, para melhor visualização).

Certifiquem-se de que são atribuídos aos usuários o grupo *audio*, necessário para que ele tenha permissões de acesso para a utilização dos recursos de áudio pelos usuários dos sistema. Para isto, consultem a *2a. Parte: Conhecimentos Gerais -> Contas de usuários de grupos de acesso*.

Os devices

Os *device* do sistema de áudio ficam armazenados em */dev/sound*.

```
# cd /dev/sound
# ls -l
total 0
crw-rw--w- 1 root audio 14, 12 2007-08-11 17:40 adsp
crw-rw--w- 1 root audio 14,  4 2007-08-11 17:40 audio
crw-rw--w- 1 root audio 14,  3 2007-08-11 17:40 dsp
crw-rw--w- 1 root audio 14,  0 2007-08-11 17:40 mixer
crw-rw--w- 1 root audio 14,  1 2007-08-11 17:40 sequencer
crw-rw--w- 1 root audio 14,  8 2007-08-11 17:40 sequencer2
# _
```

Novamente reafirmando, não é recomendável redefinir as permissões de acesso destes dispositivos. O ideal é cadastrar o usuário no grupo referido (*audio*), através do uso de ferramentas como o *gpasswd* e o *KUser*.

Problemas mais freqüentes

Em muitas circunstâncias teremos algumas anomalias na utilização do sistema de áudio. Um problema clássico está no fato de ser apresentado uma terrível chiadeira ao ligar os periféricos responsáveis pela reprodução do áudio (caixas de som e fones de ouvido).

Na maioria dos casos, o problema é resolvido apenas com o comando...

```
# alsactl restore
```

Caso não dê certo, verifiquem as propriedades das ferramentas de configuração de áudio de seu ambiente gráfico preferido. Muitas vezes bastam apenas habilitar e/ou desabilitar alguns recursos de áudio no *KMix* (*KDE*), ou *Alsamixer* (*Alsa*) para que o áudio funcione corretamente.

Sobre os antigos drivers OSS

- ✔ *<http://www.4front-tech.com/linux.html>.*

O *kernel* dos sistemas *GNU/Linux* suporta nativamente algumas placas de som. Isto é graças ao antigo *Projeto OSS/Free,* que por sua vez derivou-se do projeto *OSS/Linux*, que são *drivers* produzidos pela empresa *4Front Technologies*, que os comercializa sua licença de uso oferecendo em troca algumas implementações, que por sua vez disponibiliza maiores recursos e possui maiores facilidades para a sua instalação.

Como exemplo, a placa *Sound Blaster Live!* é suportada pela distribuição, do qual na seção *### Sound support ###* consta as seguintes referências:

```
# Sound Blaster Live support:
#/sbin/modprobe emu10k1
```

Após desmarcar o comentário, ficará então assim:

```
# Sound Blaster Live support:
/sbin/modprobe emu10k1
```

Para habilitar imediatamente o suporte ao periférico desejado, basta utilizar na linha de comando a referência acima:

```
# modprobe emu10k1
```

Verifique se existam outros módulos para serem habilitados, pois como exemplo, esta placa de som possui suporte ao *joystick*, que deverá ser habilitado também desmarcando o comentário, este descrito na seção *### Joystick support ###* desta forma:

```
# SoundBlaster Live! Gameports:
/sbin/modprobe emu10k1-gp
```

Sem muito mistério, os demais periféricos, caso constem nesta listagem, deverão ser habilitados utilizando o mesmo procedimento.

Atenção: poderemos utilizar este procedimento para as versões posteriores do *Slackware*; porém estaremos utilizando a arquitetura *OSS* ao invés do *ALSA*, que por sua vez é muito inferior tecnológicamente. Portanto, utilizem estes *drivers* apenas como uma opção para utilizar as antigas placas de som que não são suportadas atualmente pelos *drivers ALSA*.

Conclusão

Felizmente a grande maioria das placas de som possuem suporte nos

sistemas *GNU/Linux* através dos *drivers ALSA*. Na pior das hipóteses, poderemos cogitar o uso dos antigos *drivers OSS* em substituição. Se ainda assim não houver disponibilidade destes para habilitar a placa de som, poderemos consultar a página eletrônica do fabricante do periférico, e na seção *Drivers*, verificar se existem *drivers* disponíveis (geralmente estes criam uma seção *Linux* para os usuários deste sistema). &;-D

III. Modem – placa de fax-modem

Introdução

As placas de *fax-modem* – mais especificamente os *softmodens* - são os periféricos que mais dão (ou davam) dor de cabeça para a sua instalação em sistemas *GNU/Linux*. Dos usuários mais antigos, quais destes ao realizar algumas instalações do sistema em computadores à pedido de amigos não teve acessos de raiva ao chegar a vez de configurar estes periféricos? Que sacrifício! Mas felizmente, isto é coisa do passado...

Em virtude dos esforços da comunidade *GNU/Linux* e de alguns fabricantes, hoje é possível a sua utilização sem maiores incômodos. Mas que incômodos são estes? Saibam agora quais são e o porque neste capítulo.

Preparativos iniciais

Detecção do chipset para o fax-modem

Para obtermos as informações sobre os *chipsets* do *fax-modem* em uso, deveremos utilizar o comando *lspci*, combinado-o com a palavra-chave *Modem* para filtrar as referências dos dispositivos:

```
# lspci -v | grep -i Modem
```

O resultado obtido deverá ser algo similar a...

```
00:0a.0 Serial controller: 3Com Corp, Modem Division 56K FaxModem Model 5610
(rev 01) (prog-if 02 [16550])
        Subsystem: 3Com Corp, Modem Division USR 56k Internal Voice Modem
(Model 2976)
# _
```

As vezes, a palavra-chave *Modem* pode não servir para a filtragem do processo de identificação do dispositivo. Nestas circunstâncias, poderemos utilizar outros termos como *Communication*, *FaxModem*, *HSP* ou qualquer outra referência – inclusive a sua marca, se for do nosso conhecimento:

```
# lspci | grep -i 3Com
00:09.0 Serial controller: US Robotics/3Com 56K FaxModem Model 5610 (rev 01)
# _
```

Se ainda não obtivermos os resultados desejados, deveremos tentar...

```
# lspci
00:00.0 Host bridge: VIA Technologies, Inc. VT82C693A/694x [Apollo PRO133x]
(rev c4)
00:01.0 PCI bridge: VIA Technologies, Inc. VT82C598/694x [Apollo MVP3/Pro133x
AGP]
```

```
00:04.0 ISA bridge: VIA Technologies, Inc. VT82C686 [Apollo Super South] (rev
40)
00:04.1 IDE interface: VIA Technologies, Inc. VT82C586/B/686A/B PIPC Bus
Master IDE (rev 06)
00:04.2 USB Controller: VIA Technologies, Inc. USB (rev 1a)
00:04.3 USB Controller: VIA Technologies, Inc. USB (rev 1a)
00:04.4 Host bridge: VIA Technologies, Inc. VT82C686 [Apollo Super ACPI] (rev
40)
00:07.0 Multimedia audio controller: Creative Labs SB Live! EMU10k1 (rev 07)
00:07.1 Input device controller: Creative Labs SB Live! MIDI/Game Port (rev
07)
00:09.0 Serial controller: US Robotics/3Com 56K FaxModem Model 5610 (rev 01)
01:00.0 VGA compatible controller: nVidia Corporation NV34 [GeForce FX 5200]
(rev a1)
# _
```

... e localizar as informações desejadas.

Em muitos casos existirão a necessidade de se obter informações mais detalhadas sobre o periférico em questão. Para isto, digitem...

```
# lspci -vv
```

..., onde serão exibidas as informações de diversos periféricos. Localizem somente aqueles trechos referentes ao periférico...

```
...
00:0a.0 Serial controller: 3Com Corp, Modem Division 56K FaxModem Model 5610
(rev 01) (prog-if 02 [16550])
        Subsystem: 3Com Corp, Modem Division USR 56k Internal Voice Modem
(Model 2976)
        Flags: medium devsel, IRQ 10
        I/O ports at a000 [size=8]
        Capabilities: [dc] Power Management version 2
...
```

O CÓDIGO-FONTE DO KERNEL

Diversos modelos de *softmodens* possuem *drivers* que necessitam dos pacotes contendo os cabeçalhos (*kernel-headers*) e o código-fonte (*kernel-source*) do *kernel*, onde para isto será necessário utilizar as mídias de instalação do *Slackware*.

Com a mídia inserida no *drive*, digitem os comandos...

```
# mount /dev/cdrom
```

... para montar o *drive*...

```
# cd /mnt/cdrom/slackware/d
# installpkg kernel-headers-[VERSÃO]-[ARQUITETURA].tgz
```

... para instalar o pacote *kernel-headers*....

```
# cd ../k
# installpkg kernel-source-[VERSÃO]-[ARQUITETURA].tgz
```

... para instalar o pacote *kernel-source*.

Atalho e permissões de acesso

Diferente das placas de áudio, os *modens* necessitam da criação de um atalho nominal chamado */dev/modem* para o *device* correspondente:

```
# ln -s /dev/[DISPOSITIVO_DO_MODEM] /dev/modem
```

Lembre-se de que estes dispositivos possuem um *device* de acordo com o perfil *(softmodem/hardmodem)*, o modelo *(ISA/PCI/AMR)* e a porta o qual os *modens* estão configurados. Para saber quais são, consultem a seção *Conflitos de IRQs e portas seriais -> Configuração padrão das portas serias*.

Para que todos os usuários do sistema tenham acesso aos recursos do *modem*, deveremos incluí-los no grupo *uucp*. Digitem na linha de comando:

```
# gpasswd -a [USUÁRIO] uucp
```

Segue um simples e básico exemplo com o usuário *darkstar*:

```
# gpasswd -a darkstar uucp
Adding user darkstar to group uucp
# _
```

Se preferirem, usem utilitários gráficos para esta finalidade (*KUser*).

Processo genérico de configuração

Softmodens

Diferente das placas de audio, infelizmente é impossível desenvolver um processo de configuração genérico para estes periféricos; porém seguem abaixo as principais intervenções necessárias para a grande maioria.

1. Obtenção de informações gerais do *modem*:

Para obterem maiores informações, consultem neste capítulo a seção *Preparativos iniciais -> Detecção do chipset para o fax-modem*.

2. Obtenção dos *drivers*:

Isto deve ser feito tomando-se como base os dados obtidos sobre o *softmodem* em questão. Estes *drivers* podem ser abertos (código-fonte livre) ou fechados (*drivers* pré-compilados). Para obtê-los, consultem a página oficial do fabricante e/ou as páginas especializadas no assunto.

3. Obtenção de dados técnicos referentes ao *modem* e/ou *drivers*:

Para obterem informações gerais, leiam a documentação que acompanha o periférico; para instruções mais específicas, leiam a documentação que acompanha o pacote dos *drivers*.

4. Instalação do código-fonte do *kernel*:

Para realizar a compilação dos *drivers* do *modem*, normalmente será

necessária a presença do código-fonte do *kernel* devidamente instalados. Para isto, consultem o capítulo *O Kernel* para obterem maiores detalhes.

5. Compilação dos *drivers* (módulos):

Necessitaremos de executar um programa de instalação, em caso de *drivers* pré-compilados, ou a famosa combinação *make* e *make install* para os pacotes com o código-fonte. Alguns incluem parâmetros especiais para a configuração, bastando para isto consultar as instruções contidas em arquivos *READMEs* e *INSTALLs* fornecidos como documentação.

6. Atualização das pendências dos módulos:

Na maioria das vezes será necessária a utilização do comando...

```
# depmod -a
```

... para a atualização das pendências dos módulos instalados.

7. Carregamento dos módulos.

Na maioria das vezes, bastará apenas digitar...

```
# modprobe [MÓDULO]
```

Da mesma forma que a configuração de áudio, poderá ser necessário a inclusão de comandos em arquivos de configuração para garantir o carregamento automático dos módulos instalados. Geralmente basta acrescentar este parâmetros no arquivo */etc/rc.local* ou */etc/rc.modules*.

8. Entre outros ajustes finais...

Existirão casos em que serão necessário a edição dos atalhos simbólicos e permissões para os *device drivers* das portas seriais para a placa de *fax-modem*, além da existência de alguns *scripts* personalizados para a automatização de diversos processos, como por exemplo, a habilitação e carregamento dos módulo e suas respectivas pendências. Consultem a documentação de seus *drivers* para obterem maiores detalhes.

HARDMODENS

Somente necessitaremos de criar o atalho simbólico */dev/modem* apontando para o *device* correspondente e assim habilitar os *hardmodens* instalados em seu sistema. Inicialmente excluam o atalho simbólico disponibilizado durante a instalação do sistema operacional em */dev*.

```
# rm /dev/modem
```

Após isto, deveremos apenas criar o atalho simbólico */dev/modem* especificando qual o *device* em que o *hardmodem* se comunica.

```
# ln -s /dev/[DEVICE] /dev/modem
```

Em alguns casos, alguns ajustes extras serão necessários com o *setserial*.

```
# setserial [DISPOSITIVO] irq [IRQ] port [PORTA] uart [UART] [PARÂMETROS]
```

Consultem a seção *Conflitos de IRQs e portas seriais -> Configuração*

padrão das portas seriais para obter as referências necessárias para a correta configuração do *modem*.

Preparativos finais

Carregamento de módulos para a discagem

Certifiquem-se de que os módulos *PPP* para conexão discada estejam préviamente carregados. Como superusuário do sistema, digitem:

```
# lsmod | grep ppp
```

Dentre os módulos disponíveis, deverão constar os seguintes:

```
# lsmod | grep ppp
ppp_deflate             4736  0
ppp_async               8832  1
crc_ccitt               1920  1 ppp_async
ppp_generic            21780  7 ppp_deflate,bsd_comp,ppp_async
slhc                    6144  1 ppp_generic
# _
```

Caso não estejam disponíveis, utilizem o *modprobe* para habilitá-los...

```
# modprobe ppp_generic
# _
```

Para habilitá-los toda vez que o sistema é inicializado (se o mesmo não o fizer), editem o arquivo */etc/rc.d/rc.modules* e inclua esta linha:

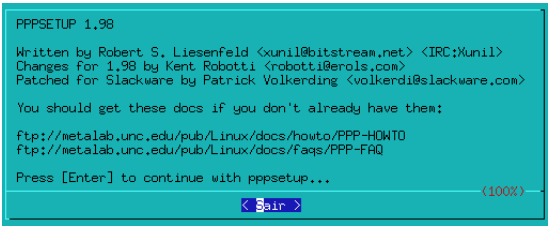
```
modprobe ppp_generic
```

Pronto! Agora, é só realizar a configuração das propriedades de discagem.

Geralmente estes módulos são carregados por padrão, já que os sistemas de autodetecção do sistema fica encarregado deste processo.

Configuração e inicialização do discador

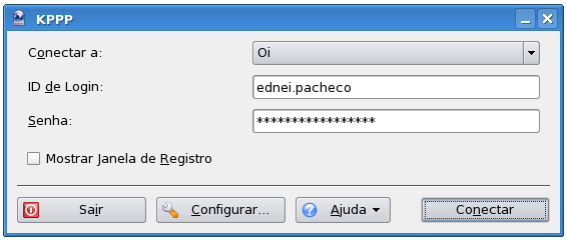
O *Slackware* possui sua ferramenta nativa de configuração, o *PPPD*, um programa relativamente fácil de usar.



pppsetup.

Porém, por se tratar de uma ferramenta em modo texto, possívelmente

não será do agrado de muitos usuários realizar a discagem através deste. Como opção, temos o utilitário *KPPP*, parte integrante do ambiente gráfico *KDE*.



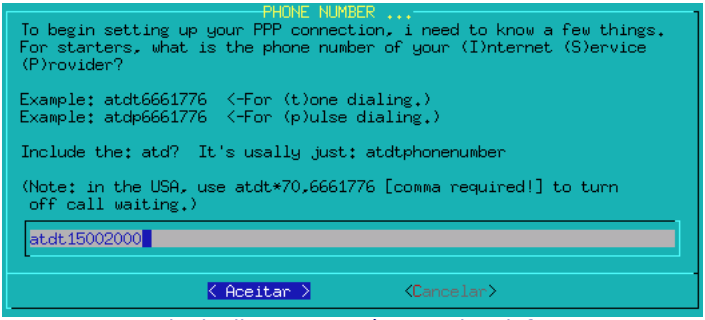
KPPP.

