

fedora 



 ubuntu

 Mandriva

Curso de Formação LPIC-1

Exame 101



Curso Linux: formação

- › Instalação de software
 - › Métodos
 - › Instalação desde o código fonte
 - › Bibliotecas partilhadas
 - › Debian Package Manager
 - › Red Hat Package Manager

Instalação de Software

Métodos

- › Instalação de Software é efectuada de três formas
 - › Desde o código fonte
 - › Via pacotes
 - › Gestor de pacotes da distro (distribuição)
 - › manualmente

Instalação de Software

Instalação desde código fonte

- › O código fonte é o coração de toda a computação
- › Acesso ao código fonte é toda a natureza de Linux
- › Sem acesso ao código não se pode procurar por erros, questões de segurança ou outras falhas
- › Quando os programadores produzem o código fonte
 - › Tipicamente vem em *tar*
 - › Criam um arquivo comprimido do código
 - › Software é disponibilizado na internet

Instalação de Software

Instalação desde código fonte: vantagens

- › Permite pesquisar por falhas, erros e questões de segurança
- › É disponibilizado antes dos pacotes binários
- › Compilar otimiza o programa resultante para o sistema corrente
- › Alimenta a necessidade por software “cutting-edge” (testar aplicações em fase beta pode ser divertido)

Instalação de Software

Instalação desde código fonte: desvantagens

- › Tipicamente não se vê uma aplicação que está em desenvolvimento em qualquer outro formato que não código fonte
- › É demorado e nada comum criar um pacote binário de um código fonte até que a aplicação seja terminada ou chegue ao estado beta
- › Compilar um pacote fonte pode demorar horas, por isso, é inteligente começar com pacotes pequenos.
 - › rdesktop: 1m e 29s
 - › Openoffice 3.1: cerca de 6 horas
- › Requer uma instalação mais complexa
- › As dependências necessárias podem não existir ou ser muito difícil resolver
- › Muitas vezes está mal documentado, com apenas um ficheiro *readme* ou *install*
- › Remover aplicações instaladas a partir de código fonte pode ser complicado, principalmente se o Makefile do pacote não tiver uma rotina de desinstalação

Instalação de Software

Instalação desde código fonte

- Quando instalando a partir do código fonte, um número de conflitos e problemas de dependências pode existir.
- Até que um pacote seja distribuído por várias distros como parte da instalação, tipicamente não é para os fracos de coração.
- Se o objectivo for:
 - Sistema estável: esperar que a distribuição disponibilize uma versão
 - De outra forma, passar horas à procura de bibliotecas perdidas e outros ficheiros necessários
- Deve-se usar instalações a partir do código fonte quando precisamos de performance e segurança

Instalação de Software

Instalação desde código fonte: componentes

- *Tarballs* de um código fonte, tipicamente incluem:
 - Script de configuração – Isto verifica o sistema e configura o Makefile
 - Makefile – Este ficheiro define a localização da instalação, parâmetros e outras variáveis
 - Ficheiros fonte – Estas são as directorias e ficheiros do código fonte
 - Readme – este ficheiro contém informação importante sobre o processo de instalação
 - Install – Tipicamente, contém as actuais instruções para a instalação. **NÃO ESTÁ PRESENTE EM TODOS OS *TARBALLS***
 - Script de instalação – Geralmente apenas um ficheiro. Se executado, executa a instalação

Instalação de Software

Instalação desde código fonte: Makefile

- Um Makefile é um conjunto de instruções e parâmetros de compilação que tornam a instalação possível
- Após ser atualizado pelo script de configuração, tipicamente os comandos *make* e *make install* são corridos para compilar e instalar o software

O propósito do utilitário *make* é determinar que pedaços de um programa grande precisam de ser recompilados e executar os comandos para os recompilar.

