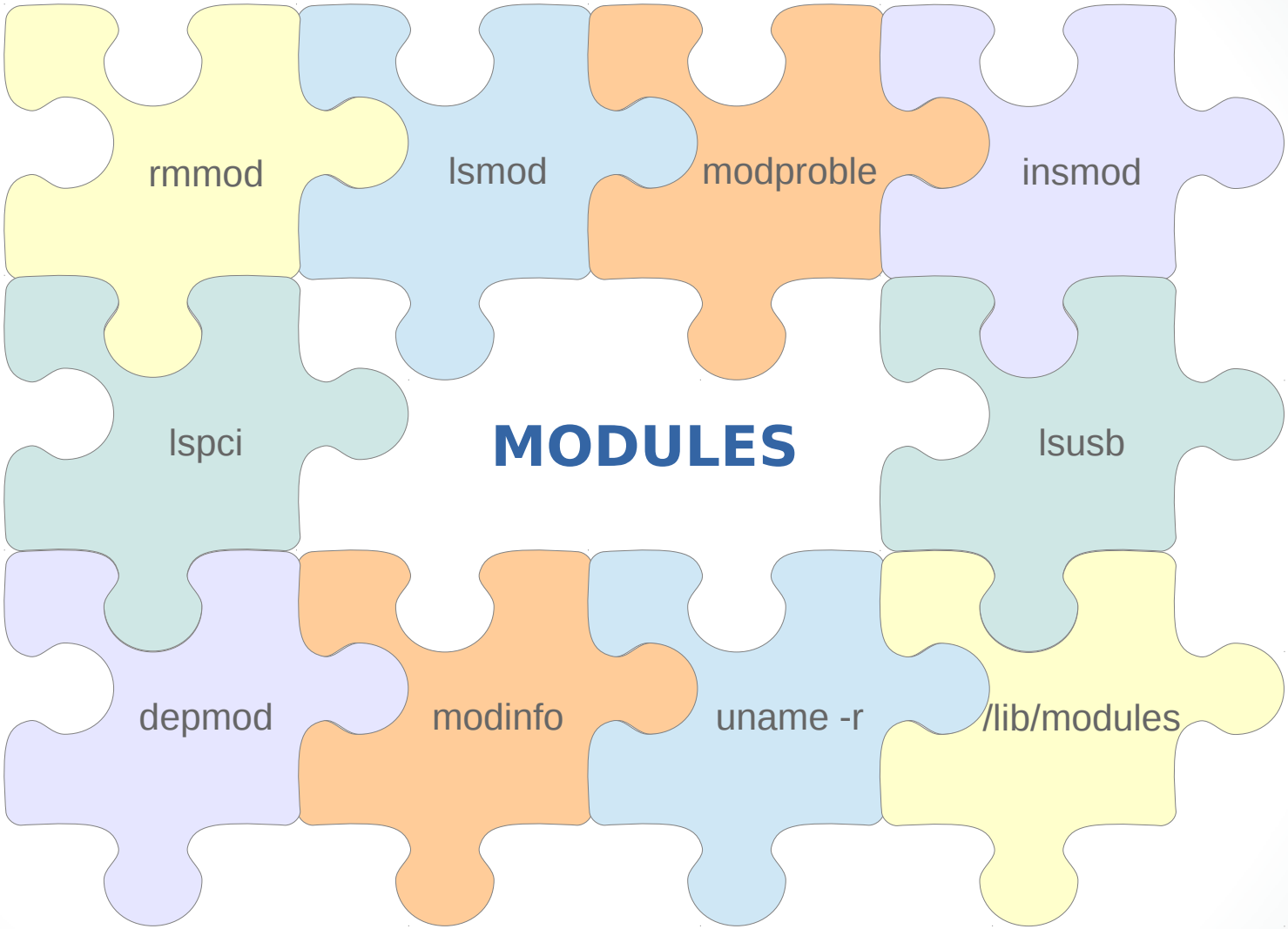
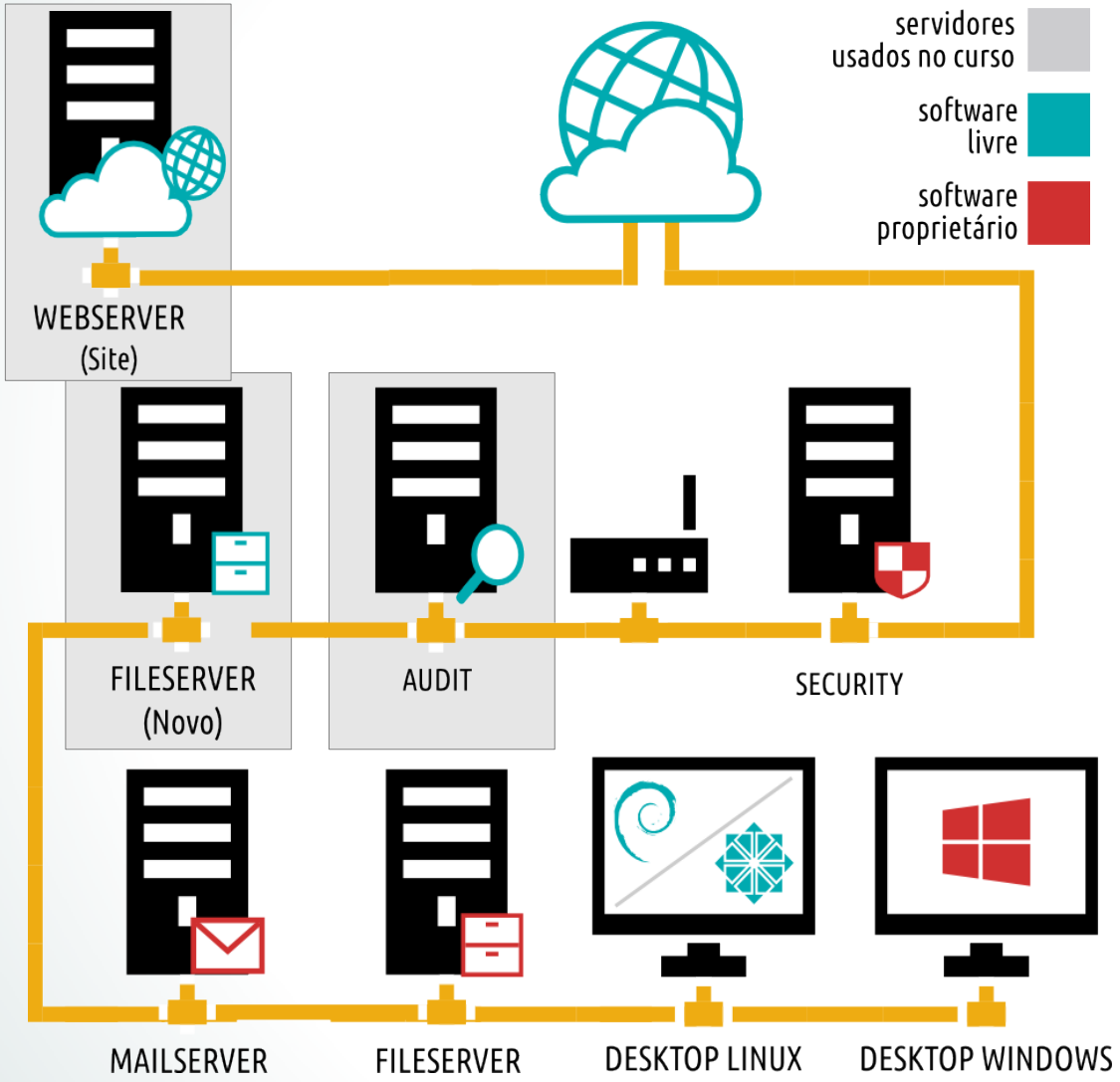