Devido a facilidades proporcionadas pelo *KPPP* para a configuração da discagem, omitiremos as informações gerais sobre este processo, onde nos concentraremos apenas na ferramenta padrão da distribuição.

Para configurar o discador, basta evocá-lo com o comando...

```
# pppsetup
```

... e seguir as instruções do utilitário.

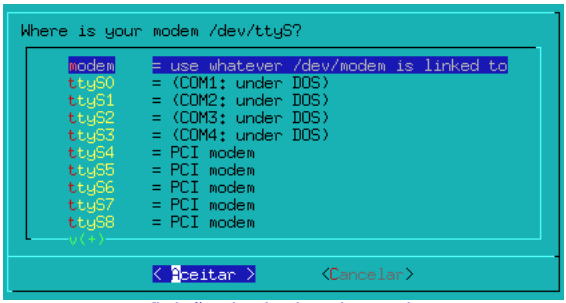


Modo de discagem e números do telefone.

Esta parte é bem simples. Basta digitar o número do telefone do provedor de acesso , precedido antes dos seguintes caracteres:

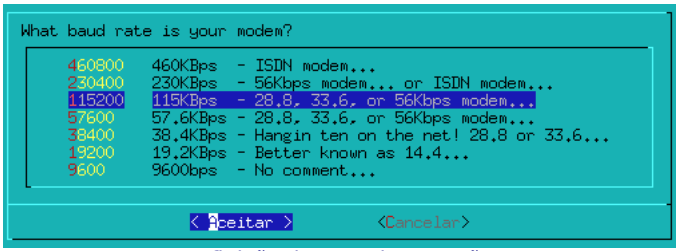
- *atdt* -> Discagem por tom;
- *atdp* -> Discagem por pulso.

Por exemplo, para realizar uma discagem por tom ao provedor *iG*, devemos definir o número de discagem conforme mostrado na figura acima.



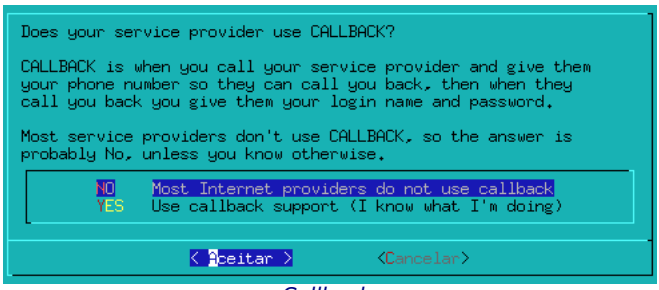
Definição do device do modem.

Nesta seção deveremos identificar em qual porta o *modem* se encontra. É recomendada a criação do atalho */dev/modem* apontando para o *device* correto, e assim selecionar a *1a.* opção para a configuração do *PPPD*.



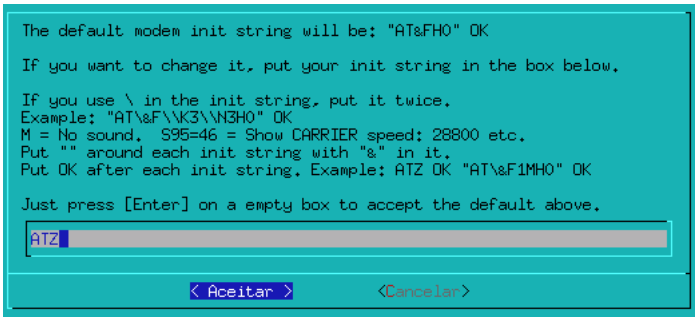
Definição da taxa de conexão.

Neste ponto deveremos informar a capacidade de conexão dos *modens*. Atentem-se para as especificações técnicas do periférico para definirem corretamente os parâmetros. Atualmente os *modens* suportam valores entre *115200* e *57600 bps*.



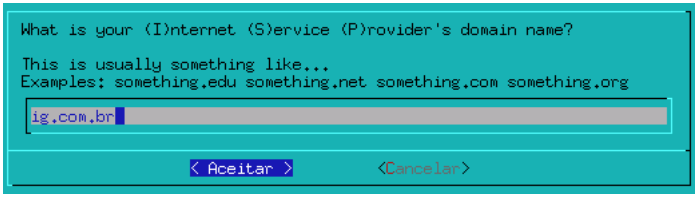
Callback.

Callback é um serviço especial dos provedores de acesso os quais retornam a ligação após efetuada uma discagem para acesso à *Internet*, para a confirmação da conexão e autentificação da senha. Caso não saiba se o seu provedor suporta este serviço, simplesmente selecione *NO* + <ENTER>.



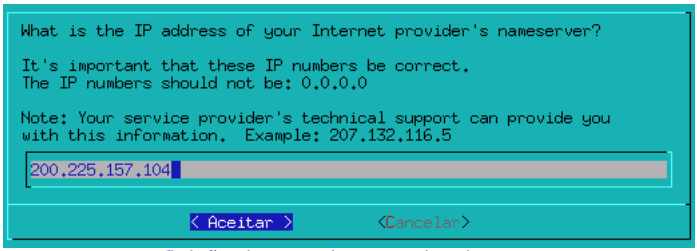
Strings de inicialização.

Para melhorar a velocidade e qualidade da conexão, algums *modens* possuem uma seqüência de caracteres específicas para o equipamento. Consultem na página do fabricante ou em páginas eletrônicas especializadas quais são as *strings* do periférico em uso e informem ao utilitário de configuração. Por padrão são utilizadas as *strings ATZ*.



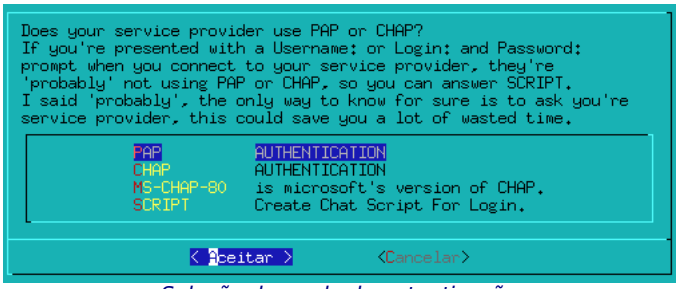
Definição de strings de inicialização personalizada.

Sem muito mistério, nesta parte deverá ser informado o domínio do provedor de acesso, bastando apenas remover as iniciais *http://www*. O exemplo acima citado mostra a definição do provedor de acesso *iG*.



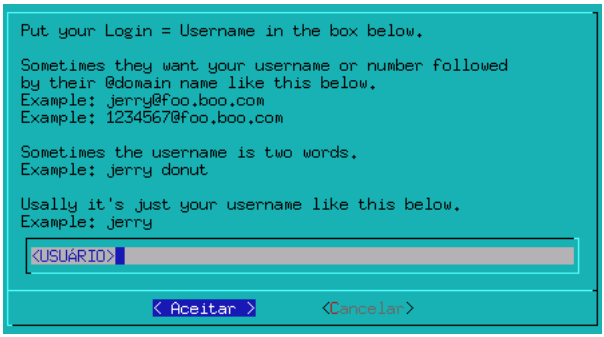
Definição do DNS do provedor de acesso.

Aqui deveremos informar o *IP* do provedor de acesso, que pode variar de acordo com o provedor utilizado. Entre em contato com o provedor para obter as informações desejadas.



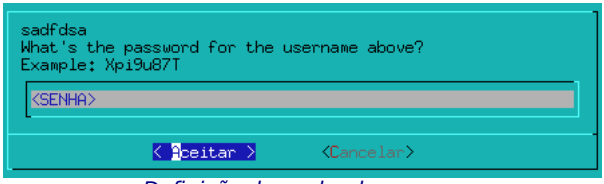
Seleção do modo de autenticação

Geralmente a maioria dos provedores utilizam a autenticação *PAP/CHAP*. Caso não saibam, experimentem inicialmente estes dois. Se estiver tudo certo, a conexão ocorrerá sem maiores problemas.



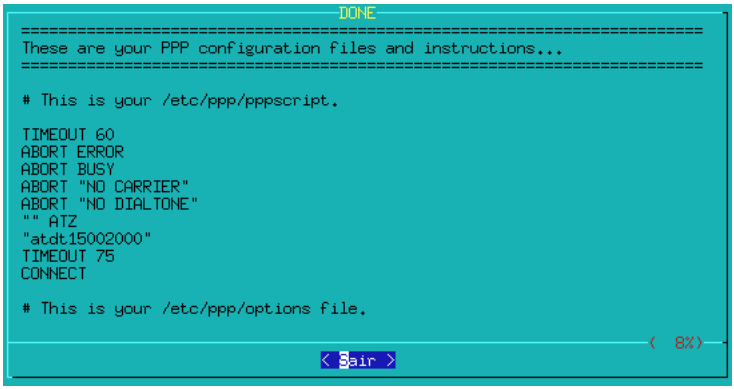
Definição do Nick (apelido) do usuário.

Nesta parte, deverá ser informado o apelido do usuário para realizar a autentificação. Digitem apenas os caracteres antes de *@.[PROVEDOR]*.



Definição da senha de acesso.

Com certeza esta é a parte mais fácil. Basta apenas digitarmos a senha de acesso. Após percorridas todas as etapas, será mostrado as alterações lançadas no arquivos de configuração do discador.



Conclusão do processo, onde são mostradas as definições realizadas.

Estes parâmetros estão gravados nos arquivos */etc/ppp/pppsetup.txt* e */etc/ppp/options* (que poderá ser editado quando necessário).

```
#### A copy of this text can be found in: /etc/ppp/pppsetup.txt ####
#### Look at the /usr/doc/pppsetup/pppsetup-1.98.README. ####

# _
```

Após termos completado todo o procedimento de configuração da discagem, bastará digitarmos na linha de comando...

```
# ppp-go
```

... e aguardar os resultados obtidos durante a conexão.

Para se desconectar, bastam apenas utilizar o comando *ppp-off*.

```
# ppp-off
```

Todas os parâmetros passados ao utilitário *pppsetup* são armazenadas no arquivo */etc/ppp/options*, o qual pode ser alterado conforme necessidade.

```
# General configuration options for PPPD:
lock
defaultroute
noipdefault
modem
/dev/modem
115200
crtscts
# Uncomment the line below for more verbose error reporting:
#debug
# If you have a default route already, pppd may require the other side
# to authenticate itself, which most ISPs will not do.  To work around this,
# uncomment the line below.  Note that this may have negative side effects
# on system security if you allow PPP dialins.  See the docs in /usr/doc/ppp*
# for more information.
#noauth
passive
asyncmap 0
name "[USUÁRIO]"
```

Dentre as alterações necessárias, talvez seja necessário comentar a linha referente à opção *lock*. Ficará então da seguinte forma:

```
# lock
```

Isto se faz necessário pelo fato de alguns discadores simplesmente não funcionarem se esta linha estiver sem comentários.

Em alguma circunstância, pode ser que não seja possível navegar na Internet, embora consigamos realizar a conexão através do discador. Isto acontece com freqüência devido a inexistência de alguns dados do provedor de acesso em */etc/ppp/resolv.conf*. Para isto, editem o arquivo...

```
# mcedit /etc/ppp/resolv.conf
```

... e adicionem as seguintes linhas:

```
search [ENDEREÇO_DO_PROVEDOR]
nameserver [DNS_PRIMÁRIO]
nameserver [DNS_SECUNDÁRIO]
```

Tomemos como exemplo os dados do provedor de acesso *iG*:

```
search ig.com.br
nameserver 200.225.157.104
nameserver 200.225.157.105
```

Salvem as alterações realizadas e a navegação entrará em vigor.

Conflitos de IRQs e portas seriais

Em alguns caso podem ocorrer conflitos de *IRQs* e portas seriais no sistema que impedirão a utilização destes periféricos. Para isto, verifiquem no *setup* da *BIOS* do sistema se estes parâmetros encontram-se corretamente configurados, consultando o manual técnico da placa-mãe, se necessário.

Vejam nas próximas seções as informações gerais sobre as configurações padrões em sistemas *GNU/Linux* e recursos para a realização de ajustes destes, conforme necessário.

Configuração padrão das portas seriais

A configuração padrão das portas utilizadas em sistemas *GNU/Linux* e seus respectivos *IRQs* são as seguinte:

Porta / IRQ / Endereços / UART / Windows				
/dev/ttyS1	*4*	*3F8*	*16550A*	*COM2*
/dev/ttyS2	*3*	*2F8*	*16550A*	*COM3*
/dev/ttyS3	*4*	*3E8*	*Unknow*	*COM4*
/dev/ttyS0	*3*	*2E8*	*Unknow*	*COM1*

Estas informações são de grande valia na execução do *setserial*.

Setserial

O comando *setserial* é utilizado para realizar ajustes e configurações otimizados, de acordo com os atributos dos *hardmodens* disponíveis.

Sintaxe:

```
# setserial [DEVICE] irq [IRQ] port [PORTA] uart [UART]
```

Onde:

- *DEVICE:* o dispositivo propriamente dito;
- *IRQ:* número da interrupção utilizada;
- *PORTA:* definição da porta utilizada;
- *UART:* endereço da memória utilizada pela porta.

Estes são os dados básicos de configuração necessários para a correta configuração do *hardmodem* nos sistemas *GNU/Linux*. Mas como obtê-los? Para isto, utilizem novamente o comando *lscpi*...

```
# lspci -vv
```

... e localizem os dados desejados referentes ao periférico:

```
00:09.0 Serial controller: US Robotics/3Com 56K FaxModem Model 5610 (rev 01)
(prog-if 02 [16550])
        Subsystem: US Robotics/3Com USR 56k Internal Voice Modem (Model 2976)
        Control: I/O+ Mem- BusMaster- SpecCycle- MemWINV- VGASnoop- ParErr-
Stepping- SERR- FastB2B-
        Status: Cap+ 66Mhz- UDF- FastB2B- ParErr- DEVSEL=medium >TAbort-
<TAbort- <MAbort- >SERR- <PERR-
        Interrupt: pin A routed to IRQ 10
        Region 0: I/O ports at a000 [size=8]
        Capabilities: [dc] Power Management version 2
                Flags: PMEClk- DSI- D1- D2+ AuxCurrent=0mA
PME(D0+,D1-,D2+,D3hot+,D3cold+)
                Status: D0 PME-Enable- DSel=0 DScale=2 PME-
```

Após obtermos a *IRQ* e a interrupção desejada, executem...

```
# less /proc/tty/driver/serial
```

... para descobrirmos em qual porta se encontra.

```
serinfo:1.0 driver:5.05c revision:2001-07-08
0: uart:16550A port:3F8 irq:4 baud:9600 tx:0 rx:0
1: uart:16550A port:2F8 irq:3 baud:9600 tx:0 rx:0
```

4: uart:16550A port:A000 irq:10 baud:9600 tx:0 rx:0 CTS|DSR

Observem que foi encontrado em */dev/ttyS4* a *IRQ* e a interrupção obtida com a utilização do comando anterior, além de ser informado o valor da *UART* utilizada. Com todos estes dados em mãos, bastam utilizar...

```
# setserial /dev/ttyS4 irq 10 port 0xA000 uart 16550A [OPÇÕES]
```

... para configurar corretamente o periférico no sistema.

Além das definições exatas dos parâmetros de configurações, poderemos também ajustar a velocidade de transmissão com o objetivo de obter a melhor performance possível. Para isto, utilizem os seguintes parâmetros:

- *spd_normal:* taxa de transmissão de *38.4 kb*;
- *spd_hi:* taxa de transmissão de *57.6 kb*;
- *spd_vhi:* taxa de transmissão de *115 kb*;
- *spd_shi:* taxa de transmissão de *230 kb*;
- *spd_warp:* taxa de transmissão de *460 kb*.

Para obterem maiores informações, consultem a sua documentação.

Sobre os softmodens

Os *modens* são dispositivos de *hardwares* feitos exclusivamente para realizarem a conversão de sinais analógicos para digitais – e vice-versa –, nos possibilitando realizar a conexão com a *Internet* através do uso de uma linha telefônica analógica. Daí o seu nome: ***mod****ulador*/***dem****odulador*.

São subdivididos em duas categorias: os *softmodens* e os *hardmodens*.

São considerados *softmodens* todos os *modens* que não possuem o *chipset* exclusivo para a realização de suas funções, deixando a cargo o processador principal do sistema. São também conhecidos como *winmodens*, pelo fato de que os fabricantes somente disponibilizam *drivers* para o funcionamento destes periféricos no sistema operacional *Windows*.