Instalação de Software

Instalação desde código fonte: Makefile

- › Um Makefile tipicamente contém as secções:
 - › Platform – a plataforma do sistema
 - › Debug – Como gerir erros
 - › Optimize – Itens que são customizados pelo .configure
 - › Source – Onde os ficheiros fonte são encontrados
 - › Targets – All, install, clean, dist e por daí para a frente
- › O comando *make* por si só geralmente compila o código fonte, mas não faz mais nada.
- › Correr o *make clean* remove qualquer ficheiro temporário de uma tentativa anterior de compilar o código
- › *make install* coloca o código compilado nas directorias correctas na PATH e por aí adiante

Instalação de Software

Instalação desde código fonte: Makefile

- Um makefile típico tem as seguintes variáveis e PATHS (novamente, isto pode variar de programa para programa)

```
install-prefix =.  
bin_dir       = $(install-prefix)/bin  
uparm_dir     = $(install-prefix)/lib/uparm  
include_dir   = $(install-prefix)/include
```

- Estas informações indicam à instalação e outros *targets* onde colocar as várias peças do programa, com o \$ (*install-prefix*) a indicar que, após a compilação, o software é instalado em sub-directorias da directoria corrente:
 - ./bin
 - ./lib/uparm
 - ./include

Instalação de Software

Instalação desde código fonte

- › Efectuar download do pacote com o código fonte
- › Fazer o download da versão estável
- › Descomprimir o pacote
- › Ir para a nova directoria
- › ls para ver o conteúdo
- › Correr os scripts para a instalação

```
./configure --help (para ver opções que se podem passar)  
./configure  
make  
sudo make install
```

Instalação de Software

Bibliotecas partilhadas

- Bibliotecas fornecem acesso a funções comuns
- Bibliotecas partilhadas fornecem funções comuns a muitos programas diferentes em Linux
- Os programas “ligam-se” às bibliotecas de duas formas
 - Estaticamente – Têm um tamanho maior, mas menos dependências
 - Dinamicamente – Têm um tamanho menor, e não funciona se faltar
- Um bom exemplo de bibliotecas partilhadas é o uso de chamadas de sistema para
 - Abrir um ficheiro
 - Informações de um ficheiro com o *stat*
- Num Linux, as bibliotecas partilhadas encontram-se em
 - `/lib` – Maioria das bibliotecas partilhadas
 - `/usr/lib` – bibliotecas adicionais
 - `/usr/X11R6/lib` – Bibliotecas partilhadas do X
 - (sistemas a 64bits também têm `lib64` e `lib32`)

Instalação de Software

Bibliotecas partilhadas

- › Nestas directorias, encontramos dois tipos de bibliotecas partilhadas, algumas com a seguinte convenção:

```
nomebiblioteca-major-minor-patch.so (libc-2.8.90.so)  
nomebiblioteca.so.major-minor-patch (libc.so.6)
```

- › Reparar que `ls -l` mostra que alguns nomes de bibliotecas são *links* para outras.

```
ls libc* -l  
-rwxr-xr-x 1 root root 1315024 2009-01-29 10:23 libc-2.8.90.so  
lrwxrwxrwx 1 root root 14 2009-07-20 14:42 libcap.so.1 -> libcap.so.1.10  
-rw-r--r-- 1 root root 10316 2007-07-31 20:20 libcap.so.1.10  
lrwxrwxrwx 1 root root 14 2009-07-20 14:42 libcap.so.2 -> libcap.so.2.10  
-rw-r--r-- 1 root root 13792 2008-06-12 14:39 libcap.so.2.10  
-rw-r--r-- 1 root root 185820 2009-01-29 10:23 libcidn-2.8.90.so  
lrwxrwxrwx 1 root root 17 2009-07-20 15:15 libcidn.so.1 -> libcidn-2.8.90.so  
lrwxrwxrwx 1 root root 17 2009-07-20 14:42 libcom_err.so.2 -> libcom_err.so.2.1  
-rw-r--r-- 1 root root 9576 2008-10-13 14:09 libcom_err.so.2.1  
-rw-r--r-- 1 root root 38300 2009-01-29 10:23 libcrypt-2.8.90.so  
lrwxrwxrwx 1 root root 18 2009-07-20 15:15 libcrypt.so.1 -> libcrypt-2.8.90.so  
lrwxrwxrwx 1 root root 14 2009-07-20 15:15 libc.so.6 -> libc-2.8.90.so
```