4LINUX
FREE SOFTWARE SOLUTIONS

Só na 4Linux você aprende MUITO MAIS!

Módulos, Kernel, Boot Loader



IT EXPERIENCE



Nesta Aula:

- Audit – Local

  Acesso pelo VirtualBox

  SO: Debian Linux

Módulos

- Um módulo é um trecho de código que oferece suporte a algum hardware ou funcionalidade.
- Podem ser do próprio kernel ou de terceiros.
- Além disso, quando compilamos ou recompilamos um kernel, os módulos podem ser “built-in” ou externos.

Módulos

- Módulos Buil-In:

```
1# uname -r
   3.2.0-4-486
2# cd /boot
3# grep -i quota config-3.2.0-4-486
   CONFIG_QUOTA=y
```

- Módulos Externos:

```
4# grep -i vfat config-3.2.0-4-486
   CONFIG_VFAT_FS=m
5# ls /lib/modules/
```

O arquivo **config-VERSÃO** guarda as informações do kernel que foi compilado no Sistema, informando os módulos buit-in e os módulos externos que ficam armazenados no diretório /lib/modules/VERSÃO.

Y = Buit-In

M = Módulo Externo

Módulos

- Visualizar módulos sendo utilizados:

```
1# lsmod

2# cat /proc/modules
```

- Listar módulos disponíveis:

```
3# modprobe -l

4# find /lib/modules/$(uname -r) -iname *.ko
```

O comando **lsmod** tem a função de listar todos os módulos externos ativos.

O comando **modprobe** tem a função de carregar e descarregar módulos externos na memória.

Na versão do Debian 7 a opção -l para listar todos os módulos disponíveis foi desativada.

Módulos

- Para saber informações a respeito de um módulo:

```
1# modinfo vfat

2# modinfo cdrom
```

O comando **modinfo** tem a função mostrar informações de um determinado módulo.

Para que o comando funcione o módulo precisa estar presente no diretório de módulos.

- Verifique se o módulo vfat está carregado:

```
3# lsmod | grep fat
```

- Verifique as dependências de um módulo:

```
4# modprobe --show-depends vfat
```

Módulos

- Carregue o módulo vfat:

```
1# modprobe vfat
```

- Verifique se o módulo foi carregado:

```
2# lsmod | grep vfat
```

Verifique que o modprobe além de carregar o módulo, carregou também as dependências do módulo.

- Descarregue o módulo:

```
3# modprobe -r vfat
```

```
4# lsmod | grep vfat
```

Módulos

- Existe outro comando para carregar módulos, mas é necessário passar o caminho completo do módulo e carregar primeiro todas as suas dependência:

```
1# modinfo -n vfat
```

- Tente carregar o módulo sem carregar suas dependências:

```
2# insmod $(modinfo -n vfat)
```

- Carregue as dependências do Módulo vfat:

```
3# modprobe --show-depends vfat
4# insmod $(modinfo -n fat)
5# insmod $(modinfo -n vfat)
6# lsmod | grep fat
```

Módulos

- Para descarregar o módulo individualmente temos o comando rmmod.
- Somente remove o módulo se ele não estiver sendo utilizado por outro módulo, é possível forçar com a opção “-r”, mas tome cuidado ao utilizar esta funcionalidade e parar determinada aplicação.

```
1# rmmod  vfat
```

- Repare que ele não remove as dependências:

```
2# lsmod | grep fat
3# rmmod fat
4# lsmod | grep fat
```

Módulos

- Como o “modprobe” sabe quais as dependências dos módulos?

```
1# cd /lib/modules/$(uname -r)
2# ls -l
3# less modules.dep
```

O arquivo **modules.dep** diz quais são as dependências para carregar determinado módulo.

```
4# rm -rf modules.dep*
```

- Tente iniciar o Módulo vfat com o modprobe:

```
5# modprobe vfat
libkmod: ERROR ../libkmod/libkmod.c:554 kmod_search_moddep: could not open
moddep file '/lib/modules/3.2.0-4-486/modules.dep.bin'
FATAL: Module vfat not found.
```

Módulos

- Para gerar o arquivo novamente faça:

```
1# depmod    ou   depmod -v   (Verbose)
2# ls -l
```

- Tente iniciar o módulo vfat agora que o arquivo modules.dep foi gerado:

```
3# modprobe vfat
4# lsmod | grep fat
```

Módulos

- Iniciando Módulos no Boot da máquina:

```
1# vim /etc/modules
   vfat
```

- Criando alias de módulo:

```
2# vim /etc/modprobe.d/alias.conf
   alias windows vfat
```

- Reinicie o Servidor para Validar as alterações do slide anterior:

```
3# init 6
```

Módulos

- Validando Módulos iniciado no boot:

```
1# lsmod | grep fat
```

- Validando o Alias de Módulo:

```
2# modprobe -r vfat
3# lsmod | grep  fat
4# modprobe windows
5# lsmod | grep vfat
```

Pergunta LPI

Após compilar um módulo o mesmo foi movido para /lib/modules/VERSÃO corretamente, porém quando é executado o comando modprobe <module> é retornado um erro. O que é preciso fazer para que o módulo possa ser usado?