Os *softmodens* tradicionalmente apresentam as seguintes características:

- São placas menores em virtude da inexistência dos circuitos integrados para o processamento de sinal;
- Utilizam *slots PCI* e *AMR*, exceto as primeiras unidades (*ISA*);
- Apresentam estampadas em suas embalagens termos tais como *softmodens*, *windmodens* e/ou as siglas *HSP*, *HCF*, *HSF*, *SM* (estes últimos são fabricados pela *Motorola*);
- Custo mais em conta, comparado aos tradicionais *hardmodens*.

A principal vantagem da fabricação e comercialização dos *softmodens* está no custo. Mas infelizmente as desvantagens são inúmeras. Devido as suas funcionalidades serem dependentes exclusivamente do sistema operacional, todo o processo é realizado na maior parte ou exclusivamente via *software*, o que certamente sobrecarregará severamente os processadores de médio e baixo desempenho durante sua utilização. Há também a possível instabilidade da linha em decorrência da inexistência de alguns recursos específicos presente nas placas de *fax-modens* tradicionais.

Nos sistemas *GNU/Linux* existe um agravante ainda maior, pois muitos fabricantes[6] não fornecem os respectivos *drivers* para a utilização destes periféricos, além de não disponibilizarem suas especificações técnicas para que outros possam desenvolver os *drivers* de código abertos. Face à estas circunstâncias, evitem ao máximo utilizar estes periféricos. Mas apesar disto, ainda é possível – e viável – a utilização destes periféricos.

Já os *hardmodens* são periféricos que apresentam todos os circuitos necessários para as atividades. Subdividem-se em duas classes: a dos *hardmodens* jumpeados e não-jumpeados. Nesta literatura, trataremos apenas dos não-jumpeados, já que os jumpeados são raros na atualidade.

Estes modens por sua vez, são mais fáceis de configurar, sendo geralmente reconhecidos pelo sistema e ajustados com o comando *setserial* – se não houverem conflitos. A maioria dos *hardmodens* que utilizam slots *PCI* felizmente não são jumpeaveis (especialmente os mais novos), onde com certeza não teremos problemas maiores para a utilização destes periféricos.

Conclusão

Uma observação importante que gostaríamos de enfatizar: a instalação, a configuração e o suporte aos *modens* em sistemas *GNU/Linux* ainda necessitam de intervenções detalhadas e certos conhecimentos técnicos pelo fato de que muitas das especificações técnicas não serem divulgadas pelo fabricante para o desenvolvimento de *drivers* e não pelo fato de que o sistema operacional em si é limitado ou de má qualidade. Na maioria das vezes os desenvolvedores voluntários não são culpados – e sim heróis – por desenvolverem e disponibilizarem *drivers* que muitas vezes ainda se encontram imaturos ou instáveis, face a ausência das informações técnicas necessárias para a sua construção. &;-D

6 Em especial, aqueles fabricados pela *US Robotic*, pois os *drivers* atuais existentes para seus periféricos ainda se encontram em um estágio bastante imaturo

IV. Unidades de armazenamento

Introdução

Apesar das unidades de armazenamento serem muito bem suportadas pelos sistemas *GNU/Linux*, em algumas circunstâncias necessitaremos realizar alguns ajustes para o perfeito funcionamento destes periféricos.

Para esta tarefa, reservamos este capítulo à parte.

Discos rígidos

Em sistemas *GNU/Linux*, os discos rígidos (e os leitores de disquetes) não necessitam de serem configurados, pelo menos não no sentido exato da palavra, já que são automaticamente reconhecidos pelo sistema.

Normalmente deveremos apenas definir sua formatação e os pontos de montagem. No caso das unidades de disco rígido, talvez tenhamos que realizar um ajuste mais fino com o uso de utilitários especiais, como o *HDParm* (para obtermos melhores taxas de performance), ou ainda, o uso do *SmartMon* para a realização de diagnósticos preventivos. Mas isto é feito raramente, em virtude da excelente detecção dos parâmetros do disco rígido pelos atuais sistemas *GNU/Linux*.

Particularmente não recomendamos a utilização da ferramenta *HDParm*, devido a eficiência da detecção de periféricos e do alto risco de algumas opções existentes. Porém, para aqueles que gostam de dar uma *"envenenada"* em seus equipamentos, as instruções pertinentes para este utilitário encontra-se logo abaixo. Por isto, reforçaremos que o seu uso ficará por conta e risco dos interessados. Prevenir nunca é demais...

Para obterem maiores informações sobre a formatação e a montagem de unidades de disco rígido e flexível, consultem a *2a. Parte: Conhecimentos Gerais -> Unidades, partições e formatos*.

HDParm

- ✔ *<http://hdparm.sourceforge.net/>.*

O *HDParm* – abreviação de *HD PARaMeter* – é um utilitário desenvolvido especialmente para realizar o ajuste de parâmetros das unidades de disco rígido, com o objetivo de obter o máximo de desempenho possível.

Sintaxe:

```
# hdparm [PARÂMETROS] [UNIDADE]
```

Para obterem maiores informações sobre os parâmetros disponíveis no *HDParm*, utilize a ajuda interativa para uma explicação mais simples...

```
# hdparm -help
```

... ou as páginas de manual do utilitário, para obter maiores detalhes.

```
# man hdparm
```

Conforme o desenvolvimento das etapas de configuração, forneceremos dicas e instruções de utilização dos principais parâmetros existentes.

Obtendo informações sobre o disco rígido

Inicialmente, certifiquem-se de tenham descrito em mãos todos os recursos e características técnicas de sua unidade, para que estes possam ser consultados na utilização deste utilitário. Manuais (apesar de serem poucos), dados do fabricante, programas de detecção, etc., são excelentes formas de termos disponíveis estas informações. Não só da unidade, como também da placa-mãe, pois o suporte a determinados recursos da unidade também deverá estar nela presente. Aliás, não adianta nada ter um fusquinha em casa e querer colocar um motor de *Ferrari*.

Informações básicas

Para obtermos as informações básicas da unidade de disco rígido em questão, utilizem o parâmetro *i*, digitando na linha de comando...

```
# hdparm -i /dev/[DISPOSITIVO]
```

Em nosso caso, a unidade situa-se na *1a.* controladora, unidade mestre. Como é este o padrão na grande maioria dos computadores domésticos, consideraremos o *device /dev/hda* como exemplo para a otimização. Este valor poderá ser diferente de acordo com a máquina a ser utilizada.

```
# hdparm -i /dev/hda

/dev/hda:

 Model=Maxtor 2B020H1, FwRev=WAK21R90, SerialNo=B1HDVY3E
 Config={ Fixed }
 RawCHS=16383/16/63, TrkSize=0, SectSize=0, ECCbytes=57
 BuffType=DualPortCache, BuffSize=2048kB, MaxMultSect=16, MultSect=16
 CurCHS=16383/16/63, CurSects=16514064, LBA=yes, LBAsects=40020624
 IORDY=on/off, tPIO={min:120,w/IORDY:120}, tDMA={min:120,rec:120}
 PIO modes:  pio0 pio1 pio2 pio3 pio4
 DMA modes:  mdma0 mdma1 mdma2
 UDMA modes: udma0 udma1 udma2 udma3 udma4 *udma5
 AdvancedPM=yes: disabled (255) WriteCache=enabled
 Drive conforms to: ATA/ATAPI-7 T13 1532D revision 0:  ATA/ATAPI-1 ATA/ATAPI-2
ATA/ATAPI-3 ATA/ATAPI-4 ATA/ATAPI-5 ATA/ATAPI-6 ATA/ATAPI-7

 * signifies the current active mode

# _
```

O utilitário mostrará todas as informações básicas do disco rígido, incluindo modelo, capacidade, *serial* e tudo o que tem direito! Vamos

analisar detalhadamente todas as suas propriedades, em especial, aquelas passíveis de ajustes para uma boa melhora de performance.

Recursos habilitados ou desabilitados

Para realizarmos outra simples, porém rápida e eficiente pesquisa das propriedades e parâmetros habilitados, simplesmente utilizem o comando...

```
# hdparm /dev/hda

/dev/hda:
 multcount    = 16 (on)
 IO_support   =  1 (32-bit)
 unmaskirq    =  1 (on)
 using_dma    =  1 (on)
 keepsettings =  0 (off)
 readonly     =  0 (off)
 readahead    =  8 (on)
 geometry     = 2491/255/63, sectors = 40020624, start = 0
# _
```

... onde poderemos obter a maioria dos dados essenciais para a sua boa performance. Veja a descrição de cada um deles:

Avaliando o tempo de acesso

Após obter todos os dados necessários, iremos verificar a performance do disco rígido para que possamos avaliar quais as alterações que deverão ser realizadas. Para isto, digite na linha de comando:

```
# hdparm -Tt /dev/hda

/dev/hda:
 Timing cached reads:   992 MB in  2.00 seconds = 496.00 MB/sec
 Timing buffered disk reads:  112 MB in  3.02 seconds =  37.09 MB/sec
# _
```

Serão exibidas duas taxas de velocidade distintas: a *1a.* é a transferência do *buffer-cache*, o qual o desempenho de *hardware* (controladora), memória e processador reflete diretamente no valor obtido; já a *2a.* está íntimamente ligada à taxa de transferência do disco rígido.

Mão na massa

Enfim, após uma minuciosa análise das informações, iremos por a mão na massa e realizar as intervenções necessárias para obter uma boa performance da unidade em questão. Descreveremos as intervenções por recursos suportados pelo *HDParm*.

Felizmente em virtude da boa detecção de *hardware*, a nossa unidade o qual realizamos os testes encontram-se perfeitamente configurada, onde muitos dos parâmetros já se encontrarão habilitados. Em vista disto,

infelizmente não teremos como disponibilizar exemplos práticos e diferenciados para uma melhor noção das melhorias proporcionadas. Mas ainda assim as instruções estarão claramente descritas para o bom entendimento dos leitores, bastando à eles avaliarem os ganhos obtidos.

AVISO: A EXECUÇÃO DESTE UTILITÁRIO REQUER O CONHECIMENTO PRÉVIO DAS PARTICULARIDADES DO DISCO RÍGIDO O QUAL SE DESEJA UTILIZAR, POIS QUALQUER ERRO OU ATRIBUIÇÃO DE PARÂMETROS INCORRETOS PODERÁ ACARRETAR EM PERDAS DE DADOS ALI GRAVADOS OU ATÉ MESMO DANOS À UNIDADE. PORTANTO, TENHAM TODO O CUIDADO POSSÍVEL.

LBA

A *LBA – Logical Block Addressing* – é uma tecnologia desenvolvida para que as *BIOS* das placas mãe reconheçam unidades de disco rígido com capacidade superior à *528 mbytes.* Todas as placas mãe atuais suportam esta tecnologia, portanto, não há nada para ser modificado aqui.

UDMA Modes

Suporte da unidade ao modo de transferência *Ultra DMA*.

O *DMA – Direct Memory Access* – é um recurso que possibilita o *chipset* da placa-mãe acessar diretamente a memória do sistema sem a intervenção da *CPU*, resultando para ele uma menor carga. Disponível nos discos rígidos atuais, verifiquem se este recurso se encontra habilitado.

Ao obtermos as informações básicas da unidade, observem a existência de um asterisco na sigla *udma5*, o que indica que a unidade suporta esta tecnologia e que a taxa nominal de transferência é de *100 MB.*

```
 PIO modes:  pio0 pio1 pio2 pio3 pio4
 DMA modes:  mdma0 mdma1 mdma2
 UDMA modes: udma0 udma1 udma2 udma3 udma4 *udma5
```

Ao realizarmos a consulta para verificar se o recurso encontra-se habilitado, encontraremos a confirmação através da seguinte informação.

```
 using_dma    =  1 (on)
```

Felizmente em nosso caso o modo de transferência já se encontra habilitado corretamente. Mas caso estivesse desabilitado, faz-se necessário utilizar o seguinte comando:

```
# hdparm -d 1 /dev/hda
/dev/hda:
 setting using_dma to 1 (on)
 using_dma    =  1 (on)
# _
```

Nesta próxima etapa, definiremos o tipo de acesso *DMA*, mas em vista do grande risco que esta opção nos expõe, não recomendamos o seu uso (exceto se o usuário saiba exatamente o que está fazendo). Será de sua responsabilidade quaisquer problemas que possam ocorrer...

Segue a tabela com todos os valores aplicáveis:

Modo, Perfil e Valores		
Padrão	*-*	*00*
PIO MODE	*pio0*	*08*
	pio1	*09*
	pio2	*10*
	pio3	*11*
	pio4	*12*
DMA MODE	*mdma0*	*32*
	mdma1	*33*
	mdma2	*34*
ULTRA DMA MODE	*udma0*	*64*
	udma1	*65*
	udma2	*66*
	udma3	*67*
	udma4	*68*
	udma5	*69*
	udma6	*70*

Para utilizar o modo de transferência correto, utilizem o comando...

```
# hdparm -X [VALOR] /dev/hda
```

Onde *[VALOR]* deverá ser substituído pelo modo correto, lembrando mais uma vez deveremos ter todo o cuidado para a definição deste parâmetro.

IO_SUPPORT

Modo de controle de transmissão de dados do barramento *PCI* para a controladora. Todas as controladoras modernas suportam os modos *1* (*32 bits*) e *3* (*32 bits w/sync*).

Ao checarmos a unidade-exemplo, temos o seguinte valor...

```
IO_support   =  1 (32-bit)
```

... que por sua vez, ao se encontrar habilitada, irá refletir na avaliação do tempo de acesso da unidade:

```
Timing buffered disk reads:  112 MB in  3.02 seconds =  37.09 MB/sec
```

Pelo fato de estar corretamente configurado, a unidade apresenta o valor de *34.78 MB/seg.*, ou seja, dentro dos padrões aceitáveis.

Os possíveis modos de transmissão de dados existentes são:

- *0: 16 bits*;
- *1: 32 bits*;
- *3: 32 bits sync*.

Como a grande maioria dos computadores atuais sejam da família *Pentium* (*i586*) e superiores, acreditamos que a melhor recomendação esteja na utilização do modo *1*. Por via das dúvidas, certifique-se no manual da placa-mãe se o *chipset* suporta ou não o modo *3*. Para a nossa felicidade, geralmente este modo de controle de dados é detectado por padrão.

Caso a unidade suporte este modo de transmissão de dados, mas por algum motivo não se encontra habilitado no valor correto, deveríamos utilizar...

```
# hdparm -c 1 /dev/hda
/dev/hda:
 setting 32-bit I/O support flag to 1
 I/O support  =  1 (32-bit)
# _
```

Esta configuração somente deverá ser feita caso a propriedade *IO_support* já estiver habilitada.

WriteCache

Habilita / desabilita a memória cache da unidade para acesso rápido. Esta opção deverá ser ativada para a melhoria da performance do disco rígido. As unidades de disco rígido atuais suportam esta tecnologia, onde geralmente possuem a capacidade de *2 mbytes* para esta finalidade.

Inicialmente verifiquem se o mesmo suporta o recurso de memória cache através das informações básicas da unidade.

```
 BuffType=DualPortCache, BuffSize=2048kB, MaxMultSect=16, MultSect=8
```

Como podem ver, além de suportada, ainda é indicado a capacidade da propriedade em *2048 kbytes*. Caso a unidade possua este recurso, porém não se encontra habilitada, utilizem o comando...

```
# hdparm -W 1 /dev/hda

/dev/hda:
 setting drive write-caching to 1 (on)
# _
```

MaxMultSect / MultSect

O *MaxMultSect* trata da máxima quantidade de setores que a unidade poderá ler de uma única vez. Na exibição de informações básicas do nosso caso, o resultado para esta configuração são *16* setores.

```
 BuffType=DualPortCache, BuffSize=2048kB, MaxMultSect=16, MultSect=8
```

Basta apenas configurar a unidade para utilizar esta capacidade:

```
# hdparm -m 16 /dev/hda
/dev/hda:
 setting multcount to 16
 multcount    = 16 (on)
# _
```

Diferente da propriedade anterior, o *MultSect* apenas informa quantos destes setores estão sendo realmente utilizados no exato momento da utilização do comando. O valor padrão da configuração exibida nas informações básicas é *8*.

```
 BuffType=DualPortCache, BuffSize=2048kB, MaxMultSect=16, MultSect=8
```

MULTCOUNT X READAHEAD

O *multcount* – abreviatura de *multiple sector count* (contagem de múltiplos setores) – é uma propriedade da unidade em ler mais de um setor a cada acesso realizado. Já o *readahead* é utilizado para definir quantos setores o sistema deverá ler em diante.

