

## 102 Instalação do Linux e gerenciamento de pacotes

### 102.1 Design do esquema de partições do disco rígido

Os dispositivos de armazenamento (antigos discos) funcionam como um receptáculo físico que guardam os dados. Eles podem ser **particionados, separados em várias outras partes**, separando, por exemplo, dados do sistema e do usuário. As informações a respeito dessas partições ficam em uma tabela de partições, pode-se ver as partições com o comando **fdisk -l /dev/sda**. Dentro de cada partição há um sistema de arquivos, sendo a forma como as informações são armazenadas. Os **dispositivos externos** são montados no diretório **/media/USER/LABEL**. Para montar manualmente um sistema de arquivos, montá-lo em **/mnt**. É recomendado **manter alguns diretórios em partições separadas para evitar falhas**, como o carregador de inicialização, dados do usuário **/home** e dados variáveis (**servidor web, banco de dados**) **/var**.

A partição de inicialização (**/boot**) possui arquivos usados pelo gerenciador de inicialização (GRUB2 ou GRUB Legacy) para carregar o sistema operacional, e é geralmente a primeira do disco, terminando antes de 528MB, garantindo o carregamento do kernel. A partição do sistema EFI (ESP), form. em FAT, é usada para armazenar gerenciadores de inicialização e imagens do kernel. A **partição de troca (swap)** é usada para passar as páginas de memória da RAM para o disco quando ela está muito sobrecarregada. O Gerenciamento de Volumes Lógicos (LVM) é **um método de alocação de espaço**, por meio de volumes lógicos (LV) que podem ser facilmente redimensionados, diferentemente das partições.

### 102.2 Instalar um gerenciador de inicialização

Após ligar o computador, o primeiro software executado é o **gerenciador de inicialização** que depois carrega o kernel de um sistema operacional e entrega o controle do sistema a ele, então carregará drivers e hardware e depois o restante do sistema operacional. O GRUB é o gerenciador usado na maioria dos distros do Linux, podendo carregar Linux e outros sistemas, sendo o GRUB 2 uma versão melhorada do GRUB. O **bootloader GRUB fica no primeiro setor do disco** em sistemas **MBR** com BIOS e em uma partição chamada **ESP (EFI System Partition)** em sistemas com **UEFI**. A partição **/boot** contém os arquivos necessários para o processo de inicialização, como o arquivo de **configuração do kernel, o kernel do Linux, o sistema de arquivos raiz mínimo na RAM**. Instala-se o GRUB 2 com **grub-install** e a configuração pode ser feita em **/etc/default/grub**, executando depois **update-grub** para gerar um arquivo compatível do **/boot/grub/grub.cfg**. É possível adicionar novas entradas em **/etc/grub.d** no arquivo **40\_custom**, com a sintaxe:

```
menuentry "Default OS" {
```

```
set root=(hd0,1)  ou  search --set=root --fs-uuid UUID do sistema
linux /vmlinuz root=/dev/sda1 ro quiet splash
initrd /initrd.img
}
```

É possível também inicializar a partir do shell do GRUB 2, acessado por c. Para inicializar o Linux, precisa-se de um kernel (vmlinuz) e de um disco RAM inicial (initrd), então procura-se estes arquivos e usa os comandos: **set root=(hd0,msdos1), linux /vmlinuz root=/dev/sda1, initrd /initrd.img, boot**. No GRUB Legacy, algumas passos são diferentes.

### 102.3 Gerenciar bibliotecas compartilhadas

As bibliotecas de software ou objetos compartilhados são coleções de código que podem ser usadas por vários programas diferentes. **As bibliotecas estáticas** são aquelas em que o programa não tem dependências na biblioteca em tempo de execução, pois tem uma parte do código incorporada ao programa, por isso formam programas mais pesados. Nas **bibliotecas compartilhadas ou dinâmicas**, o linker não copia nenhum código da lib para o arquivo do programa. Por isso, a biblioteca precisa estar disponível durante a execução. É uma abordagem econômica para gerenciar recursos do sistema, pois ajuda a reduzir o tamanho dos arquivos de programas.

