

Boot Loader e o Kernel

Abordagem

- Conceitos do carregador de SO
 - GRUB versão 1
 - GRUB versão 2
 - Compilando o kernel
 - Atualizando o carregador para nova imagem de kernel
-
-

Carregador de S.O.

- Primeiro programa a ser carregado na memória e acionado pelo BIOS
 - Reconhecimento do sistema de arquivos e carregamento do kernel
 - Orientação GNU para classificar um carregador como multiboot
 - Principais: GRUB e Lilo
-
-

GRUB

Projeto de reescrita do carregador existente no FreeBSD a fim de atender as exigências da especificação GNU para multiboot

Necessidade de se adequar às plataformas e hardwares distintos levou aos desenvolvedores a criarem uma versão 2, que também é um código reescrito "desde o zero"

Hoje o GRUB superou o Lilo em recursos e se mantém como o mais adotado pelas distros. Ubuntu utiliza a versão 2



GRUB 1- Recursos

- Reconhecimento de binários ELF e a.out.
 - Suportar kernel que respeitam as exigências de um sistema multiboot, como os kernel 32 bit de código livre (FreeBSD, Linux, NetBSD e OpenBSD).
 - Permite o carregamento de módulos.
 - Interface de execução (CLI).
 - Possui arquivo de configuração para módulos e comandos do carregamento do kernel.
 - Suporta diversos sistemas de arquivos como BSD FS, FAT 16, FAT 32, Minix FS, ext2, Reiserfs, etc.
 - Suporta boot pela rede.
 - Independente da geometria do disco
 - Conseguir acessar qualquer dado em dispositivos que tenham sido reconhecidos pelo BIOS.
-
-

GRUB - Funcionamento

- 3 binários principais:
 - stage1 (512 bytes – MBR)
 - stage1.5 (mapa do file system)
 - stage2 (binário do GRUB)



Carregamento da imagem do kernel

É possível ao GRUB carregar uma imagem de kernel de um sistema proprietário ? Como ?



GRUB - Carregamento

- Carregamento direto - comandos :

```
#>kernel
```

```
#>modules
```

```
#>boot
```

- Carregamento por cadeia – comandos:

```
#> rootnoverify(dev,part)
```

```
#> makeactive
```

```
#> chainloader
```

```
#> boot
```

GRUB 1 - convenções

Dispositivo : (dispositivo[número],[partição],BSD
slice)

Ex: (hd1,0,a) (hd2,1,)

/boot – imagens de kernel

/boot/grub/menu.lst – configurações das imagens

menu.lst --- exemplo

```
title      Ubuntu 9.04, kernel 2.6.28-11-generic
uuid       90bb9e6a-9973-4bc7-a96f-e108e6dc74f7
kernel     /boot/vmlinuz-2.6.28-11-generic root=UUID=90bb9e6a-
9973-4bc7-a96f-e108e6dc74f7 ro quiet splash
initrd     /boot/initrd.img-2.6.28-11-generic quiet
```

```
title      Ubuntu 9.04, kernel 2.6.28-11-generic (recovery mode)
uuid       90bb9e6a-9973-4bc7-a96f-e108e6dc74f7
kernel     /boot/vmlinuz-2.6.28-11-generic root=UUID=90bb9e6a-
9973-4bc7-a96f-e108e6dc74f7 ro single
initrd     /boot/initrd.img-2.6.28-11-generic
```

menu.lst --- configurações gerais

```
default 0
fallback 1
hiddenmenu
password --md5 $1$gLhU0/$aW78kHK1QfV3P2b2znUoe
timeout      3
```

```
title      Windows 95/98/NT/2000
root       (hd0,0)
makeactive
chainloader +1
```

```
title      Linux
root       (hd0,1)
kernel     /vmlinuz root=/dev/hda2 ro
```

GRUB 2 – Novo código

/etc/default/grub – definições gerais (segundos de espera, imagem padrão, etc)

/etc/grub.d/ - entradas e definições das imagens disponíveis

/boot/grub/ - módulos e imagens do binário GRUB

Binários

- /boot/grub/boot.img

-/boot/grub/core.img

.

.

.

