



Curso 450

Linux Fundamentals in Cloud

Versão 2015_3.0

Instalação de Programas



4LINUX

2

Introdução

Os diversos programas para GNU/Linux são distribuídos em forma de pacotes específicos para cada distribuição. Neste capítulo aprenderemos um pouco sobre esses pacotes e como instalá-los e removê-los do sistema.

Pacotes são conjuntos de binários pré-compilados, bibliotecas, arquivos de controle e arquivos de configuração, que são instalados no sistema operacional.



- Explorar os gerenciadores de pacotes das principais distribuição Linux;
- Instalação de programas de Baixo Nível (DPKG e RPM);
- Instalação de programas de Alto Nível (APT e YUM);
- Preparar o ambiente para o site da Dexter



4LINUX

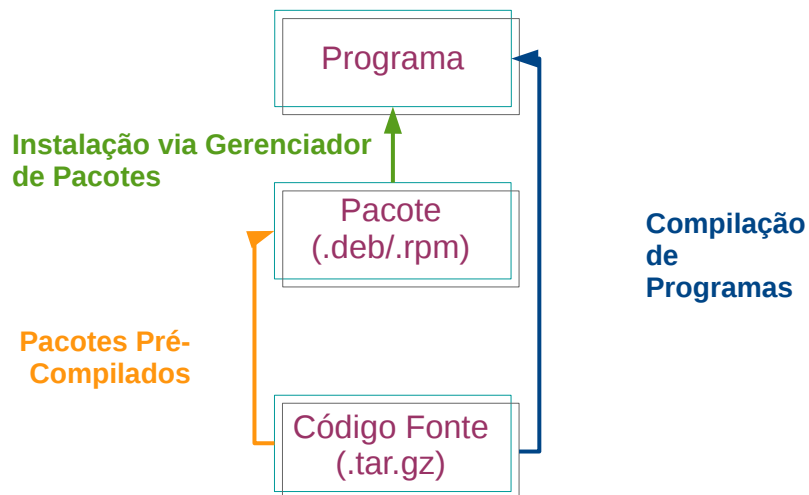
4

Anotações:

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

Instalação de Programas

Métodos de Instalação de Programas do GNU/Linux:



4LINUX

5

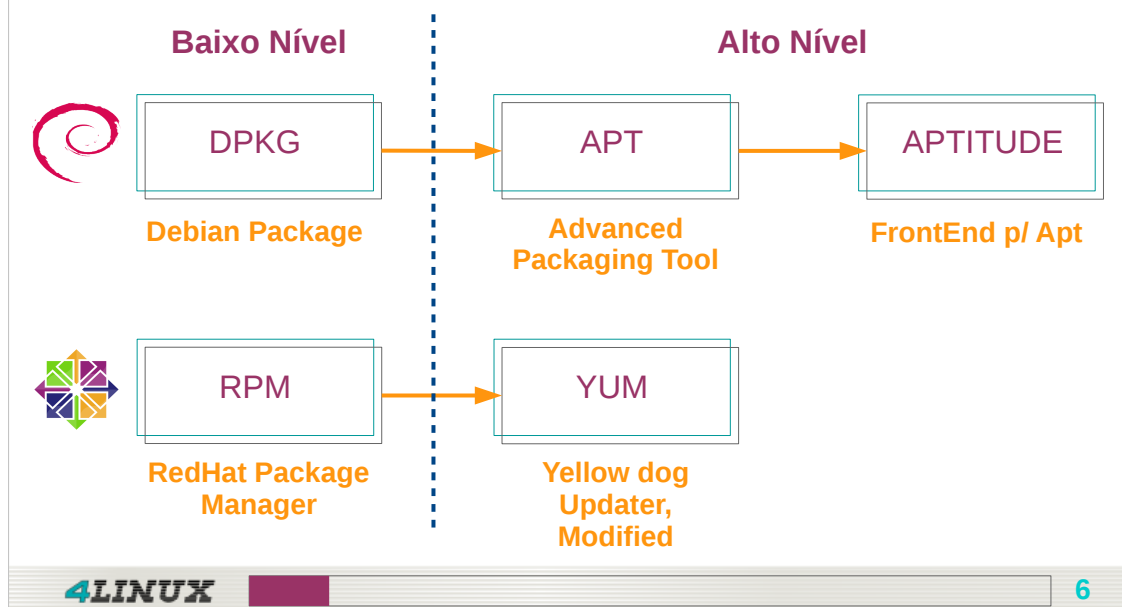
Debian Package x Red Hat Packager Manager

Os pacotes nos sistemas baseados em Debian têm uma extensão característica: **“.deb”**.

Já nas distribuições baseadas em RedHat, temos pacotes com a extensão característica: **“.rpm”**.

Instalação de Programas

Gerenciadores de Pacotes de Baixo e Alto Nível:



O que é um gerenciador de pacotes?

Um gerenciador de pacotes é um sistema para a instalação, atualização e remoção de programas em uma distribuição GNU/Linux. Parece muito simples falar em instalação de pacotes, mas temos que lembrar que é o gerenciador de pacotes quem faz toda a parte “suja” para nós.

Instalação de Programas

Gerenciadores de Pacotes de Baixo e Alto Nível:

Baixo Nível

- Instalação de forma manual;
- Sem tratamento de dependências;
- Difícil manter compatibilidade de versões;
- Tende a manter lixo no sistema.

Alto Nível

- Instalação de forma automática;
- Com tratamento de dependências;
- Fácil de manter compatibilidade de versões;
- Remove pacotes deixando o sistema limpo.

Costumamos dizer que existe duas formas de instalar programas em sistemas Linux:

- 1- Instalação de Baixo Nível - Utilizando os Gerenciadores DPKG e RPM
- 2- Instalação de Alto Nível - Utilizando os Gerenciadores Apt e Yum

A diferença básica entre as duas formas, é a facilidade que programas como Apt e Yum.