Padrão dos nomes (sonames): Nome da biblioteca com o prefixo lib, so - shared "object", número de versão da biblioteca. O nome das lib estáticas terminam em .a (libpthread.a). O vinculador dinâmico (ld.so ou ld-linux.so) busca por bibliotecas em uma série de diretórios, que são especificados em /etc/ld.so.conf.d. O comando **ldconfig** deve ser usado como root, ele é responsável por ler os arquivos de configuração, pode-se usar os parâmetros -v ou -p para ver informações das bibliotecas armazenadas na cache atual.

Pode-se usar a variável de ambiente LD\_LIBRARY\_PATH para adicionar temporariamente novos caminhos para bibliotecas compartilhadas, usando comandos como **echo**, **export**, **unset**. Entretanto, para tornar essas alterações permanentes, é necessário escrever com o comando **export** nos arquivos /etc/bash.bashrc. O comando **ldd** é usado para ver as bibliotecas compartilhadas exigidas por um certo programa em /bin.

### 102.4 Utilização do sistema de pacotes Debian

A ferramenta de pacotes do Debian (Debian Package dpkg) é o utilitário para instalar, configurar, manter e remover pacotes de software .deb. O dpkg não conseguirá instalar o pacote, se as dependências do pacote não estiverem instaladas no sistema, pois ele não consegue resolver isso sozinho.

Comandos:

- **dpkg -i** - instalar um pacote
- **dpkg -r** - remover um pacote
- **dpkg -P** - remove tudo que esteja associado ao pacote. (purge)
- **dpkg -I** [nome do arquivo]
- **dpkg --get-selections** - obtém uma lista de todos os pacotes instalados
- **dpkg -L [PACKAGENAME]** - lista todos os arquivos instalados por um pacote.
- **dpkg-query -S [caminho do arquivo]** - informa qual pacote possui este arquivo.
- **dpkg-reconfigure [nome do pacote]** - restaurar as configurações originais de um pacote, resetando-o

O Advanced Package Tool (APT) é um sistema de gerenciamento de pacotes, com ferramentas que possibilitam a instalação, atualização, remoção e gerenciamento de pacotes. Trabalha com repositórios de software, os quais possuem os pacotes disponíveis para instalação. Além disso, resolve instalando automaticamente as dependências necessárias.

Principais utilitários:

- **apt-get ou apt**: baixar, instalar, atualizar ou remover pacotes. Parâmetros:
  - **update** - verifica pacotes que podem ser atualizados
  - **upgrade** - atualiza pacotes
  - **install** - instala
  - **remove** - remove somente pacote
  - **purge ou remove -P** - remove pacote e os arquivos de configuração)
  - **install -f** - conserta pacotes quebrados
  - **clean** - limpa conteúdo dos diretórios /var/cache/apt/archives e /var/cache/apt/archives/partial/
- **apt-cache**: Buscar um pacote específico ou listar quais pacotes contém determinado arquivo. Parâmetros:
  - **search [padrão de pesquisa]** - lista todos os pacotes que contêm o padrão no nome do pacote, na descrição ou nos arquivos fornecidos.
  - **show [nome]** - mostra informações mais completas.
  - Pode-se usar somente apt search e apt show.
- **apt-file** - listar o conteúdo de um pacote ou localizar um pacote que contenha um arquivo específico.
  - **list [nome do pacote]** - listar o conteúdo de um pacote
  - **search [nome do arquivo]** - fornece qual pacote possui o arquivo procurado.

## 102.5 Uso e gerenciamento de pacotes com RPM e YUM

### O RPM Package Manager (rpm)

O RPM Package Manager (rpm) é a ferramenta essencial para gerenciar pacotes de software em sistemas baseados (ou derivados) no Red Hat. Não consegue resolver dependências sozinho.