Instalação de Software

Bibliotecas partilhadas

- Programas que esperam a presença de uma versão (e ficheiro) podem ser “enganados” com a criação de um *link* com aquele nome para a biblioteca requerida mais recente e com a mesma funcionalidade
- O sufixo *so* significa que é um objecto partilhado e as várias convenções de nomes suportam ambos os nomes mais recentes e os mais antigos.
- Usar *links* para as bibliotecas permite actualizações e substituição de bibliotecas por versões mais recentes
- Desta forma
 - Todos os programas que esperam versões mais antigas das bibliotecas usam o *link* com o nome antigo, que aponta para a versão mais recente, mas que contém a funcionalidade desejada

Instalação de Software

Bibliotecas partilhadas: determinar bibliotecas usadas

- Queremos saber se as bibliotecas que um programa precisa estão presentes no sistema, temos duas alternativas
- Corremos o programa e vemos se corre (forma insegura)
- Questionamos o programa pelas bibliotecas necessárias. Utilitário ldd

```
ldd /bin/ls
linux-gate.so.1 => (0xb7f3c000)
librt.so.1 => /lib/tls/i686/cmov/librt.so.1 (0xb7efc000)
libselinux.so.1 => /lib/libselinux.so.1 (0xb7ee2000)
libacl.so.1 => /lib/libacl.so.1 (0xb7ed9000)
libc.so.6 => /lib/tls/i686/cmov/libc.so.6 (0xb7d7b000)
libpthread.so.0 => /lib/tls/i686/cmov/libpthread.so.0 (0xb7d62000)
/lib/ld-linux.so.2 (0xb7f22000)
libdl.so.2 => /lib/tls/i686/cmov/libdl.so.2 (0xb7d5e000)
libattr.so.1 => /lib/libattr.so.1 (0xb7d59000)
```


Instalação de Software

Bibliotecas compartilhadas: determinar bibliotecas usadas

- › Estas bibliotecas têm que estar acessíveis ao programa de duas formas:
 - › Ou nas PATHS já mencionadas
 - › Ou definidas numa variável chamada `LD_LIBRARY_PATH`, que contém os caminhos das bibliotecas separadas por dois pontos (:)
- › Para declarar esta variável, podemos fazer de duas formas:
 - › Na linha de comandos, usando o `export`
 - › Ou editar o `/etc/profile` ou `~/.bash_profile`

Instalação de Software

Bibliotecas partilhadas: determinar bibliotecas usadas

- Algumas bibliotecas podem não conter informações suficientes para serem identificadas pelo programa `ldconfig` quando constroi os *links* nas directorias das bibliotecas
- O `ldconfig` lê os cabeçalhos de todas as bibliotecas em `/lib` e `/usr/lib` e em qualquer directoria definidas no ficheiro `/etc/ld.so.conf`.
- Desde estes cabeçalhos e os ficheiros nas directorias, o `ldconfig` cria os *links* necessários para as bibliotecas, principalmente para os programas mais antigos que esperam que bibliotecas mais antigas existam
- `ldconfig` também cria o ficheiro `/etc/ld.so.cache`
- Este ficheiro binário é uma lista ordenada de todas as bibliotecas existentes no sistema
- Este ficheiro não é lido como texto, mas é usado como uma referência rápida para o `ld.so`, o programa de lincagem do sistema.

Instalação de Software

Instalar software

- › Uma distro típica de Linux tem mais de 2000 pacotes, com vários ficheiros por pacote
- › Um pacote é um “contentor” que contém:
 - › Aplicações
 - › Ficheiros de configuração
 - › Outros ficheiros necessários
- › Exames LPI
 - › Debian Package Manager (DPKG)
 - › Red Hat Package Manager (RPM)

Instalação de Software

Instalar software: dpkg

- › Cada pacote Debian contém:
 - › Aplicação
 - › Ficheiros de configuração
 - › Documentação
 - › Dependências de outros pacotes
- › Os nomes dos pacotes têm três elementos comuns
 - › Nome do pacote – Pequeno e descritivo. Quando várias palavras são usadas, são separadas por hifên (-)
 - › Versão – Cada pacote contém uma versão. A maioria dos pacotes contém a mesma do software que contém. O formato varia de pacote para pacote, mas a maioria é numérica (major.minor.patchlevel)
 - › Uma extensão
 - › Todos os pacotes para Debian terminam em .deb