A aplicação neste último é indicado para a leitura de grandes arquivos tornando-se ideal para a leitura de imagens *ISO*, vídeos, jogos com centenas de *megabytes*, etc., tornando performance do sistema melhor em virtude do rápido acesso proporcionado. Em contrapartida, é contra-indicado para sistemas que lidam com diversos arquivos de pequenas dimensões, como as atividades cotidianas de um simples usuário (carga do sistema operacional e ambiente gráfico, execução de aplicativos, navegação, etc.).

Fica à critérios dos usuários definirem o que será melhor para seu caso, mas caso este tenha dúvidas, mantenham as mesmas definições padronizadas pelo *hardware* (*multcount*).

Na checagem dos parâmetros habilitados na nossa unidade, à encontramos corretamente configurada conforme abaixo, ...

```
 multcount    = 16 (on)
```

Porém as definições dadas ao *readahead* estão muito aquém do desejado...

```
 readahead    =  8 (on)
```

Lembrem-se que, caso o sistema apresente a propriedade *readahead* desabilitada, deveremos inicialmente habilitá-la com o comando...

```
# hdparm -A 1 /dev/hda

/dev/hda:
 setting drive read-lookahead to 1 (on)
# _
```

Posteriormente, para reconfigurar a quantidade de setores à serem lidos após o acesso à unidade, mantendo as mesmas definições da capacidade do periférico, devemos utilizar o comando...

```
# hdparm -a 16 /dev/hda
```

```
/dev/hda:
 setting fs readahead to 16
 readahead    = 16 (on)
# _
```

ADVANCEDPM

Habilitação do recurso de monitoramento do consumo de energia elétrica. A grande maioria das unidades atuais suportam esta tecnologia, que possibilita o baixo consumo de energia quando da pouca ou nenhuma utilização pelo sistema. É também chamado de hibernação.

UNMASKIRQ

A *unmaskirq* – "*desmascaração da IRQ*", em português – é um recurso que permite ao sistema operacional trabalhar com as demais interrupções mesmo quando o disco rígido estiver trabalhando; trocando em miúdos, permite ao sistema operacional realizar outras tarefas sem ter que aguardar o retorno da interrupção da unidade.

O ideal é tê-la habilitada, porém temos uma observação à fazer: este é mais um dos parâmetros considerados PERIGOSOS para ser utilizado, pois conforme as instruções do manual eletrônico, dependendo da não-tolerância do *drive* e da controladora, este recurso poderá acarretar em danos no sistema de arquivos e conseqüentemente perda de dados.

Analisando os resultados obtidos na checagem dos recursos, vejam:

```
 unmaskirq    =  1 (on)
```

Novamente em nossa unidade-teste encontra-se previamente habilitada, e com isto nos garantindo um ótimo desempenho nas atividades de escrita.

O comando para habilitar a *unmasirq* é:

```
# hdparm -u 1 /dev/hda

/dev/hda:
 setting unmaskirq to 1 (on)
 unmaskirq    =  1 (on)
# _
```

KEEPSETTINGS

O *keepsettings* – manter configurações – como a própria tradução diz, mantém as configurações pré-definidas com a utilização do *hdparm*.

Em nosso exemplo, observe que esta se encontra desabilitada:

```
 keepsettings =  0 (off)
```

O comando para habilitar a *keepsettings* é:

```
# hdparm -k 1 /dev/hda
```

```
/dev/hda:
 setting keep_settings to 1 (on)
 keepsettings =  1 (on)
# _
```

Recomendamos em mantê-la desabilitada, pois na definição de parâmetros incorretos, os mesmos serão desabilitados na reinicialização do sistema.

READONLY

Modo somente leitura. Como na esmagadora maioria das vezes a unidade de disco rígido não trabalha neste modo (lógico), é de melhor conveniência mantê-la desabilitada, onde não haverá necessidade de intervir para isto.

Otimizações finais

Após definidas toda as propriedades do disco rígido, verifiquem novamente o tempo de acesso da unidade para realizarmos uma comparação geral de seu desempenho. Para isto, digitem...

```
# hdparm -Tt /dev/hda
```

... e analisem os resultados obtidos e compare com as definições originais antes da utilização do *HDParm*. Apesar da pouca possibilidade destas unidades estarem mau configuradas após a instalação do sistema, é sempre bom realizar estas verificações e intervir para garantir uma ótima performance deste e de todo o sistema em geral.

"*Infelizmente*" a detecção dos parâmetros da unidade aqui testada estava bem definida, o que impossibilitou-nos de melhorar as instruções através de exemplos práticos. Mas assim que encontrarmos um disco rígido mal configurado, atualizaremos estes exemplos para que os usuários possam ter melhor noção das melhorias proporcionadas por este utilitário.

Para tornarmos efetivas as alterações realizadas todas as vezes que necessitar reiniciar a máquina, editem o arquivo */etc/rc.d/rc.local* e acrescentem o comando com os parâmetros necessários.

```
#!/bin/sh
#
# /etc/rc.d/rc.local:  Local system initialization script.
#
# Put any local setup commands in here:

### HDParm ###
hdparm -P1 -P2 -P3 -P... /dev/[DISCO_RÍGIDO]
```

Onde *P[X]* são os parâmetros que habilitarão cada propriedade. Enfim, sempre que inicializarem o sistema, estas alterações estarão em vigor.

Parâmetros opcionais

Dentre outros parâmetros que encontramos disponíveis em diversos tutoriais, "*destacam-se*".

```
# hdparm -Y /dev/hda
```

Desliga a unidade de disco rígido, tornando-o inativo ao cortar a energia elétrica (sem alimentação). Só não nos perguntem em que circunstância este parâmetro nos será útil...

```
# hdparm -q[PARÂMETROS]
```

Desabilita qualquer mensagem de aviso impressa pela utilização dos parâmetros do *HDParm*. Torna a utilização do utilitário bem mais elegante, quando o mesmo é utilizado com inúmeros parâmetros, como na edição do arquivo */etc/rc.d/rc.local*. Ficaria então assim:

```
### HDParm ###
hdparm -qP1 -qP2 -qP3 -qP... /dev/[DISCO_RÍGIDO]
```

Observações importantes

Riscos

Apesar das melhorias proporcionadas por este utilitário, o *HDParm* deve ser utilizado com o extremo cuidado, pois caso venhamos a especificar algum parâmetro fora do intervalo suportado pelo periférico, teremos sérios riscos de perda de dados ou até mesmo da própria unidade de armazenamento. Por isto, realizem periodicamente uma cópia de segurança dos seus dados.

Limitações

O *HDParm* suporta somente unidades que utilizam a interface *IDE*. Outra questão bastante importante é a utilização da interface correta. Para os discos rígidos com taxa de transferência igual ou superior aos *100 MB/seg* (*ATA 100*), é necessário a utilização do cabo *flat* de *80 vias*, caso contrário, este periférico irá apresentar algumas anomalias de funcionamento em virtude das interferências ocorridas durante o uso.

Independência

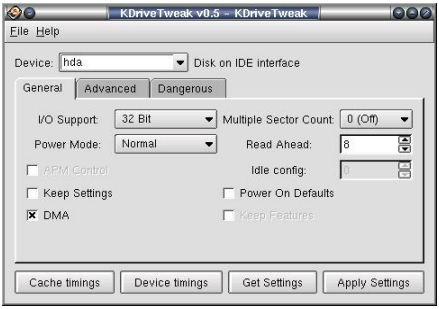
Uma das perguntas freqüentes oriundas em diversos arquivos de lista de discussão é a questão da influência do sistema de arquivos utilizados em uma unidade onde foi feita a otimização com o *HDParm*. O utilitário apenas foca na habilitação dos recursos e propriedades das unidades de disco rígido no sistema. Independente do sistema de arquivos que estivermos utilizando, as alterações realizadas que entrarão em vigor funcionarão do mesmo jeito. Porém quando houver a utilização errônea ou mal dimensionada de um determinado parâmetro, poderá sim haver um

comportamento diferenciado de acordo com o sistema de arquivos utilizado, porém este último NÃO será a causa principal dos distúrbios.

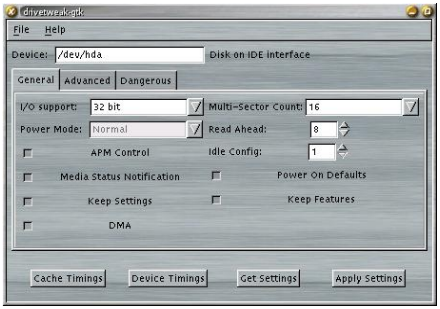
Interfaces gráficas

- ✔ <http://drivetweak.sourceforge.net/>.

Para aqueles habituados a interfaces gráficas (ou que tenham horror a linha de comando), existe uma ótima interface gráfica chamada *DriveTweak* que, desenvolvida para o *HDParm*, utilizando as bibliotecas gráficas *Qt* e *GTK+*.



KdriveTweak
(Tela obtida da página oficial do projeto).



DriveTweak-gtk
(Tela obtida da página oficial do projeto).

O mais interessante é que este utilitário provê todas as opções necessárias classificadas em três categorias: *General* (gerais), *Advanced* (avançadas) e *Dangerous* (perigosas): este recurso é muito interessante para evitarmos "*pequenos enganos*" ao ajustarmos as opções...

Agradecimentos

Em virtude do alto risco das operações citadas, realizamos um estudo intensivo em diversas documentações e tutoriais disponíveis sobre o assunto, os quais gostaríamos destacar e agradecer aos autores *Piter Punk* e *Rogério Ferreira* pela excelente qualidade de seus materiais didáticos, os quais foram preponderantes para a criação destas instruções.

- ✔ *"Melhorando a performance do seu HD", por Piter PUNK*, <http://www.piterpunk.hpg.ig.com.br/artigos/hdparm.html>.
- ✔ *"Melhorando o desempenho do Linux com o hdparm", por Rogério Ferreira*, <http://brlinux.linuxsecurity.com.br/news2/006328.html?redirected=1>.

A documentação eletrônica do utilitário também ajudou bastante, mesmo apesar desta se encontrar no idioma inglês.

SMART

- ✔ <http://smartmontools.sourceforge.net/>.

O *SMART* é um recurso provido pelas *HDs* modernas que possibita a controladora da unidade registrar em uma memória interna as ocorrências de falhas e mau funcionamento. Assim, poderemos nos antecipar e tomar atitudes antes que o *hardware* apresente problemas e ponha em risco a integridade das informações ali contidas. Em sistemas *GNU/Linux*, o acesso a este registro é feito através do programa *SmartMon*.

SMARTCTL

O *smartctl* imprime um relatório do estado atual do disco rígido.

Sintaxe:

smartctl [PARÂMETROS] [UNIDADE]

Onde:

- *-a / -A:* exibe as informações gerais da unidade em uso e o suporte ao *SMART* de forma completa e detalhada;
- *-H:* exibe as informações conforme reportado pela controladora;
- *-i:* exibe as informações tal conforme é exibido pelo parâmetro *-a*, porém de forma simplificada;
- *-s:* habilita ou desabilita o suporte ao *SMART*;
- *-t:* realiza diagnósticos.

Para verificar o estado da unidade do sistema, digitem:

```
# smartctl -i /dev/hda
smartctl version 5.36 [i486-slackware-linux-gnu] Copyright (C) 2002-6 Bruce
Allen
Home page is http://smartmontools.sourceforge.net/

=== START OF INFORMATION SECTION ===
Model Family:     Maxtor Fireball 541DX family
Device Model:     Maxtor 2B020H1
Serial Number:    B1HDVY3E
Firmware Version: WAK21R90
User Capacity:    20,490,559,488 bytes
Device is:        In smartctl database [for details use: -P show]
ATA Version is:   7
ATA Standard is:  ATA/ATAPI-7 T13 1532D revision 0
Local Time is:    Sun Nov  5 09:02:26 2006 BRST
SMART support is: Available - device has SMART capability.
SMART support is: Enabled

# _
```

Neste caso, o *SMART* encontra-se habilitado por padrão. Mas caso não

tivesse, poderíamos habilitá-lo através dos parâmetros...

```
# smartctl -s on /dev/hda
```

Para obtermos um relatório do status atual da unidade, utilizem...

```
# smarctl -A /dev/hda
```

... ou...

```
# smarctl -a /dev/hda
```

... para um resultado mais depurado.

Para realizar um simples e prático diagnóstico, utilizem...

```
# smartctl -t short /dev/hda
```

Se utilizarmos o termo *long* ao invés de *short*, a análise será mais demorada, embora com maior confiabilidade.

```
# smartctl -t long /dev/hda
```

Para exibir um relatório dos diagnósticos realizados...

```
# smartctl -l selftest short /dev/hda
```

SMARTD

O *smartd* carrega o serviço de monitoração do *SMART*. Embora seja possível carregá-lo manualmente através do executável *smartd*, optem por fazê-lo na inicialização do sistema, editando o arquivo */etc/rc.d/rc.local*...

```
# mcedit /etc/rc.d/rc.local
```

E em seguida, incluam o comando *smartd* em seu conteúdo:

```
#!/bin/sh
#
# /etc/rc.d/rc.local:  Local system initialization script.
#
# Put any local startup commands in here.  Also, if you have
# anything that needs to be run at shutdown time you can
# make an /etc/rc.d/rc.local_shutdown script and put those
# commands in there.
```

smartd

Periódicamente poderemos consultar as ocorrências através dos registros de *log* do sistema com a leitura do arquivo */var/log/messages*:

```
$ less /var/log/messages | grep -i smart
```

Em alguns trechos, poderemos ver o andamento das ocorrências...

```
...
Jul 10 05:37:53 darkstar smartd[3091]: Device: /dev/hda, SMART Prefailure
Attribute: 8 Seek_Time_Performance changed from 244 to 243
Jul 10 06:37:54 darkstar smartd[3091]: Device: /dev/hda, SMART Prefailure
Attribute: 8 Seek_Time_Performance changed from 243 to 242
Jul 10 09:37:53 darkstar smartd[3091]: Device: /dev/hda, SMART Prefailure
```

```
Attribute: 8 Seek_Time_Performance changed from 242 to 241
Jul 10 11:37:53 darkstar smartd[3091]: Device: /dev/hda, SMART Prefailure
Attribute: 8 Seek_Time_Performance changed from 241 to 243
Jul 10 12:07:53 darkstar smartd[3091]: Device: /dev/hda, SMART Prefailure
Attribute: 8 Seek_Time_Performance changed from 243 to 245
...
```

Gravadores de CD/DVDs

Quanto mais o custo unitário destas unidades vai baixando, mais popular estes periféricos vão se tornando. As unidades gravadoras de mídia óptica – chamaremos aqui de gravadores de *CD/DVDs* – são dispositivos largamente utilizados pelos usuários *desktops*.

Reconhecimento do sistema

Os gravadores de *CDs/DVDs* são reconhecido normalmente, sem maiores problemas. Caso queiram se certificarem, utilizem o comando...

```
# cdrecord -scanbus
Cdrecord-ProDVD-Clone 2.01.01a23 (i686-pc-linux-gnu) Copyright (C) 1995-2006
Jörg Schilling
cdrecord: Warning: Running on Linux-2.6.21.5
cdrecord: There are unsettled issues with Linux-2.5 and newer.
cdrecord: If you have unexpected problems, please try Linux-2.4 or Solaris.
Linux sg driver version: 3.5.27
Using libscg version 'schily-0.9'.
scsibus1001:
        1001,0,0 100100) 'HL-DT-ST' 'CD-RW GCE-8525B ' '1.03' Removable CD-ROM
        1001,1,0 100101) 'HL-DT-ST' 'DVD-ROM GDR8161B' '0100' Removable CD-ROM
        1001,2,0 100102) *
        1001,3,0 100103) *
        1001,4,0 100104) *
        1001,5,0 100105) *
        1001,6,0 100106) *
        1001,7,0 100107) *
# _
```

Poderemos também lançar mão do comando...

```
$ dmesg | grep CD-R
hdc: ATAPI 52X CD-ROM CD-R/RW drive, 2048kB Cache, UDMA(33)
Uniform CD-ROM driver Revision: 3.20
$ _
```

Após estes passos, as unidades estarão configuradas e prontas para o uso.