Gerenciador de Pacotes DPKG

Servidor: Máquina Linux Interna

Coletando informações dos pacotes instalados:

```
1# su - root (Senha 4linux)
2# dpkg -l
3# dpkg -l | grep htop
4# dpkg -l | grep bash
5# dpkg -s bash
6# dpkg -l | grep top
7# top
8# which top
9# dpkg -S /usr/bin/top
```

Opções do dpkg

- l** → (list) Lista todos os programas instalados;
- s** → (status) Verifica o status de um programa instalado;
- S** → (search) Busca o nome do programa que instalou determinado comando, arquivo ou biblioteca;

4LINUX

8

DPKG (Debian Package) - Gerenciador de Pacotes do Debian

O DPKG é um programa que é a base do Sistema de Gerenciamento de Pacotes para distribuições GNU/Linux baseadas em Debian. Criado por Ian Jackson em 1993, o DPKG é usado para instalar, remover e fornecer informações sobre os pacotes .deb.



Gerenciador de Pacotes DPKG

Servidor: Máquina Linux Interna

Instalação de baixo nível da ferramenta sysstat:

```
1# cd /root/pacotes/  
2# dpkg -I htop.deb  
3# dpkg -I htop.deb | grep -i depends  
4# dpkg -l | egrep "libc6|libncursesw5|libtinfo5"  
5# dpkg -i htop.deb  
6# dpkg -l | grep htop  
7# dpkg -L htop
```

Opções do dpkg:

- I → (Info) Exibe informações de um pacote a ser instalado;
- i → (install) Realiza a instalação de um pacote;
- L → (list) Lista todos os arquivos gravados no disco referente a um programa instalado;

4LINUX

9

DPKG (Debian Package) - Gerenciador de Pacotes do Debian

O DPKG é uma ferramenta em linguagem de baixo nível. Front ends de alto nível são exigidos para buscar pacotes em lugares remotos ou ajudar no solucionamento de conflitos nas dependências dos pacotes. Para esta finalidade, o Debian fornece aptitude e o apt-get.

Se necessário consulte as opções do dpkg:

```
# dpkg -- help
```



Gerenciador de Pacotes DPKG

Servidor: Máquina Linux Interna

Removendo programas desnecessários nos servidores:

```
1# dpkg -l | less
2# dpkg -l | grep figlet
3# dpkg -r figlet
4# dpkg -l | grep figlet
5# dpkg -r fetchmail
6# dpkg -l | grep fetchmail
7# dpkg --purge fetchmail
8# dpkg -l | grep fetchmail
```



Opções do dpkg

-r → (remove) Remove um programa sem suas dependências. Remoção Parcial;

--purge → Remove um programa e seus arquivos de configuração. Remoção Completa.

4LINUX

10

Expurgando pacotes

É muito comum que, ao removermos um determinado pacote dependências deste pacote não sejam desinstaladas, neste caso o parâmetro **--purge** pode ser utilizado para remover estes pacotes limpando completamente o sistema da aplicação anteriormente instalada.



Gerenciador de Pacotes RPM

Servidor: Webserver Interno

Coletando informações dos pacotes instalados:

```
1# rpm -qa
2# rpm -qa | grep htop
3# rpm -qa | grep bash
4# rpm -qi bash
5# rpm -qa | grep top
6# top
7# which top
8# rpm -qf /bin/top
```



Opções do rpm

-qa → Lista todos os programas instalados;

-qi → Exibe informações sobre o programa instalado;

-qf → Busca o nome do programa responsável por instalar um determinado comando, arquivo ou biblioteca;

4LINUX

11

RPM (RedHat Package Manager) - Gerenciador de Pacotes Red Hat

O RPM é gerenciador de pacotes para sistemas GNU/Linux baseados em RedHat. Ele instala, atualiza, desinstala e verifica softwares. Originalmente desenvolvido pela RedHat Linux, é agora usado por muitas distribuições como Novell e Suse que possuem sua própria versão de RPM.



Gerenciador de Pacotes RPM

Servidor: Webserver Interno

Instalação de baixo nível da ferramenta sysstat:

```
1# cd /root/pacotes
2# rpm -qa | grep htop
3# rpm -qpi htop.rpm
4# rpm -qpR htop.rpm
5# rpm -ivh htop.rpm
6# rpm -qa | grep htop
7# rpm -ql htop
8# htop
```

Opções do rpm

- qa** → Lista todos os pacotes instalados;
- qpi** → Exibe informações de um pacote a ser instalado;
- qpR** → Lista dependências do pacote;
- ivh** → Instala um pacote mostrando detalhes (v) com uma barra de progresso (h);
- ql** → Lista todos os arquivos do programa.

4LINUX

12

Base de dados RPM

Atrás do gerenciador de pacotes está o banco de dados RPM. Ele consiste em uma lista duplamente ligada que contém todas as informações de todos os RPM instalados.

O banco de dados lista todos os arquivos que são criados ou modificados quando um usuário instala um programa e facilita a remoção destes mesmos arquivos. Se o banco de dados fica corrompido (o que acontece facilmente se o cliente de RPM é fechado subitamente), as ligações duplas garantem que eles possam ser reconstruídos sem nenhum problema.

Em computadores com o sistema operacional RedHat instalado, este banco de dados encontra-se em `/var/lib/rpm`.



Gerenciador de Pacotes RPM

Servidor: Webserver Interno

Removendo programas desnecessários nos servidores:

```
1# rpm -qa | less
2# rpm -qa | grep figlet
3# rpm -e figlet
4# rpm -qa | grep figlet
```



Opções do rpm

-e → Remove um programa sem suas dependências;

4LINUX

13

Verificação da integridade de Pacotes:

Uma vantagem que o RPM possui sobre DPKG é que possui ferramentas de verificação criptográfica como o GPG e o md5, além de verificação de integridade dos arquivos já instalados. Dessa forma, periodicamente você pode verificar se ocorreu alguma alteração no seu sistema sem você saber ou se sua máquina foi invadida, pode-se tentar identificar o que foi mexido nela.

Existe uma documentação que pode ser usada para qualquer distro baseada em RPM que pode ser encontrada em: <http://www.rpm.org/RPM-HOWTO/>.