A. modules.conf ou modprobe.conf devem ser editados.

B. O Sistema precisa ser reiniciado.

C. É necessário executar make modules_install

D. É necessário executar o depmod.

E. O Kernel precisa ser recompilado.

Pergunta LPI

Após compilar um módulo o mesmo foi movido para /lib/modules/VERSÃO corretamente, porém quando é executado o comando modprobe <module> é retornado um erro. O que é preciso fazer para que o módulo possa ser usado?

A. modules.conf ou modprobe.conf devem ser editados.

B. O Sistema precisa ser reiniciado.

C. É necessário executar make modules_install

D. É necessário executar o depmod.

E. O Kernel precisa ser recompilado.

Resposta: D

Pergunta LPI

Sua configuração de Rede se baseia na eth0 que é uma placa 3Com que reque o módulo 3c59x. Que linha deve ser adicionada ao arquivo de configuração de módulos, para garantir que eth0 sempre usará este módulo?

A. eth0=3c59x

B. alias eth0=3c59x

C. alias eth0 3c59x

D. set eth0 3c59x

E. set eth0=

Pergunta LPI

Sua configuração de Rede se baseia na eth0 que é uma placa 3Com que reque o módulo 3c59x. Que linha deve ser adicionada ao arquivo de configuração de módulos, para garantir que eth0 sempre usará este módulo?