Permissões de acesso

Para o *kernel 2.6*, os gravadores utilizam os tradicionais dispositivos, de acordo com a sua controladora. Neste caso, estamos trabalhando com gravadores *IDE*, portanto, o dispositivo referenciado é */dev/hdc*:

```
$ ls -l | grep hdc
lrwxrwxrwx 1 root root         3 2007-08-11 18:35 cdr -> hdc
lrwxrwxrwx 1 root root         3 2007-08-11 18:35 cdr1 -> hdc
lrwxrwxrwx 1 root root         3 2007-08-11 18:35 cdrom1 -> hdc
lrwxrwxrwx 1 root root         3 2007-08-11 18:35 cdrw -> hdc
lrwxrwxrwx 1 root root         3 2007-08-11 18:35 cdrw1 -> hdc
lrwxrwxrwx 1 root root         3 2007-08-11 18:35 cdwriter -> hdc
lrwxrwxrwx 1 root root         3 2007-08-11 18:35 cdwriter1 -> hdc
brw-rw---- 1 root cdrom  22,   0 2007-08-11 18:35 hdc
lrwxrwxrwx 1 root root         3 2007-08-11 18:35 writer -> hdc
$ _
```

Por outro, alguns atalhos serão criados e apontados automaticamente para eles, como o *cdr*, *cdrw*, *cdwriter* e o *writer*. Tudo isto é feito automaticamente, através do subsistema *uDEV*.

Para habilitar o acesso aos dispositivos, recomenda-se criar um grupo especial denominado *burning*. Para isto, utilizem o comando...

```
# groupadd burning
```

... e definam as propriedades da conta de usuário...

```
# gpasswd -a darkstar burning
Adding user darkstar to group burning
# _
```

..., além de redefinir os grupos de acesso dos atalhos:

```
# cd /dev
# chown root.burning cdr cdrw cdwriter writer
```

Se necessário, redefina as permissões de acesso para as aplicações e ferramentas que realizam o processo de gravação:

```
# chmod 660 /usr/bin/cdda2wav
# chmod 660 /usr/bin/cdparanoia
# chmod 660 /usr/bin/cdrdao
# chmod 660 /usr/bin/cdrecord
```

Reafirmando: definam as permissões de acesso aos dispositivos somente necessário, caso a definição do grupo *burning* não funcione.

Todos estes procedimentos também poderão ser feitos à partir da ferramenta de configuração *K3bSetup*, que por sua vez acompanha o aplicativo *K3b*. Para obterem informações de como proceder, consultem na *7a. Parte: Aplicativos & Utilitários -> Imagem, som e multimídia*.

Memórias eletrônicas

Os dispositivos dotados de memória eletrônica a cada dia estão se tornando mais populares, graças à facilidade de manuseio, a portabilidade, a segurança e em especial a queda de preço que vem sofrendo. É bem mais fácil carregar todos os trabalhos em um único mini-estojo de plástico, ao invés de ter que carregar aquelas incômodas caixas de disquetes.

Dentre os dispositivos que utilizam memória eletrônica, destacam-se os *pendrives*, as câmeras digitais e seus respectivos leitores de cartões.

Os *pendrives* são pequenos estojos plásticos nos quais se encontra alojado um circuito integrado com a memória eletrônica para armazenamento de dados, além de uma entrada *USB* para conexão. Tendo em torno de *50 mm*, estes módulos de memória eletrônica também utilizam a emulação *SCSI* para serem acessados pelo sistema. Já as câmeras fotográficas digitais são dotadas de dispositivo de memória eletrônica, onde algumas destas também possuem leitores de cartões de memória. Estas também podem acessadas diretamente pelo sistema, seguindo os mesmos procedimentos que utilizamos nos *pendrives*.

O acesso para estes dispositivos segue de forma bastante similar a uma unidade qualquer, onde após tendo sido conectado em uma porta *USB*, bastará utilizar os parâmetros básicos de montagem de dispositivos. Sabendo-se de que estes dispositivos normalmente utilizam o device */dev/sda1*, basta digitar na linha de comando...

```
$ mount -t [SIST._DE_ARQUIVOS] /dev/sda1 /mnt/[PONTO_DE_MONTAGEM]
```

Lembrem-se de que será necessária a prévia criação do diretório de montagem, caso não exista no sistema e/ou não conste em */etc/fstab*.

Na existência dos dois dispositivos, normalmente o sistema reconhece o *pendrive* como *sda1* e as câmeras e/ou os leitores de cartões como *sdb1*. Em algumas circunstâncias, dependendo da ordem em que estes são conectados, estas definições de valores poderão se inverter.

Para obterem maiores informações de utilização, consultem a *2a Parte: Conhecimentos Gerais -> Unidades, partições e formatos*.

Gerenciamento de unidades e partições

Os arquivos de configuração

Os arquivos de configuração abaixo descritos são os responsáveis pela administração e manutenção das unidades e partições do sistema. Confiram as funcionalidades de cada um destes.

/etc/fstab

O arquivo base */etc/fstab* é o responsável pelas definições e montagem das unidades e partições existentes no sistema. Lá, deveremos encontrar uma estrutura similar a esta:

```
/dev/hda5     swap          swap       defaults              0  0
/dev/hda6     /             reiserfs   defaults              1  1
/dev/hda7     /home         reiserfs   defaults              1  2
/dev/hda1     /mnt/win      vfat       defaults,umask=000    1  2
/dev/cdrom    /mnt/cdrom    iso9660    noautoro,unhide       0  0
```

```
/dev/sda1      /mnt/flash      vfat        noauto                  0  0
/dev/fd0       /mnt/floppy     vfat        noauto                  0  0
devpts         /dev/pts        devpts      gid=5,mode=620          0  0
proc           /proc           proc        defaults                0  0
```

Em nosso caso, ele foi "*recondicionado*" com alguns espaços para uma melhor visualização que possibilite um entendimento mais fácil de seu conteúdo. Com um simples editor de textos e com poderes de superusuário, poderemos realizar as alterações que houver necessidade.

A *1a. coluna* indica todas as unidades disponíveis no sistema através da especificação dos seus *device drivers*. Notem que também é referenciado o sistema de arquivo virtual */proc*.

A *2a. coluna* indica os pontos de montagem, ou seja, os diretórios onde as unidades serão montadas. É por isso que não aparece nada neles quando as unidades estão desmontadas. Uma observação importante é que nada poderá ser armazenado nestes diretórios enquando as unidades/partições estiverem desmontadas.

A *3a. coluna* indica o formato das partições existentes. Nas partições de sistema se não estiverem formatadas com *ext3*, provavelmente estarão com *ReiserFS* ou ainda *ext2* para distribuições mais antigas. Para obterem maiores informações sobre os principais formatos de partições, consultem neste capítulo a seção *Sistemas de arquivos*.

Uma dica interessante está no formato de partição da unidade de disquetes (*auto*). Deveremos alterá-la para *vfat*, pois caso contrário ao tentar montar disquetes que contenham nomenclatura de arquivos longos, estes serão exibidos de modo truncados (no antigo formato *8.3*, do *MS-DOS*).

```
/dev/fd0       /mnt/floppy     vfat        noauto                  0  0
```

A *4a. coluna* indica os parâmetros de montagem. O atributo *defaults* especifica que estas serão montadas no ato da inicialização do sistema (eis o motivo pelo qual todas as partições do sistema são *defaults*). Já as demais unidades e/ou partições são passíveis de ajustes.

Dentre os principais parâmetros passíveis de edição, encontram-se:

- *auto:* montagem automática na inicialização do sistema;
- *defaults:* montagem padrão, onde o sistema utiliza parâmetros pré-definidos;
- *noauto:* montagem manual, que diferente de *auto*, somente poderá ser montado através da intervenção do administrador;
- *noexec:* define as permissões de acesso para não-executável, o que é útil para montagem de unidades *FAT*, pois seus arquivos são autodefinidos como executáveis (este sistema de arquivos não suporta o uso de permissões de acesso);
- *nouser:* define que somente o superusuário terá os privilégios de permissão (não-usuários);

- *ro ("read only"):* monta a partição no modo somente-leitura;
- *rw ("read write"):* permite a escrita da partição;
- *user:* concede permissão aos usuários para a montagem da partição;
- *umask:* permite uma definição padrão de permissões de acesso para a partição em questão;[7]
- *uid/gid:* define os usuários e os grupos na montagem de unidades *FAT*, pois este sistema de arquivos não suporta definições de usuários e grupos de acesso, além das permissões de acesso.

O parâmetro *defaults* por padrão – lógico... – utiliza determinadas opções, das quais dentre elas estão a *async*, *auto*, *dev*, *exec*, *nouser*, *rw* e *suid*.

Notem que, com estas definições, somente o superusuário poderá montar/desmontar as unidades removíveis. Caso queira que um simples usuário possa montar estas unidades, o superusuario deverá redefini-las com a inclusão do parâmetro *users*. Assim:

```
/dev/cdrom    /mnt/cdrom     iso9660     noauto,users,ro,unhide  0  0
/dev/fd0      /mnt/floppy    vfat        noauto,users            0  0
```

No final das definições de cada linha, acreditamos que os usuários em questão devem estar se perguntando o seguinte: "*pra quê serve aqueles dois 'zerinhos' ali no canto esquerdo?*". Estes últimos campos – a *5a. coluna* – são respectivamente as *flags* que indicam as prioridades respectivas dos comandos *dump* e *fsck*. De acordo com o valor, estes programas realizarão a checagem das partições periodicamente.

Os principais valores são:

- *0:* não realiza a checagem;
- *1, 2, 3...:* realiza a checagem, que por sua vez dependendo do valor, existem diferentes graus de prioridades (onde *1* é o maior).

/ETC/MTAB

O arquivo */etc/mtab* apenas exibe a situação atual das partições montadas pelo sistema. Da mesma forma que o arquivo */etc/fstab*, este também se encontra organizado diferente do original para facilitar o entendimento.

```
/dev/hda6     /              reiserfs   rw                  0  0
/dev/hda7     /home          reiserfs   rw                  0  0
/dev/hda1     /usr/win       vfat       rw,umask=000        0  0
devpts        /dev/pts       devpts     rw,gid=5,mode=620   0  0
proc          /proc          proc       rw                  0  0
usbfs         /proc/bus/usb  usbfs      rw                  0  0
```

7 As partições *FAT32* somente podem ser acessadas pelo superusuário. Para contornar esta situação, redefinam as permissões de acesso para que os usuários possam também acessá-las. Para obterem maiores informações, consultem nesta parte o capítulo *Contas de usuário e grupos de acesso*.

Dentre os diferenciais apresentados, notem que a partição *swap* (apesar de montada) não se encontra especificada, além das definições diferenciadas das *flags* dos utilitários *dump* e *fsck*.

Conclusão

Felizmente a maioria das unidades de armazenamento são reconhecidas automaticamente pelos sistemas *GNU/Linux*. Exceto alguns periféricos específicos, serão poucos os inconvenientes que poderão surgir. Nestas circunstâncias, recomendamos inicialmente certificarem-se de que estes periféricos realmente se encontram em perfeito funcionamento. &;-D

V. Teclado e mouse

Introdução

Por serem dispositivos simples, o *mouse* e o teclado somente recebem a devida atenção quando não funcionam ou funcionam de forma inadequada. Os maus perfis de configurações não só impossibilitam utilizar estes periféricos corretamente, como limitam tanto a máquina quanto o usuário, de todos os recursos necessários para uma boa atividade.

São muitos os aspectos gerais de ajuste e configurações para estes periféricos, porém somente manteremos as mais básicas e necessárias instruções que possibilite colocá-los em operação.

O teclado

Na verdade, não existem grandes inconvenientes na configuração do teclado. Desde que o sistema seja instalado normalmente, e os utilitários de configuração do modo gráfico estejam corretamente habilitados, o teclado funcionará sem maiores inconvenientes. Em alguns casos, dependendo do ambiente gráfico, talvez será necessário realizar algumas configurações em seus respectivos painéis de configuração para o perfeito funcionamento.

Mas como fazer para que o sistema reconheça corretamente o modelo do novo teclado que recentemente acabamos de adquirir? Ou aquele em que houve uma necessidade de troca, pois o antigo acabara de pifar?

Configurações básicas

Modo gráfico

Ao executar os utilitários de configuração do servidor gráfico, não só a placa de vídeo e o *monitor* são configurados, como também o *mouse* e o teclado. Na seção *keyboard*, responderemos uma série de perguntas feita pelo utilitário, que por sua vez de acordo com estas informações, serão definidos todos os parâmetros necessários para que o teclado funcione perfeitamente no ambiente gráfico. Mas em alguns ambientes gráficos – especialmente o *KDE* – poderão ser necessário alguns ajustes extras.

Poderemos também ajustá-lo através do utilitário *xorgcfg* e, na seção *Keyboard*, ajustá-lo, tal como fazemos com o *mouse* (veja adiante).

Para obterem maiores informações de como configurar o *mouse* e o teclado no modo gráfico, consultem nesta parte, o capítulo *Vídeo – placa de vídeo, monitor e modo gráfico*; para ajustar estes periféricos no *KDE*, consultem a *6a. Parte: Ambientes Gráficos -> As ferramentas de ajustes*.

Modo texto

Para ajustarmos o mapa do teclado, editem o arquivo */etc/rc.d/rc.keymap*.

```
# mcedit /etc/rc.d/rc.keymap
```

Lá encontraremos as seguintes definições:

```
#!/bin/sh
# Load the keyboard map.  More maps are in /usr/share/kbd/keymaps.
if [ -x /usr/bin/loadkeys ]; then
 /usr/bin/loadkeys [MAPA_DO_TECLADO]
fi
```

Definam o mapa específico ao modelo do teclado em questão na definição *loadkeys*. Em nosso caso deveremos utilizar...

```
 /usr/bin/loadkeys br-abnt2.map
```

Por último, certifiquem-se de que este arquivo possui o atributo de execução. Caso contrário (algo raro), utilizem o comando...

```
# chmod +x rc.keymap
```

Intervenções mais freqüentes

Habilitando as famosas "teclas do Windows"

Para habilitarmos as "*teclas do Windows*", deveremos editar o arquivo */etc/X11/xinit/.Xmodmap* e acrescentar as seguintes linhas:

```
keycode 115 = Meta_L
```

```
add mod4 = Meta_L
keycode 116 = Meta_R
add mod4 =Meta_R
```

Rodem o *xmodmap* para estas alterações serem adicionadas:

```
# xmodmap .Xmodmap
```

Aumentando a velocidade

Para aumentar a velocidade, utilizaremos o comando *kbdrate*.

Sintaxe:

```
# kdbrate -d [DELAY] -r [RATE]
```

Onde:

- *-d [DELAY]:* define o tempo de espera para a repetição da tecla pressionada, onde o intervalo vai de *250* a *1000 ms*;
- -r [RATE]: define a taxa de repetição (velocidade), onde o intervalo vai de *6* a *24 ms*.

Exemplo:

```
# kdbrate -d 1000 -r 20
```

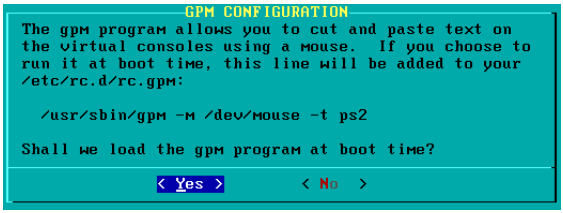
Os valores destes parâmetros poderão variar conforme assim preferir.

O mouse

Da mesma forma que o teclado, o *mouse* é configurado automaticamente durante a instalação. Mas vamos supor que, depois de instalado o sistema, o periférico passou a dar problemas e tempos depois resolvemos trocá-lo? Ao comprar um novo, não havia o modelo *X* na loja, mas o computador suportava o modelo *Y* disponível, e assim mesmo resolvemos adquiri-lo...

O GPM

O *GPM* é um servidor para o *mouse* em modo texto. Na instalação do *Slackware*, seus serviços são ativados durante a realização das etapas de configuração do sistema.