Gerenciador de Pacotes RPM

Servidor: Webserver Interno

- Desempacotar pacotes com **rpm2cpio** - O comando rpm2cpio converte o pacote RPM para um arquivo CPIO;
- Para visualizar o conteúdo do pacote .rpm com **rpm2cpio**:

```
1# rpm2cpio htop.rpm | cpio -t
```

- Para converter o pacote .rpm para cpio execute o seguinte comando:

```
1# rpm2cpio htop.rpm > htop.cpio
```

```
2# file htop.cpio
```

```
3# cat htop.cpio | cpio -t
```

4LINUX

14

Anotações:

[illegible]

Resumão LPI (parte 1/2)



| Função | DPKG | RPM |
|-------------------------------------|-------------------------------------------------|----------------------------------|
| Mostrar informações sobre um pacote | <code>dpkg -I pacote.deb</code> | <code>rpm -qpi pacote.rpm</code> |
| Listar pacotes instalados | <code>dpkg -l</code> | <code>rpm -qa</code> |
| Listar as dependências de um pacote | <code>dpkg -s pacote.deb grep ^Depends</code> | <code>rpm -qpR pacote.rpm</code> |
| Instalar um pacote | <code>dpkg -i pacote.deb</code> | <code>rpm -ivh pacote.rpm</code> |
| Remover um pacote | <code>dpkg -r pacote.deb</code> | <code>rpm -e pacote.rpm</code> |
| Atualizar um pacote | <code>dpkg -i pacote.deb</code> | <code>rpm -Uvh pacote.rpm</code> |

DICA:

Fique esperto! Comandos dos gerenciadores RPM e DPKG são amplamente abordados nas provas LPI.

| Função | DPKG | RPM |
|---------------------------------------------------|------------------------------------------------|-----------------------------------------------|
| Forçar instalação de um pacote | <code>dpkg -i --force-all pacote.deb</code> | <code>rpm -i --force pacote.rpm</code> |
| Listar arquivos de um pacote | <code>dpkg -L pacote</code> | <code>rpm -ql pacote</code> |
| Listar quais pacotes dependem daquele pacote | <code>dpkg --purge --dry-run pacote.deb</code> | <code>rpm -q --whatrequires pacote.rpm</code> |
| Listar pacote que instalou o comando especificado | <code>dpkg -S path_do_binário</code> | <code>rpm -qf path_do_binário</code> |
| Auditoria de Pacotes | <code>dpkg -C</code> | <code>rpm -Va</code> |

[illegible]

- a. `dpkg -Dh`
- b. `apt-get -y`
- c. `apt-get -u`
- d. `dpkg -C`
- e. `dpkg -l`

[illegible]

Pergunta LPI



Qual comando do gerenciador de pacotes do Debian listará todos os pacotes parcialmente instalados e sugerir o que precisa ser feito para corrigir?

- a. `dpkg -Dh`
- b. `apt-get -y`
- c. `apt-get -u`
- d. **`dpkg -C`**
- e. `dpkg -l`

Resposta: d

4LINUX

18

Alternativa D: RESPOSTA CORRETA!

A opção -C ou --audit do comando dpkg procura pacotes que foram instalados apenas parcialmente em seu sistema. Caso o dpkg encontre pacotes nesta situação o dpkg irá sugerir o que fazer com eles para devolvê-los a seu correto funcionamento.

- `rpm -qf /etc/exports`
- `rpm -qp /etc/exports`
- `rpm -qi /etc/exports`
- `rpm -Kl /etc/exports`
- `rpm -F /etc/exports`

[illegible]

Pergunta LPI



Qual seria o parâmetro do comando do rpm a mostrar o nome do pacote responsável por ter instalado o arquivo `/etc/exports`?

- a. `rpm -qf /etc/exports`
- b. `rpm -qp /etc/exports`
- c. `rpm -qi /etc/exports`
- d. `rpm -Kl /etc/exports`
- e. `rpm -F /etc/exports`

Resposta: a

4LINUX

20

Alternativa A: RESPOSTA CORRETA!

Toda vez que for necessário listar a origem de um pacote utilize as opções **qf** do comando rpm.

Gerenciador de Pacotes de Alto Nível



Gerenciadores de Alto Nível

Um pacote nem sempre depende apenas dele mesmo, ou seja, quando instalamos um programa, ele pode depender de bibliotecas de áudio, vídeo, imagens, funções e vários outros programas que precisam estar instalados antes do pacote. É esse elo entre programas que chamamos de dependências.

Ao utilizar um gerenciador de Baixo Nível essas dependências devem ser resolvidas manualmente enquanto gerenciadores de alto nível fazem o processo de resolução através de consultas a seus repositórios.

Gerenciador de Pacotes de Alto Nível

- **Unstable** → Últimas versões de programas, porém não homologado;
- **Testing** → Versões recentes, em homologação para a próxima versão;
- **Stable** → Não necessariamente as últimas versões de programas, porém testado e homologado pela Comunidade Debian, ou seja, programas estáveis.



Anotações:

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Desde 1996, as versões do Debian são batizadas com o nome dos personagens do filme Toy Story. Essa ideia veio de Bruce Perens, que assumiu a liderança do Projeto Debian e trabalhava na Pixar, onde o filme foi produzido na época.



24

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.



Gerenciador de Pacotes APT

Servidor: Webserver Externo

Configuração do repositório Debian:

```
1# vim /etc/apt/sources.list
deb http://ftp.debian.org/debian wheezy main contrib non-free
```

- **http://...** É o método de acesso ao repositório, também poderia ser `cdrom://` para CDs de pacotes, `file://` para repositórios locais;
- **wheezy** É a distribuição. Neste campo opções possíveis são `stable`, `testing` e `unstable` ou usando o codinome do Toy Story;
- **main contrib e non-free** são as seções do repositório.

4LINUX

25

O arquivo `/etc/apt/sources.list`

O arquivo “`/etc/apt/sources.list`” contém os locais onde o “APT” encontrará os pacotes, a versão da distribuição que será verificada (`stable`, `testing`, `unstable`) e a seção que será copiada (`main`, `non-free`, `contrib`, `non-US`), ou seja o conteúdo deste arquivo compõe a lista de repositórios disponíveis.