Instalação de Software

Instalar software: dpkg

Versão do software

pacote _ 1.0 - 2 . deb

Nome do pacote

Release do pacote

Instalação de Software

Instalar software: dpkg

- › A ferramenta original do debian para gerir pacotes é o dpkg
 - › Opera directamente no pacote .deb
 - › Pode ser usada para automatizar a instalação e manter os pacotes
- › Alternativa é a ferramenta apt-get
 - › Opera usando nomes de pacotes, obtendo-os de uma fonte pré-definida
- › O dselect oferece um menú interactivo que permite ao administrador escolher de uma lista de pacotes disponiveis, marcando-os para instalação
- › O comando alien permite o uso de pacotes não debian, como por exemplo RPM

Instalação de Software

Instalar software: dpkg

- › O comando do gestor de pacotes do debian
- › Consiste numa acção que especifica o modo de operação
- › Mantem as informações dos pacotes em */var/lib/dpkg*
- › Existem dois ficheiros de interesse
 - › Available – a lista de todos os pacotes disponiveis
 - › Status – contem atributos dos pacotes, como se está instalado ou marcado para remoção
- › Estes ficheiros são usados pelo dpkg, dselect e apt-get

Instalação de Software

Instalar software: dpkg – opções mais usadas

- › -E : não re-escrever um pacote previamente instalado que seja a mesma versão
- › -G : não re-escrever um pacote previamente instalado que seja de uma versão anterior
- › -R : processa pacotes recursivamente localizados em directorias e sub-directorias
- › --configure pacote : configura um pacote não configurado. Envolve o setup de ficheiros de configuração
- › -i (--install) : instala um pacote especificado. Envolve copia de segurança de ficheiros antigos, descomprimir e instalar os novos ficheiros, e configuração
- › -l [padrão] (--list [padrão]) : mostra informação para pacotes instalados cujo nome corresponda ao nosso padrão
- › -L (--listfiles) : Lista os ficheiros instalados pelo pacote
- › --print-avail : mostra os detalhes sobre o pacote encontrados em /var/lib/dpkg/available
- › --purge : remove tudo do pacote
- › -r (--remove) : remove tudo excepto ficheiros de configuração
- › -s (--status) : mostra o estado do pacote
- › --unpack : descomprime o pacote mas não o instala

Instalação de Software

Instalar software: dpkg

- Instalar um pacote
 - `dpkg -i iperf_2.0.4-1_i386.deb`

```
lpi@lpi:/var/cache/apt/archives$ sudo dpkg -i iperf_2.0.4-1_i386.deb
A seleccionar pacote anteriormente não seleccionado iperf
(A ler a base de dados ... 198391 ficheiros e directórios actualmente
instalados.)
A descompactar iperf (desde iperf_2.0.4-1_i386.deb) ...
A instalar iperf (2.0.4-1) ...

A processar 'triggers' para man-db ...
A processar 'triggers' para doc-base ...
Processing 1 added doc-base file(s)...
Registering documents with scrollkeeper...
```

Instalação de Software

Instalar software: dpkg

- Remover um pacote
- `dpkg --purge iperf`

```
sudo dpkg --purge iperf
(A ler a base de dados ... 198403 ficheiros e directórios actualmente
instalados.)
A remover iperf ...
A processar 'triggers' para doc-base ...
Processing 1 removed doc-base file(s)...
Registering documents with scrollkeeper...
A processar 'triggers' para man-db ...
```

Instalação de Software

Instalar software: dpkg

- › Listar pacotes instalados
 - › dpkg -l
- › Usando um nome
 - › dpkg -l iperf
- › Pacotes instalados aparecem com ii na primeira coluna

```
dpkg -l iperf
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Cfg-files/Unpacked/Failed-cfg/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=bad)
||/ Nome          Versão          Descrição
+++-----
pn iperf          <nenhum>        (sem descrição disponível)
```

```
dpkg -l apt
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Cfg-files/Unpacked/Failed-cfg/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=bad)
||/ Nome          Versão          Descrição
+++-----
ii apt            0.7.14ubuntu6. Advanced front-end for dpkg
```