Suas definições gerais estão centralizadas no *script /etc/rc.d/rc.gpm*:

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

```
# Start/stop/restart the GPM mouse server:

if [ "$1" = "stop" ]; then
  echo "Stopping gpm..."
  /usr/sbin/gpm -k
elif [ "$1" = "restart" ]; then
  echo "Restarting gpm..."
  /usr/sbin/gpm -k
  sleep 1
  /usr/sbin/gpm -m /dev/mouse -t [MOUSE]
else # assume $1 = start:
  echo "Starting gpm:  /usr/sbin/gpm -m /dev/mouse -t [MOUSE]"
  /usr/sbin/gpm -m /dev/mouse -t [MOUSE]
fi
```

Os parâmetros de configuração do *mouse* são referidos na seguinte linha:

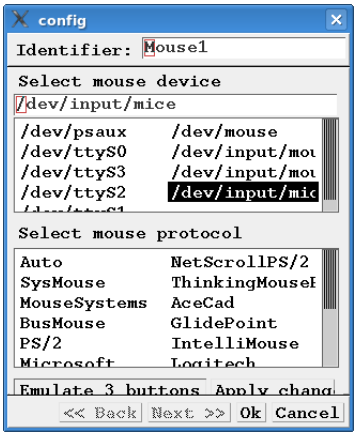
```
/usr/sbin/gpm -m /dev/mouse -t [MOUSE]
```

Basta substituir *[MOUSE]* pelo parâmetro do periférico em questão. Se desejar ter menos trabalho para esta configuração, basta utilizar o *script mouseconfig* do *Slackware* (onde fornece os parâmetros do *GPM* para serem selecionados). Vejam à seguir como utilizá-lo.

Configurações básicas

Modo gráfico

Para configurar o *mouse* de forma que funcione corretamente no modo gráfico, basta executar o utilitário *xorgcfg* e na seção *Mouse*, ajustá-lo.



Lembre-se que ao salvarem estas opções, o utilitário irá regravar o arquivo */etc/X11/xorg.conf*. Caso tenham feito alguma alteração em especial, sugerimos que realizem-nas novamente ou faça uma cópia de segurança da configuração antiga, consulte os parâmetros da nova

configuração na seção *"Core Pointers a InputDevice section"*. Veja um pequeno exemplo abaixo:

```
# ***********************************************************************
# Core Pointers InputDevice section
# ***********************************************************************

Section "InputDevice"

# Identifier and driver

    Identifier  "Mouse1"
    Driver      "mouse"
    Option "Protocol"    "IMPS/2"       # PS/2 Mouse
    Option "Device"      "/dev/input/mice"
```

Editem a antiga configuração com base nestes dados e depois copiem de volta para o seu local original. Ao iniciar o ambiente gráfico, as alterações entrarão em vigor e o *mouse* em questão estará funcionando corretamente.

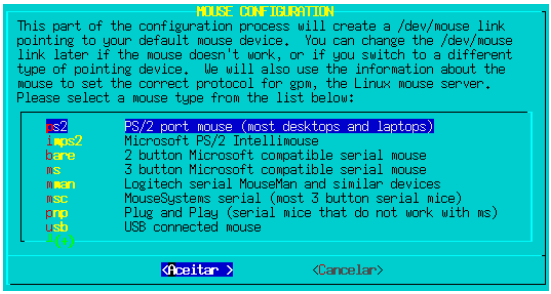
Se tivermos acesso aos parâmetros de configuração do novo *mouse*, bastará editar o arquivo normalmente.

Modo texto

Simplesmente digitem na linha de comando...

```
# mouseconfig
```

... ou utilizem a ferramenta *PkgTools*, opção *Setup -> Mouse*, e selecione o padrão do novo *mouse* adquirido.



Simples, não? Porém devem-se lembrar de que ajustaram os parâmetros do *mouse* em modo texto. Por isto, será necessário realizar outros ajustes para configurar o *mouse* no ambiente gráfico.

Intervenções mais freqüentes

Mouses ópticos...

Devido a queda de custo, os *mouses ópticos* estão cada vez mais sendo adotados por usuários *desktops* em virtude de suas vantagens em comparação aos *mouses* tradicionais.

Os sistemas *GNU/Linux* suportam normalmente estes periféricos, bastando configurá-los normalmente de acordo com a interface utilizada (*PS/2*, *USB*, etc.). No caso de *mouses* ópticos que utilizam a interface *PS/2*, configurem-no para utilizar o protocolo *IMPS/2*.

... e com scroll lock (com rodinhas)

O *mouse* deverá ser configurado para utilizar o protocolo *IMPS/2*. Na seção *Core Pointers InputDevice section* (*mouse*), deverá ser adicionada a linha...

```
# ***********************************************************************
# Core Pointers InputDevice section
# ***********************************************************************

Section "InputDevice"

# Identifier and driver

    Identifier  "Mouse0"
    Driver      "mouse"
    Option      "Protocol"   "IMPS/2"
    Option      "Device"   "/dev/mouse"
    Option      "ZaxisMapping"   "4 5"

EndSection
```

... para que no ambiente tenhamos as funcionalidades do *scroll lock* do *mouse*. À partir da versão *6.9.x* do servidor gráfico *X.org*, este passo é desnecessário, visto que o *scroll lock* é habilitado automaticamente.

Com conexão USB

Para habilitarmos os *mouses* com conexão *USB*, deveremos configurá-lo previamente conforme as suas especificações (botões, *scroll lock*, etc.). Logo em seguida, deveremos carregar os seguintes módulos do *kernel*:

```
modprobe hid
modprobe usbmouse
modprobe usb-uhci
```

Estas linhas já se encontram descritas no arquivo de configuração */etc/rc.d/rc.modules*, na seção *USB Host Controllers*...

```
### USB Host Controllers:
```

```
# Universal Host Controller Interface (Intel standard):
#/sbin/modprobe usb-uhci
```

... e *USB device support*:

```
### USB device support:
# (Note that once you've loaded USB hub support most USB devices will
# trigger the kernel to load their modules automatically)
# USB "Human Interface Device" driver; handles most USB mice, joysticks,
# gamepads, steering wheels, keyboards, trackballs and digitizers.
#/sbin/modprobe hid
# Simple HIDBP USB mouse driver, if hid doesn't work:
#/sbin/modprobe usbmouse
```

... bastando apenas descomentá-las.

Na maioria dos casos, estes passos são desnecessários, já que o próprio sistema realiza o processo de detecção do periférico. Mas não em todos...

Conclusão

Normalmente durante a instalação, estes periféricos são configurados corretamente, não acarretando maiores problemas. O inconveniente está justamente na substituição dos mesmos, onde deveremos redefinir seus parâmetros de configuração para o reconhecimento dos periféricos. Opte por utilizar as ferramentas disponibilizadas pelo *Slackware*, como o *mouseconfig* e ou *setconsolefonts*, além de outras necessárias (*xorgconfig*, ferramentas dos ambientes gráficos), etc. Caso não tenha obtido sucesso, uma boa pesquisa no *Google*, tendo como chave de busca as referências dos periféricos, tipo e o texto '*Linux*', irá ajudar bastante! &;-D

VI. O kernel Linux

Introdução

- ✔ <http://www.kernel.org/>.

O *kernel* é o coração do sistema; é ele quem "*coordena*" todo o trabalho de gerenciamento e acesso aos principais recursos do sistema. Ao acessarmos a memória principal, ao executarmos aplicações, ao exibirmos informações no vídeo, ao gravarmos arquivos, enfim, todos os eventos inerentes de um sistema operacional, acreditem: é o *kernel* que entra em cena.

Neste capítulo, iremos conhecer um pouco sobre o *kernel Linux*, suas características e particularidades, além dos conhecimentos e instruções básicas necessárias para a realização do procedimento de compilação.

Considerações básicas

As particularidades do kernel

O *kernel Linux* possui interessantes características e particularidades técnicas, que por sua vez é associado diretamente com as distribuições; mas ainda assim, possui seus elementos gerais, já que cada sistema apenas realiza customizações de seus recursos. Por isto, optamos por inserir estas informações na *1a. Parte: Os sistemas GNU/Linux -> O Linux*.

O sistema de controle de versões

Tecnicamente a versão de desenvolvimento e a versão estável do *kernel*.

A versão de desenvolvimento é composta por um número de série cujo o *2o.* termo é sempre ímpar (*2.1.x*, *2.3.x*, *2.5.x*), onde são acrescentadas melhorias e modificações para que no futuro venha a ser largamente utilizado. Ao chegar a um certo grau de amadurecimento, este *kernel* se tornará a versão estável, onde é alterada a sua composição numérica para uma nova série, onde este *2o.* termo passa a ser par (*2.2.x*, *2.4.x*, *2.6.x*).

À partir da versão estável, inicia-se novamente o trabalho para desenvolvimento de uma nova versão do *kernel*. Este processo acaba gerando um ciclo de desenvolvimento contínuo:

- A versão em desenvolvimento gera a versão estável – *2.1.x -> 2.2.x*;
- A nova versão estável torna-se a base para a criação da versão em desenvolvimento – *2.2.x -> 2.3.x*;

- E assim, é continuado o ciclo de aperfeiçoamento... – *2.3.x -> 2.4.x -> 2.5.x -> 2.6.x -> ...*

Dependendo das circunstâncias, algumas modificações são incluídas no *kernel* estável para que ele possa suportar novas tecnologias de caráter essencial, procedimento este que geralmente é implementado no *kernel* de desenvolvimento para que esteja presente no novo *kernel* estável. Por exemplo, a partir da versão *2.0.31*, o *kernel* passou a suportar nomes extensos do *Windows* nas partições *VFAT*; já acima da versão *2.4.16*, passou a suportar o sistema de dados *ReiserFS*; e assim por diante...

A localização de seus componentes

Conforme as definições da norma *FHS*, que por sua vez define a estrutura de arquivos e diretórios, o *kernel* e os arquivos de inicialização estão armazenados no diretório */boot*. Lá, encontraremos o seguinte:

```
$ cd /boot
$ ls -l
total 17103
lrwxrwxrwx 1 root root      37 2007-08-03 10:00 README.initrd ->
/usr/doc/mkinitrd-1.1.2/README.initrd
lrwxrwxrwx 1 root root      27 2007-08-07 23:53 System.map -> System.map-
generic-2.6.21.5
-rw-r--r-- 1 root root  795880 2007-06-19 17:18 System.map-generic-2.6.21.5
-rw-r--r-- 1 root root  813610 2007-06-19 16:53 System.map-generic-
smp-2.6.21.5-smp
-rw-r--r-- 1 root root 1232918 2007-06-19 17:23 System.map-huge-2.6.21.5
-rw-r--r-- 1 root root 1252098 2007-06-19 16:58 System.map-huge-smp-2.6.21.5-
smp
-rw-r--r-- 1 root root     512 2007-08-03 10:17 boot.0300
-rw-r--r-- 1 root root     209 2007-08-03 10:17 boot_message.txt
lrwxrwxrwx 1 root root      23 2007-08-07 23:52 config -> config-
generic-2.6.21.5
-rw-r--r-- 1 root root   72738 2007-06-19 17:18 config-generic-2.6.21.5
-rw-r--r-- 1 root root   72764 2007-06-19 16:53 config-generic-smp-2.6.21.5-
smp
-rw-r--r-- 1 root root   72643 2007-06-19 17:23 config-huge-2.6.21.5
-rw-r--r-- 1 root root   72669 2007-06-19 16:58 config-huge-smp-2.6.21.5-smp
-rw-r--r-- 1 root root    5040 2007-06-10 03:09 diag1.img
drwxr-xr-x 9 root root     384 2007-08-08 10:46 initrd-tree/
-rw-r--r-- 1 root root  487394 2007-08-08 10:46 initrd.gz
-rw------- 1 root root   42496 2007-08-08 10:48 map
lrwxrwxrwx 1 root root      24 2007-08-07 23:53 vmlinuz -> vmlinuz-
generic-2.6.21.5
-rw-r--r-- 1 root root 1937944 2007-06-19 17:18 vmlinuz-generic-2.6.21.5
-rw-r--r-- 1 root root 2087960 2007-06-19 16:53 vmlinuz-generic-smp-2.6.21.5-
smp
-rw-r--r-- 1 root root 4097784 2007-06-19 17:23 vmlinuz-huge-2.6.21.5
-rw-r--r-- 1 root root 4417112 2007-06-19 16:58 vmlinuz-huge-smp-2.6.21.5-smp
$ _
```

A imagem do *kernel* corresponde ao arquivo *vmlinuz*, ao passo que o arquivo *System.map* é utilizado pelas ferramentas de manipulação de

módulos. Já o arquivo *config* refere-se as opções de configuração padrão da distribuição que foram utilizados na compilação do *kernel*.

O código-fonte do *kernel Linux* deverá estar disponibilizado em */usr/src*, sob o diretório *linux* (que na verdade, é um atalho apontando para *linux-[VERSÃO]*, que por sua vez é o diretório real). Dentro deste diretório, poderemos ver a seguinte árvore de diretórios:

```
$ cd /usr/src/linux
$ ls -l
total 670
-rw-r--r--  1 root root  18693 2007-06-11 15:37 COPYING
-rw-r--r--  1 root root  90999 2007-06-11 15:37 CREDITS
drwxr-xr-x 63 root root   6288 2007-06-11 15:37 Documentation/
-rw-r--r--  1 root root   1262 2007-06-11 15:37 Kbuild
-rw-r--r--  1 root root  84150 2007-06-11 15:37 MAINTAINERS
-rw-r--r--  1 root root  50303 2007-06-11 15:37 Makefile
-rw-r--r--  1 root root 379343 2007-06-19 16:47 Module.symvers
-rw-r--r--  1 root root  16930 2007-06-11 15:37 README
-rw-r--r--  1 root root   3119 2007-06-11 15:37 REPORTING-BUGS
drwxr-xr-x 27 root root    656 2007-06-11 15:37 arch/
drwxr-xr-x  2 root root    440 2007-06-19 16:48 block/
drwxr-xr-x  2 root root   1320 2007-06-19 16:48 crypto/
drwxr-xr-x 66 root root   1688 2007-06-19 16:48 drivers/
drwxr-xr-x 62 root root   3496 2007-06-19 16:48 fs/
drwxr-xr-x 43 root root   1176 2007-08-03 10:04 include/
drwxr-xr-x  2 root root    416 2007-06-19 16:48 init/
drwxr-xr-x  2 root root    336 2007-06-19 16:48 ipc/
drwxr-xr-x  5 root root   2456 2007-06-19 16:48 kernel/
drwxr-xr-x  5 root root   2672 2007-06-19 16:48 lib/
drwxr-xr-x  2 root root   1512 2007-06-19 16:48 mm/
drwxr-xr-x 38 root root   1128 2007-06-19 16:48 net/
drwxr-xr-x  9 root root   1624 2007-06-19 16:48 scripts/
drwxr-xr-x  4 root root    320 2007-06-19 16:48 security/
drwxr-xr-x 18 root root    600 2007-06-19 16:48 sound/
drwxr-xr-x  2 root root    192 2007-06-19 16:48 usr/
$ _
```

Além do subdiretório específico para a documentação (*Documentation*), existem outros documentos para consulta situados no diretório principal, como o *README*, o *REPORTING-BUGS*, o *COPYING* e o *CREDITS*.

Por último, todos os seus módulos serão encontrados em...

```
$ cd /lib/modules/2.6.21.5/kernel
$ ls -l
total 2
drwxr-xr-x  3 root root   72 2007-06-19 17:18 arch/
drwxr-xr-x  2 root root   80 2007-06-19 17:18 crypto/
drwxr-xr-x 39 root root  976 2007-06-19 17:18 drivers/
drwxr-xr-x 49 root root 1336 2007-06-19 17:18 fs/
drwxr-xr-x  2 root root  192 2007-06-19 17:18 lib/
drwxr-xr-x  2 root root   88 2007-06-12 18:56 misc/
drwxr-xr-x 27 root root  688 2007-06-19 17:18 net/
drwxr-xr-x  2 root root  144 2007-06-19 17:18 security/
drwxr-xr-x 10 root root  304 2007-06-19 17:18 sound/
```

```
$ _
```

Poderemos armazenar diversas versões do *kernel*, apenas atribuindo aos seus respectivos diretórios o nome e a versão corrente destes para que, na eminência de uma necessidade, estes estejam sempre disponíveis.

A EQUIPE DE DESENVOLVIMENTO

Conforme brevemente comentado na *1a. Parte: Os sistemas GNU/Linux -> O Linux*, o *kernel* é desenvolvido graças a ajuda de milhares de colaboradores, que tendo em mãos o seu código-fonte, ajudam nas mais diversas formas: sejam aperfeiçoando o próprio código, analizando, comunicando erros, dando sugestões, etc., enfim, colaborações que hoje o tornam um dos maiores projetos de código-aberto atualmente.