Para buscar uma lista completa de pacotes disponíveis para Debian acesse:

<http://packages.debian.org>



Gerenciador de Pacotes APT

Servidor: Webserver Externo

Explorando o APT:

```
1# apt-get update
2# apt-get upgrade
3# apt-cache search "log color"
4# apt-cache show ccze
5# apt-get install ccze
6# tailf /var/log/messages | ccze
7# apt-get remove ccze
8# dpkg -l | grep ccze
9# apt-get remove --purge ccze
10# dpkg -l | grep ccze
```

Opções do APT

update → Atualiza a base de repositórios;
upgrade → Atualiza todos os pacotes com versões novas;
apt-cache search → Busca pacotes relacionados a palavra chave;
apt-cache show → Mostra informações detalhadas de um pacote;
remove → Remove um pacote e suas dependências;
remove --purge → Remove completamente o pacote incluindo arquivos de configuração.

4LINUX

26

APT (Advanced Package Tool) - Gerenciador de Pacotes Avançado do Debian

Os comandos apt são utilizados para diversas funções, entre elas sincronizar listas de pacotes disponíveis, Conectar-se a servidores remotos para consultar, instalar ou atualizar aplicações.

Toda vez que uma informação de repositório é adicionada ao repositório o comando apt-get update deve ser executado, permitindo a sincronização dessas informações.



Gerenciador de Pacotes APT

Servidor: Webserver Externo

Funcionalidades avançadas:

```
1# ls /var/cache/apt/archives
2# apt-get clean
3# ls /var/cache/apt/archives
4# apt-get autoremove
5# apt-get -f install
6# apt-get -f remove
7# apt-get -d install httpd
8# httpd
9# ls /var/cache/apt/archives
```

Opções do APT

clean → Limpa o diretório de pacotes que são baixados para realizar uma instalação;

dist-upgrade → Caso disponível, atualiza a distribuição do Debian;

autoremove → Localiza e remove todos os pacotes "orfãos";

-f install → Aciona o sistema de resolução de problemas do APT;

-f remove → Similar ao "-f install", porém ele dá preferência para remover os pacotes com problemas ao invés de tentar instalar;

-d → Não instala, apenas faz download.

4LINUX

27

Removendo pacotes que não serão mais usados

Quando você instala um pacote o apt busca das fontes listadas em "/etc/apt/sources.list" os arquivos necessários são baixados e guardados em um repositório local "/var/cache/apt/archives/", para só então ser feita a instalação, em algum tempo o repositório local pode crescer e ocupar muito espaço em disco.

Felizmente o apt fornece uma ótima ferramenta para lidar com seu repositório local, o "apt-get clean" remove tudo exceto os arquivos de lock dos diretórios "/var/cache/apt/archives/" e "/var/cache/apt/archives/partial/".

Caso a remoção seja feita e você deseje instalar um novo pacote, o apt irá buscá-lo novamente quando o comando de instalação for executado.



Gerenciador de Pacotes YUM

- O **YUM** gerencia pacotes .rpm e está presente nas distribuições Red Hat, Fedora, CentOS;
- Outras distribuições que usam RPM optaram por criar seu próprio gerenciador avançado ao invés de utilizar o YUM:
 - **YUM** → Red Hat, Fedora, CentOS;
 - **URPMI** → Mandriva;
 - **Zypper** → Suse.



4LINUX

28

Gerenciadores de Alto-nível diversos

Infelizmente não existe um padrão único em relação ao uso de gerenciadores de Alto-Nível, Cada distribuição utiliza seu modelo de gerenciador e este modelo (felizmente) é passado a novas distribuições de mesma família.

Gerenciador de Pacotes YUM



Servidor: Webserver Interno

Configuração dos repositórios CentOS:

- As configurações realizadas no diretório `/etc/yum.repos.d` também podem ser adicionadas no arquivo de configuração `/etc/yum.conf`

```
1# vim /etc/yum.conf
[main]
cachedir=/var/cache/yum/$basearch/
$releasever
keepcache=1
debuglevel=2
logfile=/var/log/yum.log
obsoletes=1
gpgcheck=1
```

Note que você pode utilizar as configurações de repositório em apenas um dos dois locais: `/etc/yum.conf` ou `/etc/yum.repos.d`.

4LINUX

30

Composição de um repositório Red Hat

O modelo acima é um exemplo de um arquivo `.repo` ou seja, um repositório Red Hat, cada repositório é composto por um arquivo com as informações acima além de outras opções possíveis.

UM dos principais repositórios utilizados em ambiente Red Hat / CentOS é o repositório epel disponível no site do projeto:

<https://fedoraproject.org/wiki/EPEL>

Certificação Red Hat



Servidor: Webserver Interno

- Na Certificação RHCSA e RHCE você precisará adicionar repositórios do YUM manualmente;

- Adicione um novo Repositório:

```
1# yum install epel-release  
2# yum repolist
```



Também é possível adicionar o repositório a partir do link:

http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm

4LINUX

31

Adicionando repositórios

Para adicionar um repositório ao ambiente basta criar um arquivo com informações similares as do modelo, este arquivo deverá ser destinado ao diretório: `/etc/yum.repos.d/` também é possível baixar o arquivo de repositório diretamente da internet e movê-los a localização citada.



Gerenciador de Pacotes YUM

Servidor: Webserver Interno

Explorando o APT:

```
1# yum repolist
2# yum search "linux logo"
3# yum info linux_logo
4# yum install linux_logo
5# linux_logo -L23
6# yum remove linux_logo
7# rpm -qa | grep linux_logo
```

Repositório YUM

repolist → Validar quais repositórios estão sendo usados;

update → Atualiza todos os pacotes com versões mais novas;

search → Busca pacotes relacionados a palavra chave;

info → Mostra informações detalhadas de um pacote;

remove → Remove um pacote e suas dependências.