(módulos adicionais)

-/boot/grub/raid.mod

-/boot/grub/resiserfs.mod



GRUB.D

Arquivos recebem nomes inicializando com número para determinar a sequência de entrada no menu de carregamento do GRUB.

```
-rwxr-xr-x 1 root root 3296 2009-10-23 22:44 00_header
-rwxr-xr-x 1 root root 1154 2009-10-23 22:31 05_debian_theme
-rwxr-xr-x 1 root root 3778 2009-10-23 22:44 10_linux
-rwxr-xr-x 1 root root 772 2009-10-23 14:11 20_memtest86+
-rwxr-xr-x 1 root root 5467 2009-12-07 21:08 30_os-prober
-rwxr-xr-x 1 root root 214 2009-10-23 22:44 40_custom
```

GRUB.D

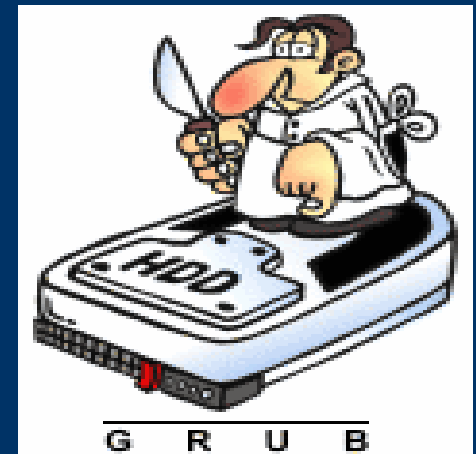
/etc/grub.d/

Arquivos que definem as mesmas sintaxes existentes em menu.lst

```
#!/bin/sh -e
echo "Adding Windows 8 to
GRUB 2 menu"
cat << EOF
menuentry "Windows 8" {
set root=(hd0,1)
chainloader (hd0,1)+1
}
EOF
```

Configuração

- Interpretação do arquivo `/boot/grub/grub.cfg` pelo binário do GRUB
- Arquivo `/boot/grub/grub.cfg` é gerado automaticamente todas as vezes que se executa o `update-grub`
- Não editar o `grub.cfg`
Entendeu?..... NAO!



Atividade

- Quantas imagens disponíveis tem configurado o seu GRUB?
- Como criar uma imagem pronta para recuperação de senha do root?



Enviando parâmetros ao kernel

- *single*
 - *root*
 - *ro*
 - *rw*
 - panic*
 - *maxcpus*
 - *debug*
 - raid*
 - selinux*
 - mem*
-
-

Carregamento do kernel

Initrd : módulos carregados em formato de imagem para auxiliar o kernel durante seu carregamento

System.map : mapa de relacionamento entre endereçamento e escopo de funções para melhor declarar as mensagens de erro durante o carregamento do kernel