Encabeçado por *Linus Torvalds* e com grande apoio de seu braço direito *Alan Cox*, o *kernel Linux* possui um grupo de desenvolvedores, subdividido por uma hierarquia democrática, onde as sugestões enviadas são avaliadas pelos membros da equipe, porém em casos de opiniões adversas, somente os líderes de cada hierarquia é que dão a palavra final.

Tendo um grupo de desenvolvedores formado por programadores de diversos países do mundo inteiro, fica a pergunta no ar: "*será que não tem nenhum representante brasileiro neste grupo*"? Ah, tem sim! Este representante é o gaucho *Martelo Tosatti*, que faz parte do grupo de desenvolvedores do *kernel Linux*. Há pouco tempo, ele era somente o responsável pela manutenção da versão corrente do *kernel*, porém em virtude de um convite do próprio *Linus Torvalds*, *Tosatti* passou a desenvolver outros componentes importantes deste sistema operacional.

OS MÓDULOS DO KERNEL

Os módulos do *kernel* são componentes que auxiliam a interação do *kernel* com os dispositivos do sistema. Numa comparação, poderemos dizer que eles nos sistemas *GNU/Linux* possuem recursos e funcionalidades eqüivalentes aos *drivers* do tradicional *Windows*.

A MANUTENÇÃO

Dada a importância dos módulos, iremos conhecer as principais ferramentas para a sua manipulação, bem como seus recursos e métodos de utilização para obter um melhor aproveitamento nas atividades de administração e manutenção do *kernel*.

LSMOD

Lista os módulos que se encontram carregados no sistema.

Para uma utilização simples e rápida, bastará digitarmos...

```
# lsmod
Module                  Size  Used by
nls_iso8859_1           3968  0
nls_cp437               5632  0
vfat                   10240  0
fat                    45340  1 vfat
via                    39552  3
drm                    69652  4 via
snd_seq_dummy           2692  0
snd_seq_oss            28032  0
snd_seq_midi_event      5888  1 snd_seq_oss
snd_seq                42576  5 snd_seq_dummy,snd_seq_oss,snd_seq_midi_event
snd_pcm_oss            38432  0
snd_mixer_oss          13824  1 snd_pcm_oss
nls_utf8                1792  1
ntfs                  213312  1
sg                     25756  0
capability              3336  0
commoncap               5376  1 capability
lp                      9800  0
parport_pc             23844  1
parport                30152  2 lp,parport_pc
pcspkr                  2304  0
psmouse                34440  0
usb_storage            79040  0
bt878                   8872  0
tuner                  61352  0
tvaudio                21788  0
pata_via                8324  0
serio_raw               5124  0
bttv                  168436  1 bt878
video_buf              20228  1 bttv
ir_common              29956  1 bttv
compat_ioctl32          1280  1 bttv
i2c_viapro              7828  0
i2c_algo_bit            7176  1 bttv
btcx_risc               3976  1 bttv
tveeprom               13712  1 bttv
i2c_core               17168  6
tuner,tvaudio,bttv,i2c_viapro,i2c_algo_bit,tveeprom
ata_generic             5252  0
ehci_hcd               29964  0
uhci_hcd               21004  0
videodev               25344  1 bttv
v4l2_common            14848  3 tuner,bttv,videodev
v4l1_compat            13700  2 bttv,videodev
via_rhine              20360  0
mii                     4736  1 via_rhine
shpchp                 29204  0
snd_via82xx            22296  0
gameport               10760  1 snd_via82xx
snd_ac97_codec         96292  1 snd_via82xx
ac97_bus                2048  1 snd_ac97_codec
snd_pcm                65160  3 snd_pcm_oss,snd_via82xx,snd_ac97_codec
snd_timer              17540  2 snd_seq,snd_pcm
```

```
snd_page_alloc          7432  2 snd_via82xx,snd_pcm
snd_mpu401_uart         6272  1 snd_via82xx
snd_rawmidi            18080  1 snd_mpu401_uart
snd_seq_device          6540  4 snd_seq_dummy,snd_seq_oss,snd_seq,snd_rawmidi
evdev                   7936  2
via_agp                 8192  1
agpgart                26928  2 drm,via_agp
snd                    41956  11
snd_seq_oss,snd_seq,snd_pcm_oss,snd_mixer_oss,snd_via82xx,snd_ac97_codec,snd_p
cm,snd_timer,snd_mpu401_uart,snd_rawmidi,snd_seq_device
soundcore               5728  1 snd
reiserfs              234496  2
# _
```

Por possuir uma sintaxe básica, a saída do resultado será bem simples, sendo apresentada uma listagem dos módulos carregados pelo sistema.

MODPROBE

O responsável pelo carregamento dos módulos disponíveis no sistema.

Sintaxe:

```
# modprobe [MÓDULO]
```

Para este comando existem parâmetros e funcionalidades extras, porém na maioria das vezes apenas o utilizaremos para o carregamento de módulos.

Por exemplo, para carregar os módulos *USB* para habilitar o *mouse*...

```
# modprobe usbmouse
```

INSMOD

Tal como o *modprobe*, o *insmod* também carrega os módulos do sistema.

Sintaxe:

```
# insmod [MÓDULO]
```

Porém, sua diferença está no fato de que o *modprobe* carrega as pendências necessárias para o funcionamento do módulo, ao passo que o *insmod* tão somente carrega o módulo. Veja um exemplo prático:

```
# insmod usb-storage
Using /lib/modules/2.4.22/kernel/drivers/usb/storage/usb-storage.o.gz
# _
```

Para certificarmos de que foi corretamente carregado...

```
# lsmod | grep usb
usb-storage            65536   0  (unused)
usbcore                58400   1  [usb-storage uhci ehci-hcd]
# _
```

Dependendo das circunstâncias, talvez seja necessário o carregamento forçado de um determinado módulo. Para isto, utilizem...

```
# insmod -f [MÓDULO]
```

Isto ocorre muito quando há necessidade de carregar um módulo compilado de uma versão diferente do *kernel*. Apesar desta operação não ser recomendável, poderá ser a única saída para diversas situações.

DEPMOD

Checa as pendências dos módulos.

Sintaxe:

```
# depmod [PARÂMETROS]
```

Onde:

- *-a:* checa as pendências de módulo necessárias pelo *kernel*;
- *-b:* define o caminho de um módulo específico que não esteja armazenado no diretório padrão (*/lib/modules/*);
- *-e:* verifica se determinado módulo é compatível com o *kernel* atual.

Para realizarmos uma simples checagem, deveremos utilizar...

```
# depmod -a
```

Os resultados deste comando são armazenados em um arquivo-texto chamado *modules.dep*, situado em seu diretório padrão.

MODINFO

Exibe informações básicas sobre determinados módulos.

Sintaxe:

```
# modinfo [PARÂMETROS] [MÓDULO]
```

Onde:

- *-a:* exibe o autor (desenvolvedor);
- *-d:* exibe um breve resumo;
- *-l:* exibe a licença;
- *-p:* exibe parâmetros específicos.

Exemplo:

```
# modinfo -d /lib/modules/2.4.22/kernel/drivers/hotplug/pci_hotplug.o.gz
"PCI Hot Plug PCI Core"
# _
```

Experimentem os demais parâmetros e vejam por sí os resultados.

RMMOD

Sem grandes mistérios, o comando *rmmod* foi feito exclusivamente para

descarregar um determinado módulo do sistema.

Sintaxe:

```
# rmmod [MÓDULO]
```

Lembrem-se de que estaremos apenas DESCARREGANDO o módulo da memória, e não EXCLUINDO. Poderemos então ficar sossegados...

Segue um simples exemplo para descarregar o módulo responsável pelo acesso das memórias *flash* (*Pendrive*):

rmmod usb-storage

O manuseio de módulos pré-compilados

Em muitos casos são disponibilizados módulos pré-compilados de determinados dispositivos, onde os mesmos deverão ser ativados manualmente para que o sistema possa reconhecê-lo. Exemplos típicos de periféricos que necessitam de ter tais módulos são os *softmodens*.

Para realizarmos esta operação, deveremos inicialmente obter o módulo (lógico). Este deverá ter uma extensão *.o* ou *.o.gz* no final de sua nomenclatura. Em seguida, deveremos copiá-lo para o diretório-padrão do sistema. Neste caso, o endereço-destino será...

```
# cp [MÓDULO] /lib/modules/[VERSÃO]/kernel
```

Em seguida, para ativarmos deveremos evocar o comando *modprobe*...

```
# modprobe /lib/modules/[VERSÃO]/kernel/[MÓDULO]
```

Por último, bastará apenas atualizar a listagem de módulos do sistema. Utilizem na linha de comando...

```
# depmod -a
```

Incluam o comando *modprobe* e sua sintaxe no arquivo */etc/rc.modules* para que este módulo esteja sempre disponível a partir da inicialização.

A localização dos arquivos

Os módulos encontram-se localizados por padrão no diretório...

```
$ cd /lib/modules/2.6.21.5/kernel/
$ ls -l
total 2
drwxr-xr-x  3 root root   72 2007-06-19 17:18 arch/
drwxr-xr-x  2 root root   80 2007-06-19 17:18 crypto/
drwxr-xr-x 39 root root  976 2007-06-19 17:18 drivers/
drwxr-xr-x 49 root root 1336 2007-06-19 17:18 fs/
drwxr-xr-x  2 root root  192 2007-06-19 17:18 lib/
drwxr-xr-x  2 root root   88 2007-06-12 18:56 misc/
drwxr-xr-x 27 root root  688 2007-06-19 17:18 net/
drwxr-xr-x  2 root root  144 2007-06-19 17:18 security/
drwxr-xr-x 10 root root  304 2007-06-19 17:18 sound/
$ _
```

..., onde cada subdiretório armazena cada um de acordo com sua categoria.

Todos os módulos, para serem carregados diretamente na inicialização do sistema, deverão estar especificados no arquivo */etc/rc.d/rc.modules*, onde normalmente basta apenas descomentar os comandos abaixo descritos para carregá-los.[8] Mas antes, certifiquem-se de que as ferramentas do sistema para a auto-detecção não fizerem o carregamento automático.

Para obterem maiores informações sobre este arquivo de configuração, consultem a *2a. Parte: Conhecimentos Gerais -> O sistema de inicialização*.

Observações finais

Os módulos do *kernel* estão entre as maiores fontes de insucesso na utilização do novo *kernel* personalizado. São vários os possíveis erros de ajustes e configurações que trazem diversos e inconvenientes transtornos. Diversos dispositivos e periféricos passam a ser suportados de forma incorreta e instáveis ou até mesmo, na maioria dos casos, deixam de ser suportados, ocasionando também o travamento geral do sistema. Este é um dos principais motivos pelo qual merecem atenção especial, onde o conhecimento de ferramentas e procedimentos são fundamentais para o sucesso das operações afins realizadas.

O processo de compilação

Sobre o processo de compilação

A possibilidade de realizar a compilação e/ou recompilação do *kernel* está entre as maiores vantagens para a adoção de sistemas operacionais livres como o *GNU/Linux* e *BSDs* em comparação aos outros proprietários. Mas especificamente para nós, simples usuários, quais motivos poderemos ter para realizar este delicado procedimento?

O principal motivo

Em virtude da imensa variedade máquinas com perfis de *hardwares* existentes, os sistemas geralmente reconhecem a grande maioria destes, porém ainda existem muitos componentes que, por mais variados motivos, não são suportados pelo *kernel* ou seu suporte não se encontram habilitados. O mesmo se dá para determinados programas; no geral, muitos destes elementos deverão estar habilitadas no *kernel* para que

8 Para facilitar a nossa vida, encontra-se incluso no *kernel* uma grande quantidade de módulos pré-compilados para o suporte a diversos periféricos, bastando apenas descomentar as linhas referentes. Por sua vez, estas linhas estão subdivididas por várias seções, os quais facilitarão muito a nossa procura.

esses programas possam funcionar corretamente.

Como esta parte trata efetivamente da descrição dos periféricos e os procedimentos relacionados para o seu ajuste e configuração, como também dos *softwares* necessários para o seu perfeito funcionamento, este será o principal motivo pelo qual iremos focar as instruções gerais referentes ao processo de compilação do *kernel*.

Entre outros aspectos interessantes

Existem outros aspectos interessantes que também devem ser observados...

De acordo com cada compilação de *kernel* feita pelas distribuições, existem recursos diversos que são ativados por padrão para atender a grande maioria dos usuários; porém, por mais "*exóticos*" que sejam tais usuários, a grande maioria dos recursos presentes serão desnecessários para seu uso. Nestas condições, a (re)compilação do *kernel* proverá uma base de recursos mais enxuta para o equipamento ao qual o sistema se encontra instalado. Basta apenas personalizá-lo de acordo com a configuração disponível para ter um novo *kernel*: mais simples, limpo, rápido, enxuto e mais eficiente.

Em virtude da existência de diversas arquiteturas, a grande maioria das distribuições fornecem o *kernel* pré-compilado para atender as mais variadas existentes. Por exemplo, o *kernel* do *Slackware* é compilado para suportar processadores a partir do *i486*, ao passo que as distribuições *friendly-users*, com seus vastos recursos, suportam arquiteturas *i585* em diante em virtude da demanda de processamento e de *hardware*. O *kernel* também pode ser otimizado para suportar os recursos específicos de uma única plataforma, habilitando suas extensões e otimizando-o somente para aquela arquitetura, para que o ganho de desempenho seja visível.

A possibilidade de ajustar o *kernel* para finalidades especiais também o torna altamente compatível com as tecnologias e requisitos atuais. Suporte a dispositivos exóticos e periféricos "*desconhecidos*" são também garantidos com algumas alterações na configuração do *kernel* e respectiva utilização.

Para os especialistas, técnicos e entusiastas, a compilação do *kernel* é uma ótima oportunidade de aprendizado para melhor conhecer os fundamentos básicos do funcionamento de um sistema operacional.

Os preparativos iniciais

Como qualquer outra operação que envolve riscos de perdas de dados, será necessária a realização de alguns preparativos iniciais que visam dar assistência ao desenvolvimento do processo e ao mesmo tempo resguardar o usuário contra possíveis falhas e sinistros que poderão ocorrer sob os mais variados motivos. Por se tratar da compilação do

kernel – justamente o núcleo do sistema operacional – toda a atenção necessária deverá ser dada.

Cópia de segurança dos dados do sistema

Na verdade, não são muito comuns os casos em que os processos de compilação do *kernel* feitos incorretamente produzem perdas de dados, porém, de acordo com o nosso nível de conhecimento e a possibilidade de não termos mais acesso ao sistema, será mais cômodo a utilização desta operação para resgate. Caso algo não dê certo, bastará apenas realizarmos a reinstalação do sistema, restaurando todas as definições anteriores e os dados previamente salvaguardados.

Para esta atividade, consultem a *2a. Parte: Conhecimentos Gerais -> Manipulação de arquivos e diretórios*, para obterem maiores informações para a realização da cópia de segurança com segurança e praticidade.[9]

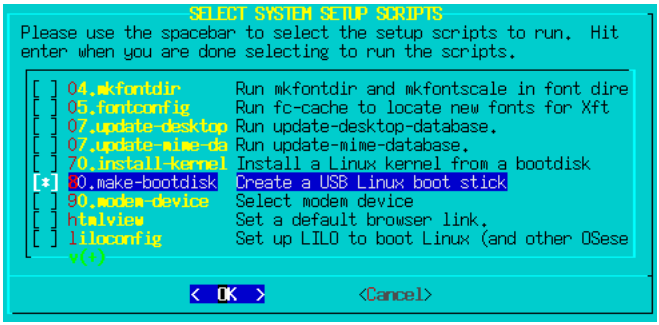
A elaboração de pendrives de inicialização

A utilização de *pendrives* de inicialização é necessária especialmente quando ocorrem erros no processo de compilação que possam resultar a não inicialização do sistema, o qual somente poderá ser feito utilizando-se este recurso. Infelizmente, este tipo de ocorrência é bastante comum, o que torna praticamente indispensável a sua disponibilidade.

Para isto, inicializem através da linha de comando o *Pkgtool*...

```
# pkgtool
```

... e, na seção *Setup*, marquem a opção *Make Bootdisk*:



Sem maiores mistérios, basta seguirem as instruções do assistente.

9 Utilizem os processos que desejarem, porém dêem preferência para as ferramentas nativas do sistema, como o empacotador *TAR* e os compactadores *bzip2* e *gzip*, ou que ainda utilizem estes formatos. Em situações mais drásticas, será mais fácil intervirmos com estes utilitários.