4LINUX

32

YUM (Yellow dog Updater, Modified) - Gerenciador de Pacotes para Distribuições baseadas em RPM (RedHat)

O yum é o gerenciador de pacotes usado por padrão no CentOS, no Fedora e no RedHat Enterprise. O yum foi originalmente desenvolvido pela equipe do Yellow Dog (uma distribuição baseada no Red Hat, destinada a computadores com chip PowerPC) e foi sistematicamente aperfeiçoado pela equipe da Red Hat, até finalmente assumir o posto atual.

O yum trabalha de forma bem similar ao apt-get e aptitude, baixando os pacotes a partir dos repositórios especificados nos arquivos de configuração, junto com as dependências necessárias.



Gerenciador de Pacotes YUM

Servidor: Webserver Interno

Funcionalidades avançadas:

```
1# ls -R /var/cache/yum
2# yum clean all
3# yum upgrade
4# yum install ccze -y --downloadonly
5# cd /var/cache/yum/x86_64/7/
6# ls epel/packages/ccze*
```

Repositório YUM

clean all → Limpa o diretório de pacotes que são baixados para realizar uma instalação;

upgrade → Caso disponível atualiza a distribuição do CentOS;

--downloadonly → Plugin do Yum para realizar o download de pacotes sem realizar a instalação do mesmo.

4LINUX

33

Características do Yum

Diferente do gerenciador de pacotes do Debian onde toda vez que uma modificação é feita na lista de repositórios é necessário que esta lista seja atualizada, no yum esta atualização é feita automaticamente cada vez que uma instalação é solicitada, checando os repositórios, baixando os headers do pacotes e calculando as dependências antes de confirmar a instalação.

Servidor: Webserver Externo

- 

4 LINUX

Anotações:

Servidor: Webserver Externo

4 LINUX

Anotações:

[illegible]



Laboratório Dexter

Servidor: Webserver Externo

Validando a Instalação do PHP5:

- Para validar a instalação do pacote PHP5, vamos criar um arquivo em php que nos fornece informações referente a versão do php.

```
1# cd /var/www/  
2# rm -rf index.html  
3# vim index.php  
    <?php  
        phpinfo();
```

Salve e Saia do arquivo.

[illegible]

Anotações:

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Visualizar a página de php:

- Na máquina **Linux Interna** abra um navegador e acesse o ip da Webserver Externo.

37

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

| Função | APT | APTITUDE | YUM |
|----------------------------------|-------------------------------------|-------------------------------------|--------------------------------|
| Adicionar Repositórios | <code>/etc/apt/sources.list</code> | <code>/etc/apt/sources.list</code> | <code>/etc/yum.repos.d</code> |
| Buscar Pacotes por Palavra Chave | <code>apt-cache seach "word"</code> | <code>aptitude search "word"</code> | <code>yum search "word"</code> |
| Informações do Programa | <code>apt-cache show</code> | <code>aptitude show</code> | <code>yum info</code> |
| Instalar um Programa | <code>apt-get install</code> | <code>aptitude install</code> | <code>yum install</code> |
| Remover um Programa | <code>apt-get remove</code> | <code>aptitude remove</code> | <code>yum remove</code> |

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

| Função | APT | APTITUDE | YUM |
|------------------------------|----------------------|-----------------------|----------------------------|
| Atualizar um Programa | apt-get install | aptitude install | yum install |
| Atualizar todos os Programas | apt-get upgrade | aptitude upgrade | yum update |
| Atualizar a Distro | apt-get dist-upgrade | aptitude safe-upgrade | yum upgrade |
| Baixar Programa sem Instalar | apt-get -d install | aptitude -d install | yum install --downloadonly |
| Limpar cache de Instalação | apt-get clean | aptitude clean | yum clean all |

[illegible]

- a. `apt-get dist-upgrade`
- b. `apt-get mirror-select`
- c. `apt-get upgrade`
- d. `apt-get update`
- e. `apt-get build-dep`

[illegible]

Pergunta LPI



Qual opção do comando apt-get você deve executar para atualizar a lista de pacotes?

- a. apt-get dist-upgrade
- b. apt-get mirror-select
- c. apt-get upgrade
- d. **apt-get update**
- e. apt-get build-dep

Resposta: d

4LINUX

41

Alternativa D: RESPOSTA CORRETA!

Utilize o comando **apt-get update** para atualizar a lista de pacotes disponível a partir dos repositórios da máquina.

- a. /etc/apt/sources.list
- b. /etc/dpkg/dselect.cfg
- c. /etc/apt/source.list
- d. /etc/apt/apt.conf
- e. /etc/dpkg/dpkg.cfg

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Pergunta LPI



Qual arquivo deve ser editado para selecionar repositórios do Debian para instalação de pacotes?

- a. `/etc/apt/sources.list`
- b. `/etc/dpkg/dselect.cfg`
- c. `/etc/apt/source.list`
- d. `/etc/apt/apt.conf`
- e. `/etc/dpkg/dpkg.cfg`

Resposta: a

Alternativa A: RESPOSTA CORRETA!

Qual opção, quando usada com o comando yum, irá atualizar todo o sistema? (Especifique apenas o nome da opção)

[illegible]

Pergunta LPI



Em qual diretório você adiciona novos repositórios para o yum?

Resposta: `/etc/yum.repos.d/`

Qual opção, quando usada com o comando yum, irá atualizar todo o sistema? (Especifique apenas o nome da opção)

Resposta: `yum update`

4LINUX

45

RESPOSTA CORRETA: `/etc/yum.repos.d`

RESPOSTA CORRETA: `update`

DICA:

Não confunda! EM ambientes Debian ou derivados a opção `update` atualiza a lista de pacotes existentes enquanto a opção `upgrade` atualiza o sistema, Já em distribuições RedHat ou derivadas os repositórios são atualizados ao executar funções do yum, sendo que a opção `update` atualiza todo o sistema.

Conversão de Pacotes



- Para que seja possível instalar um **pacote.deb** em um **RedHat** ou um **pacote.rpm** em um **Debian**, temos um utilitário de conversão chamado **Alien**;



Só use o alien em caso de necessidade - tente primeiro procurar o programa na versão corrente da sua distribuição. Embora ele tenha um papel importante, seu uso é recomendado apenas em último caso.