Instalação de Software

Instalar software: dpkg

- › Saber que pacotes têm que ficheiros
 - › `dpkg -S apt-get`
- › Verificar o estado de um pacote
 - › `dpkg -s apt`
- › Determinar em que pacote foi instalado o ficheiro
 - › `dpkg -S /bin/lis`

```
dpkg -s apt
Package: apt
Status: install ok installed
Priority: important
Section: admin
Installed-Size: 5288
(lista continua)
```

```
dpkg -S apt-get
apt: /usr/bin/apt-get
apt: /usr/share/man/es/man8/apt-get.8.gz
apt: /usr/share/man/fr/man8/apt-get.8.gz
apt: /usr/share/man/man8/apt-get.8.gz
apt: /usr/share/man/ja/man8/apt-get.8.gz
```

```
dpkg -S /bin/lis
coreutils: /bin/lis
```

Instalação de Software

Instalar software: dpkg

- › Listar que ficheiros vêm num pacote
 - › `dpkg -L apt`

```
dpkg -L apt
./
/usr
/usr/bin
/usr/bin/apt-cache
/usr/bin/apt-cdrom
/usr/bin/apt-config
/usr/bin/apt-get
/usr/bin/apt-key
/usr/bin/apt-mark
/usr/lib
/usr/lib/apt
/usr/lib/apt/methods
(lista continua)
```

Instalação de Software

Instalar software: apt

- › Advanced Package Tool (APT)
 - › Não trabalha directamente com os ficheiros .deb
 - › Usa os nomes dos pacotes
-
- › Apt-get
 - › Parte do APT
 - › Mantém uma base de dados de pacotes que permite à ferramenta automaticamente actualizar pacotes e as suas dependências assim que novas versões sejam disponibilizadas
 - › Mantem os pacotes em /var/cache/apt/archives

```
apt-get [opções] comando
```

Instalação de Software

Instalar software: apt-get

- Opções mais frequentes
 - -d : faz o download dos pacotes, mas não os instala
 - -s : simula os passos na instalação, mas não altera o sistema. Chamado de “dry-run”
 - -y : automaticamente responde sim a todas as questões

- Comandos mais frequentes
 - dist-upgrade : actualiza automaticamente para novas distribuições
 - install : instala ou actualiza pacotes pelo nome
 - remove : remove pacotes específicos
 - update : actualiza a lista de pacotes disponíveis
 - upgrade : actualiza todos os pacotes de um sistema para as versões correntes. Este comando é conservador e não processa actualizações que possam corromper o sistema.

Instalação de Software

Instalar software: apt-get

- apt-get usa uma lista de localizações para efectuar o download dos pacotes
- Esta lista encontra-se em `/etc/apt/sources.list`

```
deb http://pt.archive.ubuntu.com/ubuntu/ intrepid multiverse
```

```
deb-src http://pt.archive.ubuntu.com/ubuntu/ intrepid multiverse
```

- deb – tipo de pacote
- <http://pt.archive.ubuntu.com/ubuntu> - URL do servidor
- intrepid – o “bolso” versão dos pacotes a ir buscar
- multiverse – os componentes

Instalação de Software

Instalar software: apt-cache

- › Manipula a cache do APT
- › Executa operações de procura
- › Gera informações sobre os pacotes
- › Opções mais usadas
 - › showpkg : mostra informações sobre os pacotes
 - › search : procura na cache por pacotes que possam corresponder à nossa procura. Pode usar ER
 - › depends : mostra uma lista de dependências que um pacote tem
 - › rdepends : mostra uma lista de dependências reversa. Pacotes que precisam deste pacote

Instalação de Software

Instalar software: alien

- › Converte ou instala um pacote não debian (ou alien)
- › Formatos suportados incluem
 - › Red Hat rpm
 - › Stampede slp
 - › Slackware tgz
 - › Genericos tar.gz
- › Para o suporte de RPM, o utilitário rpm tem que estar instalado no sistema
- › Opções mais usadas
 - › -i : automaticamente instala o pacote resultante e apaga o ficheiro convertido
 - › -r : converte o pacote para o formato RPM
 - › -t : converte o pacote para o arquivo tar.gz