A. eth0=3c59x

B. alias eth0=3c59x

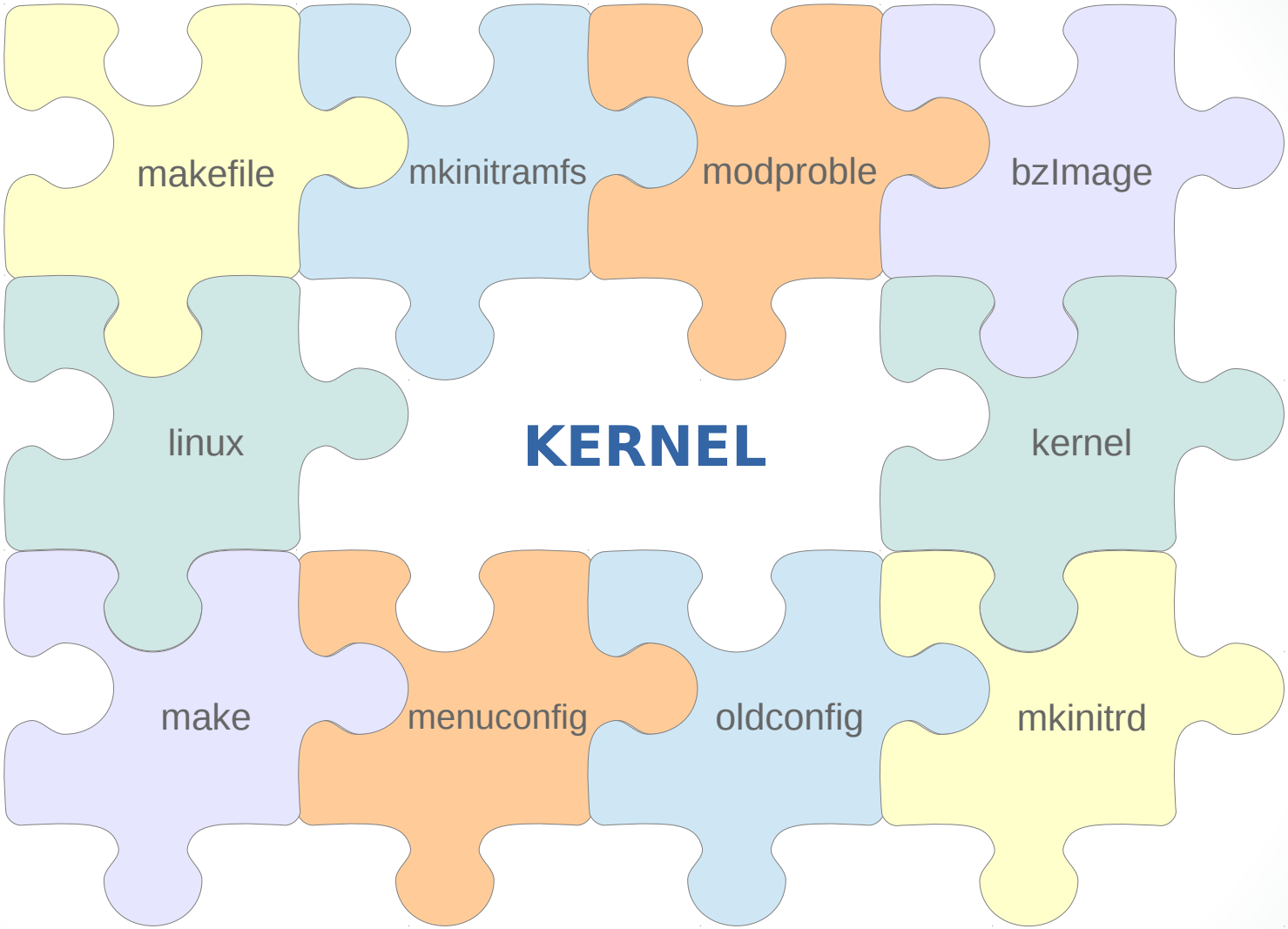
C. alias eth0 3c59x

D. set eth0 3c59x