Alien

O programa Alien também pode ser utilizado para efetuar outros tipos de conversão, dando suporte aos formatos Stampede (.SLP), Solaris (.Pkg) e Slackware (.Tgz,.Txz,.Tbz,.TLZ).

- Ligue a máquina **Linux Interna** para que possa converter um pacote .deb e em seguida instalá-lo no **WebServer Interno**;
- Faça download do Pacote sl que só tem disponível para Distribuições Debian e em seguida converta para RPM:

```
1# su - root (Senha: 4linux)
2# apt-get install alien
3# apt-get -d install sl
4# cd /var/cache/apt/archives/
5# mv sl*.deb /root ; cd
6# alien -r sl<TAB>
7# ls
8# scp sl*.rpm root@192.168.200.20: (Senha: 4linux)
```

47

[illegible]

Servidor: Webserver Interno

```
3# rpm -ivh sl<TAB>
```

```
6# /usr/games/sl
```

O exemplo com o SL tem objetivo meramente didático. **Não** é recomendado o uso do alien em Servidores, justamente porque não se tem garantias de estabilidade.

Anotações:

[illegible]

Instalação de Programas – Parte II



49

Fundamentação

Um dos pontos centrais do mundo GNU/Linux está baseado nas quatro liberdades básicas propostas pela FSF - Free Software Foundation, sendo elas:

- Liberdade de rodar o programa para qualquer propósito;
- Liberdade de acesso ao código fonte, estudar como ele funciona e adaptá-lo às suas necessidades;
- Liberdade de redistribuir cópias do software;
- Liberdade de melhorar o programa e distribuir essas melhorias em benefício da Comunidade.

Para que essas quatro liberdades básicas sejam alcançadas é necessário que tenhamos acesso ao código fonte dos programas.

Tirando a parte ideológica, há diversas situações que exigem que recompilemos um determinado software a partir do código fonte, sendo algumas delas, quando necessitamos alterá-lo para que ele satisfaça alguma necessidade pessoal, corrigir um erro ou melhorar a segurança, o software não está disponível na forma de pacote ou simplesmente o pacote não vem compilado com alguma funcionalidade que desejamos.



- Entender os passos de uma compilação de programa;
- Instalar o NTOP no WebServerInterno;
- Entender sobre as bibliotecas compartilhadas e estática do Linux.

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

Servidor: Webserver Interno

- Vamos compilar um programa criado pela 4Linux para ensinar as etapas de uma compilação;
- A distribuições por padrão não esperam que você compile programas, portanto, não trazem as ferramentas necessárias para se compilar um programa:

```
1# yum install gcc gcc-c++ make
```

gcc → GNU Compiler Collection (GCC) foi desenvolvido originalmente por Richard Stallman;
make → Utilitário de automatização de compilação;

Anotações:

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.



Laboratório Dexter

Servidor: Webserver Interno

- Vamos compilar um programa criado pela 4Linux para ensinar as etapas de uma compilação:

```
1# cd /root/pacotes
2# tar -xf program.tar.gz -C /usr/local/src
3# cd /usr/local/src/program
4# vim README
5# ./configure
6# make
7# make install
8# which program
9# program
```

4LINUX

56

Comando Configure

Em geral, sempre que pegamos o código fonte de um programa ele virá com um aplicativo chamado configure que irá executar uma verificação em seu sistema a fim de verificar se ele dispõe de todos os componentes básicos para uma compilação bem sucedida.

Arquivo Makefile

A Makefile em geral é criada utilizando a ferramenta configure e o objetivo desta é automatizar os processos de compilação, verificação e instalação dos softwares.

A Makefile é estruturada em seções; cada uma delas realiza alguma tarefa específica. Em geral essas Makefiles vêm com pelo menos três seções padrão: default, install e clean.

Servidor: Webserver Interno

- ```
1# cd /root/pacotes
2# ls ntop*
```

## This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.



## Laboratório Dexter

*Servidor: Webserver Interno*

➤ Realize a instalação do NTOP conforme INSTALL:

```
1# tar jxvf ntopng-4linux.tar.bz2 -C /usr/local/src
2# cd /usr/local/src/ntopng
3# less README.ntopng
4# ./configure
```

checking for gcc option to accept ISO C89...  
 checking how to run the C preprocessor... gcc -E  
 checking for PF\_RING source installed in /root/PF\_RING... checking  
 for pcap\_open\_live in -lpcap... no  
**Please install libpcap(-dev) (<http://tcpdump.org>)**

### Dependências....



4LINUX

58

### Instalação

É muito comum, quando compilamos um programa a partir de seu código fonte, que alguns de seus requisitos não estejam presentes, ocasionando assim um erro durante a execução do configure.

Quando isso ocorrer, deve-se identificar o componente que está faltando, instalá-lo e executar novamente o configure até que ele termine com sucesso.



## Laboratório Dexter

Servidor: Webserver Interno

### Resolvendo Dependências:

```
1# yum groupinstall "Development tools"
```

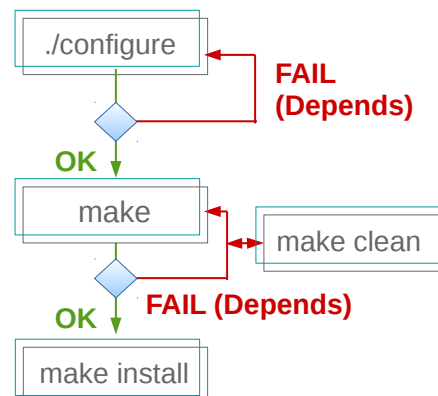
```
2# yum install libpcap-devel
```

```
3# ./configure
```

```
Please install libxml2(-devel)
```

```
4# yum install libxml2-devel
```

```
Please install libglib-2.0 (glib2-devel/libglib2.0-dev)
```



4LINUX

59

### Instalando dependências:

O pacote “**Development tools**” é uma série de pacotes onde contém todos os utilitários necessários referente a desenvolvimento, onde a grande maioria dos pacotes referente a Development (Desenvolvimento) são encontrados.