Comandos:

- rpm -i PACKAGENAME - instalar um pacote .rpm
- rpm -U PACKAGENAME - atualizar para uma versão mais recente ou faz uma cópia se não houver versão antiga
- rpm -F PACKAGENAME - atualizar um pacote que esteja instalado
  - -v: obtém uma saída detalhada durante a instalação
  - -h: símbolos # para acompanhar progresso de instalação
- rpm -e PACKAGENAME - remover um pacote instalado
- rpm -qa PACKAGENAME - “query all” lista todos os pacotes instalados
- rpm -qi PACKAGENAME - “query info” informações sobre o pacote instalado
- rpm -ql PACKAGENAME: “query list” ver lista de arquivos que estão dentro do pacote
- rpm -qip FILENAME.rpm - ver informações de pacote não instalado
- rpm -qlp FILENAME.rpm - ver lista de arquivos do pacote
- rpm -qf caminhofile - “query file” descobre qual pacote possui o arquivo

### YellowDog Updater Modified (YUM)

Antigo Yellow Dog Updater (YUP), evoluiu para YUM para gerenciar pacotes em outros sistemas baseados em RMP, como Fedora, Red Hat ENTERprise Linux, CentOS e Oracle Linux. Semelhante ao apt em termo de funcionalidade, é capaz de buscar, instalar, atualizar e remover pacotes, além de gerir automaticamente as dependências. Pode ser usado para instalar um único pacote ou para atualizar um sistema inteiro de uma vez só.

Comandos:

- yum search PATTERN - procurar por um pacote
- yum install PACKAGENAME - instalar um pacote
- yum update {PACKAGENAME} - atualizar um pacote
- yum check-update {PACKAGENAME} - verificar se há uma atualização disponível
  - Pode emitir {} para ver geral
- yum remove PACKAGENAME - remover um pacote
- yum whatprovides FILE - mostra qual pacote fornece o arquivo
- yum info PACKAGENAME - obter informações sobre o pacote

- `yum-config-manager --add-repo URL` - adicionar um repositório ou em `/etc/yum.repos.d/` ou `/etc/yum.conf`
- `yum repolist all` - obter uma lista de todos os repositórios disponíveis
- `yum-config-manager --disable/enable ID` antes da `/` - habilitar ou desabilitar repositório
- `yum clean packages/metadata` - limpar cache, excluir pacotes/metadados baixados.

## DNF

O `dnf` é a ferramenta de gerenciamento de pacotes usada no Fedora, sendo um fork do `yum`. Como tal, muitos dos comandos e parâmetros são semelhantes

Comandos:

- `dnf search PATTERN` - busca de pacotes
- `dnf info PACKAGENAME` - informações sobre o pacote
- `dnf install PACKAGENAME` - instalar pacotes
- `dnf remove PACKAGENAME` - remover pacotes
- `dnf upgrade PACKAGENAME` - atualizar um pacote. Sem o nome, atualiza todos
- `dnf provides FILENAME` - descobre qual pacote fornece um arquivo específico
- `dnf list` - lista todos os pacotes instalados
- `dnf repoquery -l PACKAGENAME` - lista o conteúdo do pacote
- `dnf repolist` - lista todos os repositórios disponíveis, mostrando o ID
  - `--enabled/disabled` - para listar somente repositórios ativados/desativados
- `dnf config-manager --add_repo URL`
  - `dnf config-manager --set-enabled/disabled REPO_ID` - ativa/desativa repo
  - Diretório: `/etc/yum.repos.d/`

## Zypper

O `zypper` é a ferramenta de gerenciamento de pacotes usada no SUSE Linux e OpenSUSE. É semelhante ao `apt` e ao `yum`, sendo capaz de instalar, atualizar e remover pacotes de um sistema, com resolução automática de dependências.