Instalação de Software

Instalar software: Red Hat Package Management

- › RPM refere não só o formato do pacote
 - › O comando rpm
 - › A base de dados RPM
- › Componentes do RPM
 - › Base de dados RPM – Colecção de ficheiros que gere o RPM num sistema
 - › Ficheiros do pacote – O pacote em formato RPM que são distribuídos com a extensão .rpm
 - › O comando rpm – usado para todas as instalações, remoções e pesquisas de pacotes

Instalação de Software

Instalar software: Base de dados RPM

- › Colecção de ficheiros da base de dados Berkley
 - › Juntos criam a base de dados de pacotes e os ficheiros instalados no sistema pelos pacotes
- › Localizada em */var/lib/rpm*
- › Cada ficheiro neste directoria é parte da base de dados como um todo
- › Como qualquer base de dados, pode tornar-se desactualizada ou corrompida com o uso massivo
 - › `rpm --rebuilddb`
- › A base de dados é bloqueada durante a maioria das operações.
- › Qualquer tentativa de correr comandos adicionais resulta num erro.

Instalação de Software

Instalar software: Pacotes RPM

- › Pacotes RPM vêm em duas formas
 - › Código fonte (source RPM)
 - › Binários
- › Código fonte (Source RPM)
 - › Coleção de ficheiros de código fonte precisos para instalar o pacote
 - › Instruções de como construir um RPM (binário) a partir dos ficheiros fonte
- › Binário RPM
 - › É um pacote discreto que, quando instalado, copia os ficheiros binários, outros ficheiros associados (documentação, etc) para localizações específicas
 - › Contêm um número de ficheiros e scripts de configuração que compõem a aplicação ou o programa.
 - › Excepções são os pacotes que apenas colocam ficheiros de configuração, como os da directoria /etc.

Instalação de Software

Instalar software: Pacotes RPM

- Os pacotes RPM devem conter pelo menos
 - Ficheiros binários comprimidos (aplicação)
 - Nome e versão do pacote de software
 - A data de construção e o nome do computador onde foi criado
 - Uma descrição do pacote e qual o seu propósito
 - As chaves de segurança (checksums) e as dependências precisas

Instalação de Software

Instalar software: Convenções dos nomes dos pacotes

- Os pacotes instalados são conhecidos por um nome pequeno
- Algumas operações precisam do que sejam usados os nomes longos
- Tipicamente, os nomes longos são precisos quando queremos apagar algum, e existem vários pacotes com nomes semelhantes

```
cups  
cups-1.4-0.b2.18.fc11.i586
```

```
kernel-2.6.29.4-167.fc11.i586  
kernel-2.6.29.5-191.fc11.i586
```

Instalação de Software

Instalar software: comando rpm

- › A ferramenta rpm em modo de texto é usada para realizar todas as tarefas em pacotes no sistema
 - › Instalados e na base de dados RPM
 - › Ou não instalados e presentes no disco
- › Operações comuns incluem
 - › Instalação de pacotes
 - › Actualização de pacotes
 - › Remoção de pacotes
 - › Pesquisar a base de dados para informações
 - › Verificar o pacote
 - › Verificar pacotes instalados
 - › Construir um rpm desde o seu código fonte
- › As opções do rpm vêm em diferentes formatos
- › As opções comuns são pequenas e usam uma letra
 - › Existem quase sempre equivalentes em formato longo
- › Algumas opções só existem em formato longo

Instalação de Software

Instalar software: validação de pacotes

- › É importante validar ou verificar as assinaturas de um pacote cujo o download é efectuado de qualquer local (excepto dos sites e mirrors oficiais)
- › Desta forma temos a certeza que o pacote é original
- › Validar um pacote tipicamente é feito quando o instalamos ou acedido, mas pode ser efectuado explicitamente
- › Verificar a assinatura md5 de um pacote

```
rpm -K rdesktop-1.6.0-4.fc11.i586.rpm  
rdesktop-1.6.0-4.fc11.i586.rpm: rsa sha1 (md5) pgp md5 OK
```

```
rpm -K mp3splt-2.2.5-5.fc11.i586.rpm  
mp3splt-2.2.5-5.fc11.i586.rpm: (SHA1) DSA sha1 md5 (GPG) NOT OK (MISSING KEYS:GPG#66534c2b)
```