▶ Encurtando o caminho:

## 4LINUX

60

## This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.



## Laboratório Dexter

*Servidor: Webserver Interno*

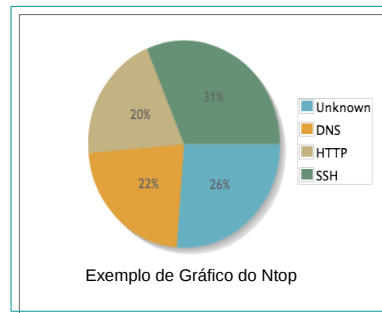
➤ Configurando o **NTOP**:

```
1# cd /usr/local/src/ntopng/confs4linux
2# cp ntopng.init /etc/init.d/ntopng
3# /etc/init.d/ntopng start
4# ss -ntlp | grep 3000
```

➤ Na máquina **Linux Interna** acesse:

192.168.200.20:3000 >

Usuário **admin** / Senha **admin**



**4LINUX**

**61**

### Script de Inicialização:

Os scripts de inicialização dos serviços são encontrados dentro do diretório `/etc/init.d` onde você terá mais embasamento na aula de inicialização de sistema.

Como estamos compilando o NTOP essa inclusão no diretório deve ser feita manualmente, portanto iremos copiar script chamado `ntopng.init` encontrado no diretório `../confs4linux`, para o diretório onde encontra-se os scripts de inicialização do sistema `/etc/init.d`.

- a. make
- b. install
- c. configure
- d. make depend
- e. make install

This image shows a full page of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page, providing a template for handwriting practice or general writing. There are no margins, text, or other markings on the page.

## Pergunta LPI



Comando usado para instalar uma aplicação através do código fonte, compilado através do arquivo Makefile:

- a. `make`
- b. `install`
- c. `configure`
- d. `make depend`
- e. `make install`

Resposta: e

4LINUX

63

**Alternativa E: RESPOSTA CORRETA!**

O comando “**make install**” é um utilitário utilizado para instalar aplicações compiladas diretamente pelo código fonte.

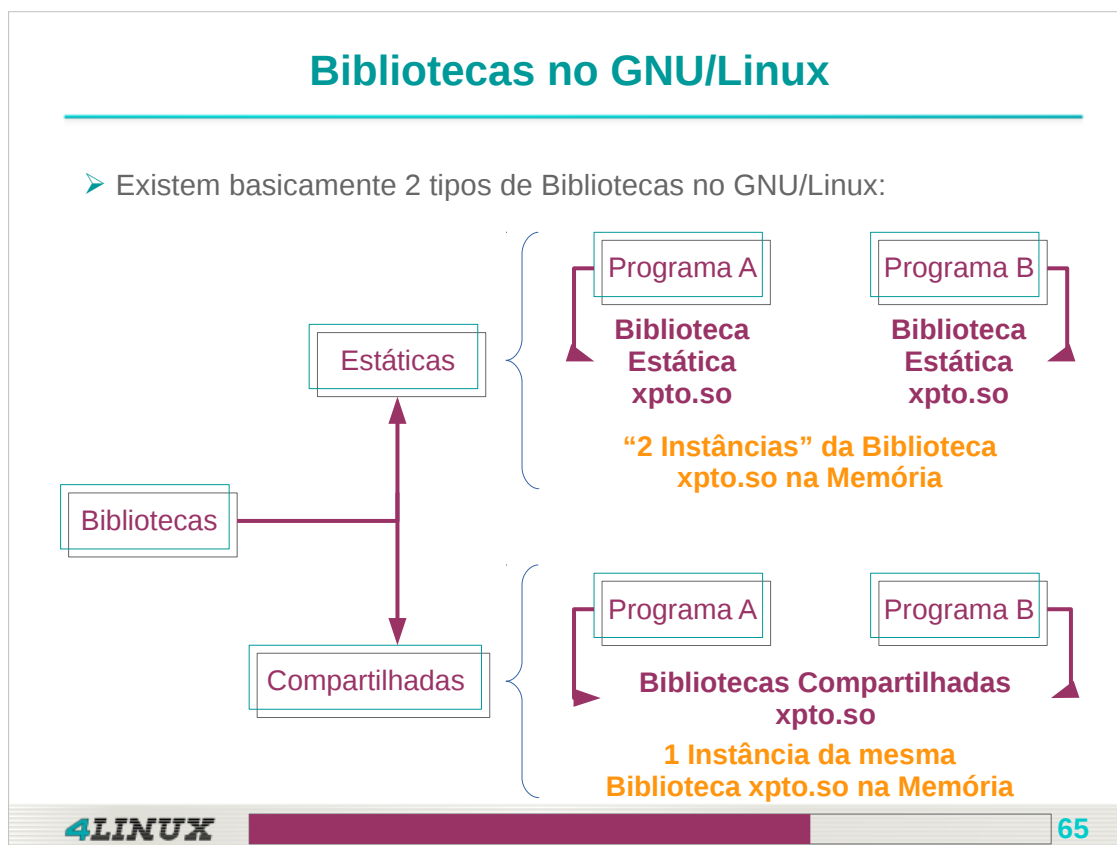
## Bibliotecas no GNU/Linux

**4LINUX****64**

Em linguagens de programação, entende-se por bibliotecas, arquivos que contêm um conjunto de módulos ou membros de códigos pré-compilados reutilizáveis. Esses códigos podem ser usados por várias aplicações de forma genérica, ou seja, sem que seja necessário uma implementação no momento da instalação/compilação desta aplicação.

A grande vantagem de usar uma biblioteca, é que uma vez criada, não será mais preciso compilar, bastando simplesmente ligá-la ao programa desejado. A função destas bibliotecas lembra um pouco a dos arquivos .dll no Windows.