Comandos:

- `zypper refresh` - atualizar metadados e repositórios
- `zypper se PACKAGENAME` - “search” procurar um pacote
- `zypper se -i` - lista de todos os pacotes instalados
  - `-u` - pesquisa pacote não instalado
- `zypper in PACKAGENAME` - “install” instalar um pacote

- zypper update - atualizar pacotes instalados
- zypper list-updates - lista atualizações disponíveis
- zypper rm PACKAGENAME - remover um pacote
- zypper se --provides caminhofile - ver quais pacotes contêm o arquivo específico
- zypper info PACKAGENAME - informações sobre o pacote
- zypper repos - lista de todos os repositórios atualmente registrados no sistema
- zypper modifyrepo -e/-d ALIAS - ativar ou desativar repositório
- zypper modifyrepo -F/-f - desativar/ativar atualização automática do repo
- zypper addrepo URL - adicionar novo repositório
- zypper removerepo ALIAS(nome) - remover repositório

## 102.6 O Linux como máquina virtual

A virtualização é uma tecnologia que permite que uma plataforma de software, chamada de hipervisor, execute processos que contêm um sistema inteiramente emulado (virtual). Este hipervisor é responsável por gerenciar os recursos do hardware físico que podem ser usados por máquinas virtuais individuais, permitindo que vários sistemas operacionais sejam executados em um único sistema hospedeiro. Hipervisores:

- Xen, tipo 1 (bare-metal), não depende de um sistema operacional subjacente para funcionar.
- KVM, Kernel Virtual Machine, módulo de virtualização do kernel do Linux. Tipo 1 e 2. As máquinas usam o daemon libvirt. Arquivos pode ser acessados em: /etc/libvirt/qemu
- VirtualBox, aplicativo desktop que facilita a criação e o gerenciamento de máquinas virtuais. Tipo 2 (requer um sistema operacional subjacente)

Tipos de máquinas virtuais: Totalmente virtualizado (HardwareVM, o convidado não sabe que é uma instância de máquina virtual em execução); Paravirtualizado (está ciente de ser uma instância de máquina virtual em execução); Híbrido (possui desempenho aprimorado, unido características dos outros). Uma máquina virtual consiste em um arquivo XML (define a máquina, hardware, rede, recursos) e um arquivo de imagem de disco rígido (instalação do sistema operacional).

Os arquivos da máquina virtual baseada em KVM podem ser acessados em: /etc/libvirt/qemu. Já a imagem de disco rígido em /var/lib/libvirt/images/rhel8. Tipos de imagens: COW (Copy-on-write, cópia na gravação) o arquivo de disco é criado com um limite máximo de tamanho predefinido; RAW é um tipo de arquivo que tem todo o seu espaço pré-alocado (Ex: um arquivo de imagem de disco raw de 10 GB consome 10 GB de espaço em disco real no hipervisor).

Muitas instalações do Linux utilizam um número de identificação de máquina gerado no momento da instalação, chamado de D-Bus machine ID. Localiza-se em `/var/lib/dbus/machine-id` e `/etc/machine-id`. Comandos:

- **dbus-uuidgen --ensure** - validar se existe um D-Bus machine ID para o sistema em execução
- **dbus-uuidgen --get** - visualizar D-Bus machine ID
- **sudo rm -f /etc/machine-id e sudo dbus-uuidgen --ensure=/etc/machine-id** - gerar um novo D-Bus caso duas tiverem o mesmo

Para acessar uma máquina virtual remota em uma plataforma de nuvem é por meio do software OpenSSH. O administrador usaria os comandos:

- **ssh-keygen** - para criar um par de chaves SSH públicas e privadas (`~/.ssh/` - chave pv)
- **ssh-copy-id -i <public\_key> user@cloud\_server** - copia chave pública para o servidor de nuvem remoto (`~/.ssh/authorized_keys`)

Contêiners é uma tecnologia que proporciona um ambiente isolado para implementar um aplicativo. Ele usa apenas um software suficiente para executar um aplicativo. Usam o mecanismo de grupos de controle (cgroups) no kernel do Linux, sendo uma forma de particionar os recursos do sistema, como memória, tempo de processador, disco e banda da rede, para um aplicativo individual.