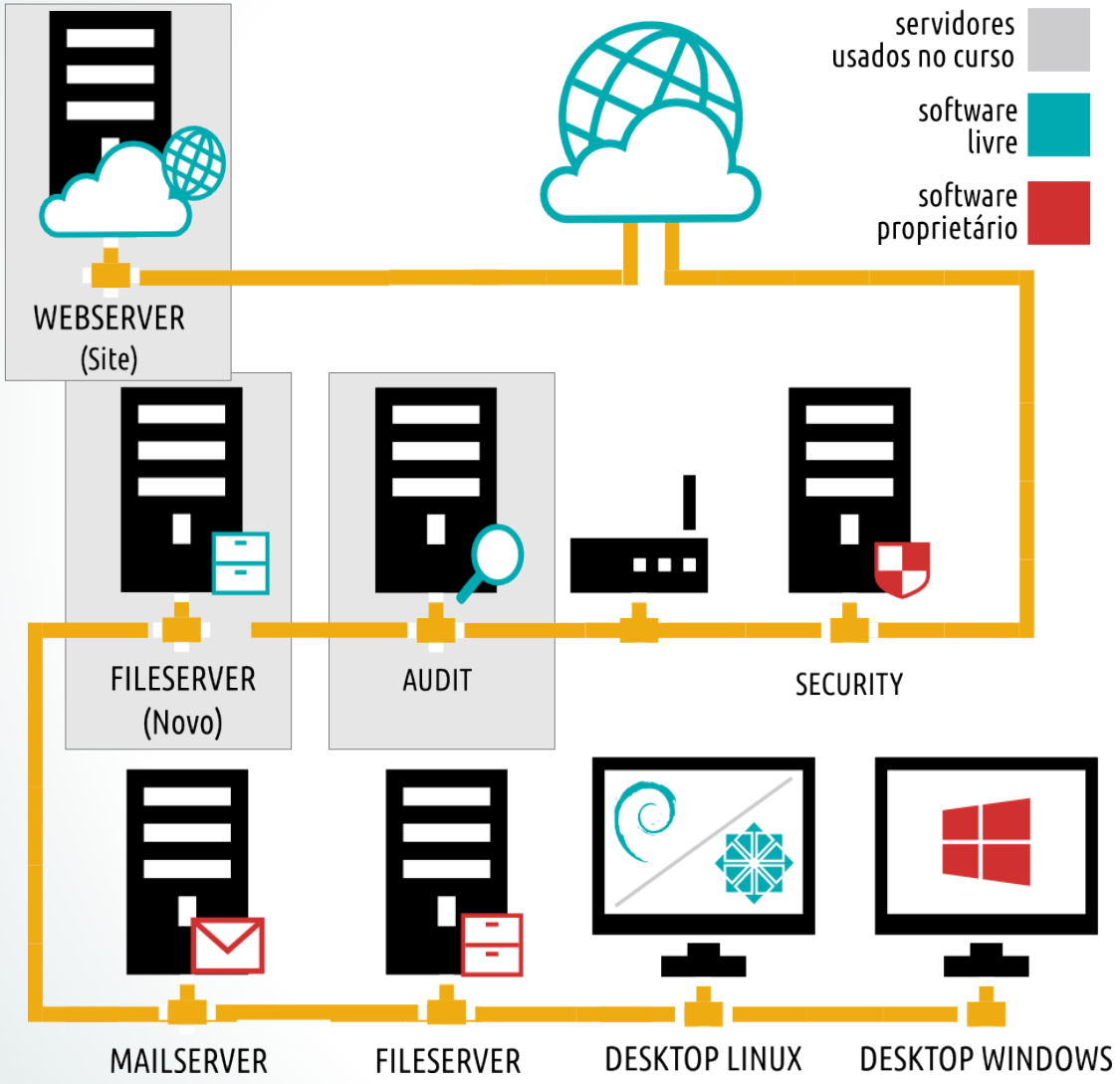
E. set eth0=

Resposta: C

Compilação de Kernel



IT EXPERIENCE



Nesta Aula:

- Audit – Local

  Acesso pelo VirtualBox

  SO: Debian Linux

Compilação de Kernel

- Antes de sair compilando o kernel é importante ter um motivo plausível para realizar essa tarefa num ambiente em produção.

Prós	Contra
- Customização (Performance e Tunning) ;	- Falta de Padronização e Controle;
- Última Versão;	- Perda de Performance se não souber fazer da melhor maneira;
- Processo independente de Distro;	- Perda de Suporte se não souber fazer da melhor maneira;
	- Perda de Suporte quando falamos de distribuições Corporativas como RedHat, Oracle Linux

Compilação de Kernel

EXTRA VERSION: Usado quando desejamos diferenciar as compilações de um kernel. As distros usam para diferenciar suas versões.

SUBLEVEL: Sua evolução, indica suporte a novos dispositivos ou correção de bugs e pequenos melhoramentos no sistema.

PATCH LEVEL: Indica mudanças importantes no funcionamento do kernel do Linux. Se **ímpar**, indica um kernel experimental, se **par**, então é um kernel considerado estável.

VERSION: Número principal da Versão. Não muda com as releases.

Compilação de Kernel

- O primeiro passo para compilarmos um “kernel” é fazer o download de seu código fonte a partir do site:

  http://www.kernel.org

- Feito isso iremos descompactá-lo no diretório apropriado:

```
1# tar xvf linux-3.2.4.bz2 -C /usr/src
```

```
2# cd /usr/src/linux-3.2.4
```

Compilação de Kernel

- Um passo extremamente importante antes de configurar o nosso “kernel” é sempre adicionar uma EXTRAVERSION afim de organizar uma eventual estrutura de módulos no “/lib”.

```
1# vim Makefile
```

EXTRAVERSION = -v1

- No Debian instale os pacotes necessários para a compilação:

```
2# aptitude install make gcc g++ autoconf libncurses5
libncurses5-dev ncurses-base ncurses-bin ncurses-term
```

Compilação de Kernel

Durante o processo de compilação, ao executar o comando make é possível utilizar o parâmetro **“-j”** para aumentar o número de trabalhos em execução diminuindo o tempo total de compilação.
Exemplo: **make -j4**

Configurar opções do “kernel”:

```
1# make menuconfig
```

Compilar o “kernel”:

```
2# make
```

Compilar os módulos do “kernel”:

```
3# make modules_install
```

Compilação de Kernel

- Depois de compilado o “kernel”, será gerado um arquivo da imagem (o bzImage) no diretório “/usr/src/linux-3.2.4/arch/XXX/boot”. (Onde XXX é a arquitetura da máquina.)
- Copie-o para o diretório “/boot”, com o nome de “vmlinuz”, este é o nome dado ao kernel:

```
1# cd arch/i386/boot/

2# file bzImage

3# cp bzImage /boot/vmlinuz-3.2.4-v1
```

Compilação de Kernel

- Se o nosso kernel tiver sido compilado com módulos que sejam extremamente necessários durante o boot, será necessário criar uma "imagem de boot".
- Para isso, precisamos instalar o seguinte pacote no Debian:

```
1# aptitude install initramfs-tools
```

- Agora devemos construir nosso arquivo "initrd" no "/boot" e atualizar o gerenciador de inicialização:

```
2# mkinitramfs -vo /boot/initrd.img-3.2.4-v1 3.2.4-v1
3# update-grub2
```

Pergunta LPI

Qual é o formato correto de numeração de versão do kernel Linux?

A. 9.04

B. 2009

C. 2.6.31

D. 2008 Server

Pergunta LPI

Qual é o formato correto de numeração de versão do kernel Linux?

A. 9.04

B. 2009

C. 2.6.31

D. 2008 Server

Resposta: C


4LINUX
FREE SOFTWARE SOLUTIONS