Cópia de segurança da configuração anterior

Se ao realizarmos a compilação do *kernel*, o sistema estiver rodando sem maiores problemas, poderemos optar por realizar uma cópia de segurança do arquivo de configuração gerado pela compilação anterior. Este arquivo se encontra na raiz deste diretório e se chama *.config*.

Se por algum motivo não o encontrarem, entrem no diretório do antigo código-fonte do *kernel* e executem o programa de configuração...

```
# make xconfig
# make gconfig
# make menuconfig
```

... com qualquer uma das três opções acima.

Em todas elas haverá uma opção para salvarmos a configuração padrão em um arquivo separado. Utilizem-na, definindo o nome *.config* e salvem-no em um local adequado, de preferência junto com os demais dados.

Se preferirem, podem também copiar o arquivo *config.in* disponível em */usr/src/linux/arch/i386* para */usr/src/linux*, renomeando-o para *.config*. Mas esta é uma configuração genérica; portanto, pode ser que não sirva...

Iniciando os procedimentos

Atendendo os requerimentos básicos

Dentre os principais requerimentos a serem atendidos, destacam-se a satisfação das seguintes pendências:

- O compilador *GCC* e a biblioteca padrão *GNU C*;
- As ferramentas *GNU* que acompanham a distribuição;
- O arquivador *TAR* e os compactadores *gzip* e *bzip2*;
- O servidor *X.org* e a biblioteca *Qt* para usar a opção *xconfig*;
- O pacote *ncurses* para a opção *menuconfig*;
- O código-fonte do *kernel* – lógico...

Todas as pendências necessárias encontram-se disponíveis na mídia de instalação em qualquer distribuição *GNU/Linux*, onde estas são requeridas durante a instalação para o perfeito funcionamento do sistema.

Limpando as definições prévias

Quando resolvemos realizar a recompilação de um mesmo *kernel* utilizado anteriormente, poderemos utilizar as mesmas definições de atributos realizadas na compilação anterior. Para isto basta apenas dar continuidade ao processo de compilação, iniciando os programas de configuração necessários. Porém, ao criarmos um novo perfil de

configuração, deveremos então "*limpar*" as definições de arquivos de configurações anteriores.

Para isto, faz-se necessário a utilização do comando *make mpromper*...

```
# make mpromper
```

... o qual se encarregará de realizar a exclusão das definições contidas na árvore de diretórios onde se situa as definições anteriores.

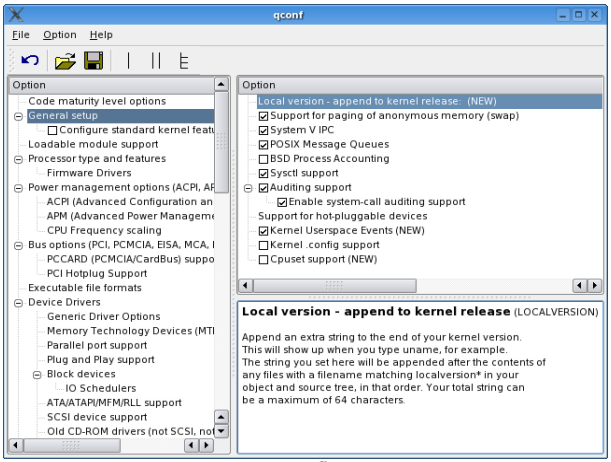
Inicializando as interfaces de interação

Para iniciar o processo de compilação, será necessário iniciar os programas de configuração para que possamos habilitar os parâmetros de configuração do *kernel*. Conforme já enfatizado, dispomos de três excelentes opções:

- *make xconfig;*
- *make menuconfig;*
- *make config.*

Todos estes três utilitários fornecem um menu de acesso e, apesar de possuírem diferentes telas de apresentação, todas as opções do *kernel* se encontram disponibilizadas. Atendem perfeitamente bem as expectativas, porém cada um possui um melhor rendimento em circunstâncias específicas, das quais iremos analisar.

O *make xconfig* fornece aos usuários, que podem disponibilizar a interface gráfica *X*, um menu gráfico simples e bem organizado, onde com apenas alguns cliques do *mouse* poderemos ter acesso a todas as categorias.

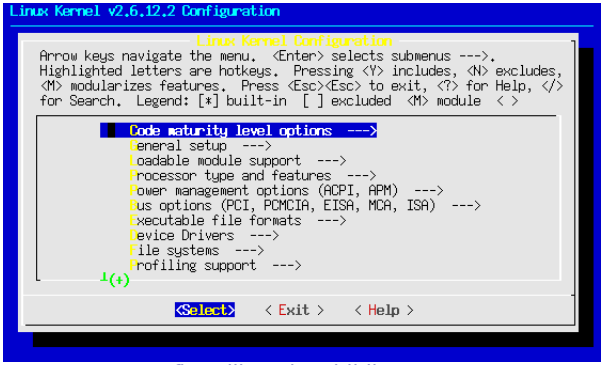


xconfig.

Para equipamentos com razoáveis recursos de *hardware* e quantidade de memória, estas são as opções mais indicada, pelas facilidades e confortos

proporcionados aos administradores.

A opção *menuconfig* também oferece um menu prático, porém em modo texto, onde a seleção dos itens é feita utilizando-se o teclado.



Menuconfig utilizando a biblioteca ncurses.

É recomendado o uso deste utilitário principalmente quando ocorre algum problema que impeça o uso da interface gráfica ou quando se deseja sobrecarregar o mínimo possível os recursos do sistema. Realizar a compilação do *kernel* com equipamentos de pouco recurso e ainda no modo texto, onde existe a necessidade de uma boa quantidade de memória realmente esta é a opção mais recomendável.

Por último, a opção *config* somente é utilizada quando existe algum problema que impeça a utilização das demais opções citadas.

```
* Linux Kernel Configuration
*
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers (EXPERIMENTAL) [Y/n/?] _
```

Esta opção é vista por muitos técnicos experientes como "*espartana*", pois além de ser a menos amigável, infelizmente apresenta as opções de forma seqüencial, onde impossibilita alterar qualquer configuração realizada previamente, ou seja, não existe a possibilidade de retornar para uma opção anterior para corrigir. Outro grande inconveniente é que, quando se utiliza os demais utilitários (*menuconfig* e *xconfig*), e de acordo com itens específicos que por ventura sejam desabilitados, suas sub-opções (exibidas de forma identadas) ficam indisponíveis (embora possamos visualizá-las); já esta opção simplesmente "*pula*" estes itens, dos quais infelizmente sequer poderemos ter noção do que deixamos para trás.

No geral, dêem preferência para a utilização do *xconfig*, lançando mão das demais opções apenas nas circunstâncias em que este não possa ser inicializado (p. ex., a impossibilidade de inicializar o modo gráfico).

Sobre a habilitação de parâmetros

Para habilitar as opções disponíveis durante a configuração do *kernel*, teremos que redefinir como estes serão utilizados pelo próprio *kernel*. Para isto, deveremos marcar as opções presentes no menu com *Y*, *M* e *N*:

- *Y (Yes):* habilita as opções selecionadas para que sejam embutidas no *kernel* principal (monolítico);
- *M (Modules):* habilita a opções selecionadas, porém somente serão carregadas sob demanda, conforme a necessidade do sistema para o seu uso (modular);
- *N (No):* desabilita as opções selecionadas.

Em muitas circunstâncias não teremos disponível a opção de habilitação ora como módulos (*M*), ora embutidos (*Y*), de diversas recursos e tecnologias de acordo com suas características e utilização.[10]

As classes de atributos

Ao inicializarmos as interfaces de interação, teremos à disposição uma série de opções estruturadas por classes, onde basta apenas navegarmos por estas estruturas e marcar e/ou desmarcar as opções que desejarmos habilitar e/ou desabilitar. Porém, a quantidade de atributos é enorme:

- *Code maturity level option*
- *General setup*
- *Loadable module support*
- *Block layer*
- *Processor type and features*
- *Power management option (ACPI, APM)*
- *Bus option (PCI, PCMCIA, EISA, MCA, ISA)*
- *Executable file formats*
- *Networking*
- *Device drivers*
- *Files system*
- *Instrumentation Support*
- *Kernel hacking*
- *Security options*

10 É somente recomendada a habilitação das opções como módulos, os dispositivos que serão utilizados ocasionalmente, pois além de possibilitar a manutenção de um *kernel* enxuto, utilizaremos menor carga de processamento e *hardware*, ganhando com isto melhor performance geral em sua utilização.

- *Cryptographic options*
- *Library routines*

Portanto, não iremos descrever as definições de cada classe, já que na atualidade, a grande maioria dos ajustes necessários para a perfeita configuração do sistema – que é o propósito desta parte – não requerem a realização do procedimento de compilação do *kernel*. Além disso, a quantidade excessiva de informações tenderá a ser desnecessárias, comprometendo assim o bom aprendizado ao invés de auxiliar.

Realizando a compilação

Após o término das operações de ajustes e configuração das opções disponíveis do *kernel*, chegamos finalmente na parte mais importante: a realização da compilação propriamente dita.

Salvando as alterações realizadas

Após definirem os novos valores para os atributos do novo *kernel* customizado, encontraremos no final da janela quatro opções para finalizar o utilitário e retornar para a linha de comando:

- *Save and Exit:* salva as alterações realizadas;
- *Quit Without Saving:* sai do menu sem salvar as alterações;
- *Load Configuration from File:* carrega uma configuração gravada;
- *Store Configuration to File:* grava as definições em um arquivo.

Caso tenham utilizado os demais menus de configuração, as opções para salvar e carregar o arquivo de configuração gerado serão diferentes, mas os mesmos conceitos aqui descritos estarão presentes.

Ao salvarem as alterações realizadas, estas serão gravadas por padrão em um arquivo chamado *.config*, situado em */usr/src/linux*, que será a base para que as ferramentas de compilação criem aquele que será o novo *kernel* do sistema e seus respectivos módulos.

Realizando a checagem das pendências

O comando *make dep* realizará a checagem das pendências necessárias e condicionará o código-fonte para a compilação do *kernel*:

```
# make dep
```

Caso já tenha sido feito o processo de compilação, deveremos utilizar...

```
# make clean
```

... para que seja feito uma limpeza dos arquivos temporários e de instalação criados durante a compilação anterior. Apesar de opcional, é recomendada a sua utilização, pois eliminará qualquer "*resíduo*" deixado

por outras compilações, além de ser uma operação bastante rápida.

Construindo a imagem compactada do kernel

Para realizarmos a compilação propriamente dita, deveremos utilizar...

```
# make bzImage
```

Este comando realizará a compilação do *kernel* baseando-se nas definições previamente realizadas durante o processo de configuração. Dependendo da capacidade de carga para o processamento, será necessário aguardarmos alguns preciosos minutos para o término da operação.

Podemos também utilizar os três últimos comandos citados neste capítulo em uma única linha para a construção da imagem compactada do *kernel*, se desejarmos. Neste caso necessitaremos digitar apenas...

```
# make dep clean bzImage
```

Uma opção interessante que poderemos utilizar, para realizar a construção da imagem do *kernel* compactada, é utilizando o comando *make bzlilo*.

```
# make bzlilo
```

Este procedimento, ao ser utilizado com o *LILO* corretamente configurado, compilará o *kernel* da mesma forma que *make bzImage*, porém logo em seguida os arquivos *System.map* e *vmlinuz* serão copiados para o diretório */boot*, e o *kernel* padrão anterior será renomeado para *vmlinuz.old*.

Isso tudo será feito de forma automática, e assim bastará apenas recondicionar o arquivo de configuração *lilo.conf* para habilitar o *kernel* recentemente compilado como padrão.

Realizando a compilação dos módulos

No que se trata da instalação, ajuste e configuração de dispositivos do sistema, certamente será requerido a compilação de módulos, já que são estes componentes do sistema que trarão suporte ao *hardware*. E para dar continuidade ao procedimento, teremos que lançar a mão dos comandos...

```
# make modules
# make modules_install
```

O comando *make modules* construirá os módulos necessários de acordo com as especificações feitas durante a configuração da compilação do *kernel*. Já o comando *make modules_install* irá instalar os (novos) módulos para o seu diretório padrão: */lib/modules/*.[11]

11 Nas edições anteriores desta literatura, utilizamos recursos técnicos que possibilitam a manunteção de vários *kernels* compilados em um mesmo sistema; mas, devido ao caráter desta parte em prover instruções gerais necessárias apenas para o processo de ajuste e configuração, resolvemos manter apenas as

A compilação dos módulos é também um pouco demorado, tempo este variado dependendo da capacidade do processador e da quantidade de memória disponível. Então aproveitem momento para esticar um pouco as pernas e tomar um bom café...

Os preparativos finais

Após a compilação do *kernel*, serão criados os seguintes arquivos:

- *bzImage:* que contém a imagem do *kernel* compilado;
- *config:* que contém a definição das novas configurações;
- *System.map:* que mapeia os módulos do *kernel*.

Estes novos arquivos estão localizados em...

```
# cd /usr/src/linux/arch/i386/boot/
```

... que por sua vez, deverão ser copiados para */boot*:

```
# mv bzImage config* System.map* /boot
```

Posteriormente, o *bzImage* terá que ser renomeado para *vmlinuz*...

```
# cd /boot
# mv bzImage vmlinux-[VERSÃO]
```

..., além de serem atualizados os atalhos...

```
# ln -s vmlinuz-[VERSÃO] vmlinuz
# ln -s config-[VERSÃO] config
# ln -s System.map-[VERSÃO] System.map
```

Mas antes, excluam os atalhos antigos.

Por fim, rodem o *LILO* para que sejam aceitas as alterações efetuadas:

```
# lilo
Added Windows
Added Linux *
# _
```

Recomendações gerais

Muitas vezes, nem tudo sai do jeito que gostaríamos que fosse. No processo de compilação do kernel, poderemos ter diversos inconvenientes ou falhas que possam comprometer o perfeito funcionamento do sistema operacional. Por isto, segue algumas recomendações gerais para evitarmos algumas ocorrências indesejadas ao processo de compilação do *kernel*, além de obtermos os melhores resultados possíveis.

- Para evitar que este erro relacionados a verificação de pendências

---

instruções básicas e necessárias. Por este motivo, recomendamos a consulta em artigos e tutoriais na *Internet* que cubram mais profundamente este assunto, caso os usuários tenham interesse em prover outras otimizações para o *kernel* além dos procedimentos básicos de ajustes e configurações.

ocorra, deveremos nos certificar de todos os requerimentos necessários para a compilação do *kernel* tenham sido atendidos.

- Ao realizarem a configuração do *kernel*, procurem habilitar apenas os dispositivos a serem suportados pela máquina em uso. Além disso apenas definam as opções como *Y* (*Yes* – embutido) para aqueles recursos essenciais e de uso constante do sistema. As demais opções deverão ficar como módulos.
- Utilizem sempre a versão estável e atualizada do *kernel* disponibilizada pelo mantenedor da distribuição. Em casos especiais optem por utilizar a versão de testes somente se esta atender a necessidades específicas, como o suporte a certas tecnologias.
- Evitem baixar diversos *patches* para atualizar *kernels* muito antigos. Além da possível demora, poderá ser um processo não muito seguro. Será mais seguro baixar um novo *kernel*.
- Sempre faça uma limpeza geral com o comando *make clean* tanto ao iniciar a operação quanto ao concluí-la. Caso contrário, teremos o risco de ocorrer diversas anomalias durante o processo, como a falha de certos procedimentos e outras anormalidades.

Outro questão de certa importância está na criação de *pendrives* de inicialização para a solução de eventuais problemas. Haverá situações em que a inicialização do sistema estará impossibilitada, seja por algum erro de configuração do *LILO* ou por qualquer outro.

Dentre as possíveis causas para diversas anomalias, a errônea definição de parâmetros durante o processo de configuração constitui a maior parte das ocorrência de erros durante o processo de compilação do *kernel*. Esse é um dos principais motivos pelo fato de recomendarmos a realização dessas definições de configuração de forma tranqüila e sossegada.

Conclusão

De acordo com a visão e a necessidade de cada usuário ou entidade, o *kernel Linux* possui aspectos e particularidades tais que poderiam render um livro inteiro se tivéssemos que detalhá-lo por completo. Aqui apenas colocamos as considerações mais básicas e importantes para obtermos uma visão geral, além das instruções necessárias para realizarmos o procedimento de compilação. Com estas informações, o usuário apenas terá um entendimento básico, porém suficiente para entender a mais importante peça do sistema operacional e realizar intervenções, se necessário. &;-D