### Tipos fundamentais de programas executáveis

Em sistemas Linux existem dois tipos fundamentais de programas executáveis. O primeiro é chamado de estático. Esse tipo de programa contém todas as funções que ele precisa para ser executado, em outras palavras, é completo. Devido a isso, os executáveis estáticos não dependem de nenhuma biblioteca externa para funcionar. O segundo tipo é o executável dinâmico.

## Estática ou Compartilhada?

### Bibliotecas Estáticas

- Binário roda de forma independente; ( + )
- Gasto desnecessário de memória; ( - )
- Tamanho maior no final do arquivo. ( - )

### Bibliotecas Dinâmicas

- Binário faz referências a várias bibliotecas; ( - / + )
- Gasto menor de memória; ( + )
- Tamanho menor final do arquivo. ( + )

### Modo Estático x Modo Compartilhado

Em relação aos tipos de Bibliotecas provavelmente a principal diferença entre a execução de aplicações no modo estático para a execução no modo dinâmico é que aplicações utilizando o modo estático são ligeiramente mais rápidas, pois não precisam buscar bibliotecas em outros diretórios, já aplicações utilizando o modo dinâmico apesar de serem mais lentas tendem a consumir menos espaço dado que vários programas compartilham dos mesmos recursos.

Em um sistema GNU/Linux, bibliotecas estáticas têm nomes como libname.a, enquanto bibliotecas compartilhadas são chamadas libname.so.x.y.z onde x.y.z é alguma forma de número de versão.

## Bibliotecas Compartilhadas

➤ Funcionamento das bibliotecas compartilhadas:



4LINUX

67

### Bibliotecas Compartilhadas

Apesar do modo compartilhado ser ligeiramente mais lento, que o modulo estático ainda é o modelo mais utilizado pois facilita a gerencia do próprio sistema operacional, centralizando a manutenção e o acesso aos recursos do sistema, por exemplo, se for necessário mudar a versão de uma biblioteca, não será preciso recompilar o programa, bastando apenas trocar o arquivo da biblioteca).

Assim o padrão do sistema é utilizar bibliotecas compartilhadas ao invés de Bibliotecas Estáticas.



## Laboratório Dexter

Servidor: Webserver Interno

### ➤ Praticando Bibliotecas:

```
1# which htop
2# ldd /bin/htop → Binário com Biblioteca Dinâmica
3# ldd /sbin/ldconfig → Binário com Biblioteca Estática
4# readelf -l /bin/ls
...
[Requesting program interpreter: /lib/ld-linux.so.2]
5# ls -ld /etc/ld*
6# cat /etc/ld.so.conf
7# rm -rf /etc/ld.so.cache
8# ldconfig -p
9# ldconfig
10# ls -ld /etc/ld*
11# ldconfig -p
```

4LINUX

68

### Onde ficam as bibliotecas?

Por padrão, os programas instalados já adicionam as bibliotecas em seus devidos diretórios, que geralmente são: `/lib`, `/usr/lib`.

A ordem de buscas por bibliotecas no sistema é:

- O valor da variável: `LD_LIBRARY_PATH`
- Os diretórios especificados em `/etc/ld.so.conf`
- Os diretórios padrão do sistema para bibliotecas: `/lib` e `/usr/lib`



## Laboratório Dexter

*Servidor: Webserver Interno*

➤ Simulando problemas de Biblioteca:

```
1# ls /etc/
2# which ls
3# ldd /bin/ls | grep libattr
4# mkdir /opt/lib
5# mv /lib64/libattr.so.1 /opt/lib/
6# ls /etc/
```



Note que ao movermos uma biblioteca para um diretório desconhecido do do ld.so (Loader) o comando ls parou de funcionar. **Vamos corrigir esse problema!**



A biblioteca **libattr.so** é utilizada pela maioria dos comandos no Linux.


**DICA:** Configurar o novo caminho da biblioteca.

**4LINUX**

69

### Referenciando Bibliotecas

No exemplo acima, não foi possível executar o comando ls, pois a biblioteca compartilhada não pôde ser carregada por estar em um diretório que não é referenciado em /etc/ld.so.cache para os linkadores carregarem-na.



## Laboratório Dexter

*Servidor: Webserver Interno*

➤ Simulando problemas de Biblioteca:

```
1# echo "/opt/lib/" >> /etc/ld.so.conf.d/bibliotecas.conf
2# ldconfig
3# ldd /bin/ls | grep libattr
4# ls /etc/
```

Pronto, agora o comando voltou a funcionar!

A biblioteca **libattr.so** é utilizada pela maioria dos comandos no Linux.

4LINUX 70

### Referenciando Bibliotecas

Você já imaginou ficar sem a librt.so.1? Sem essa lib, muitos recursos de movimentação pelo terminal não iram responder.

Nesses casos em que as bibliotecas estão em um diretório diferente é necessário dizer ao sistema para buscar bibliotecas em sua nova localização.

No exemplo acima simplesmente criamos um arquivo em /etc/ld.so.conf.d com a extensão .conf que referência a localização de nossa biblioteca.

Após criarmos a referência bastou atualizar a lista de diretórios de bibliotecas com o comando ldconfig.



( Coloque o comando com a biblioteca compartilhada)



71

**Anotações:**

[illegible]

## Pergunta LPI



Qual comando irá mostrar as bibliotecas compartilhadas do arquivo /usr/lib/libpng12.so?

Resposta: ldd /usr/lib/libpng12.so

4LINUX

72

**RESPOSTA CORRETA:** ldd /usr/lib/libpng12.so

O comando **ldd** é utilizado para listar as bibliotecas compartilhadas por um binário conforme testado em aula.



- Executar as tarefas do **Practice Lab**;
- Resolver o **Desafio Appliance Lab** e postar o resultado no Fórum Temático;
- Responder as questões do **Teste de Conhecimento** sobre o conteúdo visto em aula.

## Mãos à obra!

**Anotações:**

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

# 4LINUX

OPEN SOFTWARE SPECIALISTS



ESPECIALISTA EM "JUNTAR AS PEÇAS" DO MUNDO OPEN SOURCE

[WWW.4LINUX.COM.BR](http://WWW.4LINUX.COM.BR)