Erros no carregamento do kernel

- Erros oops



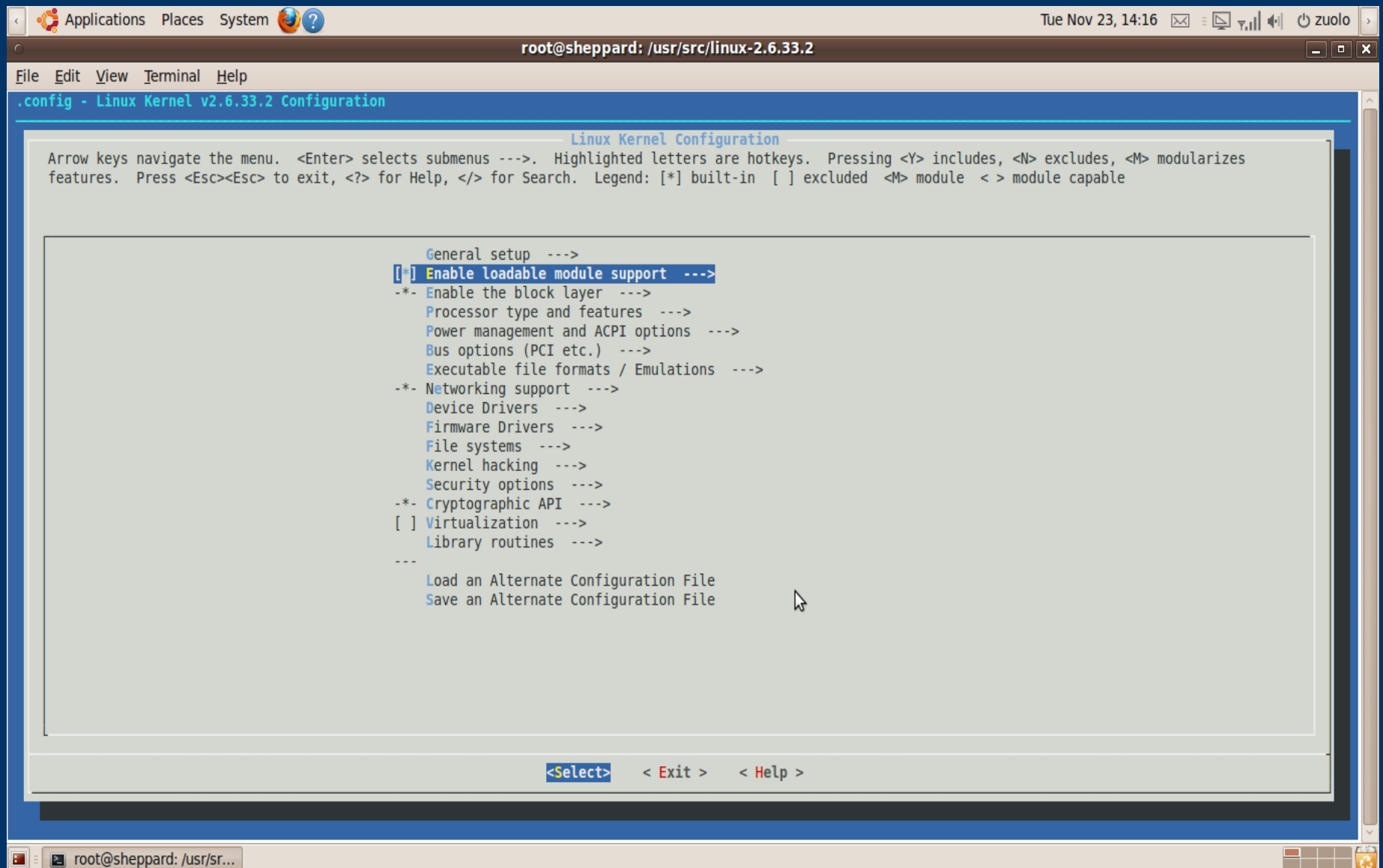
- Erros panic



Compilação do kernel

- Obter o código fonte e através do mesmo habilitar ou desabilitar as rotinas e recursos desejados
 - Compilar o código fonte, gerando o objeto.
 - Compilar os módulos
 - Instalar os módulos
 - Gerar imagem auxiliar (initramfs)
 - Gerar o System.map
 - Copiar o binário, imagem de módulos e system.map para o local definitivo.
 - Atualizar o carregado de SO
-
-

Compilação do kernel



The screenshot shows a terminal window titled "root@sheppard: /usr/src/linux-2.6.33.2". The terminal displays the "Linux Kernel Configuration" menu. The menu is titled "Linux Kernel Configuration" and includes instructions: "Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded <M> module < > module capable". The menu items are:

- General setup --->
- [*] Enable loadable module support --->**
- *- Enable the block layer --->
- Processor type and features --->
- Power management and ACPI options --->
- Bus options (PCI etc.) --->
- Executable file formats / Emulations --->
- *- Networking support --->
- Device Drivers --->
- Firmware Drivers --->
- File systems --->
- Kernel hacking --->
- Security options --->
- *- Cryptographic API --->
- [] Virtualization --->
- Library routines --->
-
- Load an Alternate Configuration File
- Save an Alternate Configuration File

At the bottom of the terminal window, there are three options: "<Select>", "< Exit >", and "< Help >".

Compilação do kernel

```
root@sheppard: /usr/src/linux-2.6.33.2
File Edit View Terminal Help
.config - Linux Kernel v2.6.33.2 Configuration
Executable file formats / Emulations
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

[*] Kernel support for ELF binaries
[ ] Write ELF core dumps with partial segments
<M> Kernel support for a.out and ECOFF binaries
<M> Kernel support for MISC binaries

<Select> < Exit > < Help >
```

Compilação do kernel passo a passo ninja



- Utilizar .config já recomendado
- #>make menuconfig
- #>make bzImage -j 16
- #>make modules
- #>make modules_install
- #>mkinitramfs -o imagem kernel_versão
- Copiar os binários e arquivos gerados para /boot
- Editar o carregador de S.O.
- Executar o carregador de S.O. E reiniciar