Instalação de Software

Instalar software: Instalação de pacotes

- › Instalar pacotes requer acesso de *root*
- › A instalação causa o comando rpm verificar as seguintes condições
 - › Existe espaço em disco suficiente
 - › Ficheiros existentes não serão re-escritos
 - › Todas as dependências listadas no pacote são validadas
- › A instalação é efectuada através da opção -i (--install)
- › Para instalar um pacote, obter uma barra de progresso e mensagens no ecrã
 - › `rpm -ivh rdesktop-1.6.0-4.fc11.i586.rpm`

```
rpm -ivh rdesktop-1.6.0-4.fc11.i586.rpm
Preparing...      ##### [100%]
 1:rdesktop      ##### [100%]
```

```
rpm -ivh *.rpm
```

Instalação de Software

Instalar software: Opções adicionais de instalação

- Quando se instalam pacotes RPM, devemos recordar que vão existir circunstâncias como ficheiros que existem no sistema que o RPM desconhece mas que serão re-escritos na instalação de um pacote
- Estes casos podem ser contornados com as opções *replacefiles* ou *force*

```
rpm -ivh tarfoo-1.2-3.i386.rpm  
Tarfoo    /etc/tarfoo.conf conflicts with file from snafu-1.1
```

```
rpm -ivh --force tarfoo-1.2-3.i386.rpm
```

```
Preparing...      ##### [100%]  
1:tarfoo          ##### [100%]
```

Instalação de Software

Instalar software: Opções adicionais de instalação

- Quando queremos substituir um pacote já instalado, mas que por alguma razão, algum ficheiro se perdeu ou destruimos o ficheiro de configuração, podemos usar a opção `--force` (tal como já visto) para forçar a instalação

```
rpm -ivh rdesktop-1.6.0-4.fc11.i586.rpm
Preparing...      ##### [100%]
package rdesktop-1.6.0-4.fc11.i586 is already installed
```

```
rpm -ivh --force rdesktop-1.6.0-4.fc11.i586.rpm
Preparing...      ##### [100%]
1:rdesktop        ##### [100%]
```

Instalação de Software

Instalar software: Opções adicionais de instalação

- Se o pacote não instala devido a dependências, é possível usar a opção `--nodeps` para forçar.
- SE AS DEPENDÊNCIAS NÃO FOREM VALIDADAS, A APLICAÇÃO NÃO VAI FUNCIONAR
- Usar a opção `--nodeps` sem fundamento ou sem saber exactamente o que se pretende irá causar problemas.

```
rpm -ivh --nodeps pebkac.rpm
Error: failed dependencies:
  libc.so.6 (GLIBC_2.3) is needed by pebkac-7.23-20031119.5
  libpthread.so.0 (GLIBC_2.3) is needed by pebkac-7.23-20031119.5
  libstdc++.so.5 (CXXABI_1.2) is needed by pebkac-7.23-20031119.5
```

Instalação de Software

Instalar software: Verificar a integridade dos pacotes

- Após a instalação de um pacote, devemos periodicamente verificar que o pacote está a funcionar correctamente e que todos os pacotes estão ok
- O mecanismo de verificação usa nove caracteres até ao extremo esquerdo da saída para indicar o que se passa com o pacote
 - S – o tamanho do ficheiro é diferente
 - M – o modo é diferente (permissões ou tipo de ficheiro)
 - 5 – a assinatura md5 é diferente (visto também com `-checksig`)
 - D – o número major/minor do dispositivo é diferente
 - L – existe um problema com a path de um link (`readLink`)
 - U – dono do ficheiro foi alterado
 - G – o grupo do ficheiro foi alterado
 - T – o mtime é diferente do original
 - C – um ficheiro de configuração foi alterado
- Opção `-V` ou `--verify`

Instalação de Software

Instalar software: Verificar a integridade dos pacotes

- Usa-se o nome pequeno do pacote para verificar a integridade de um pacote apenas

```
rpm -V setup  
S.5....T. c /etc/printcap
```

- Para todos os pacotes do sistema
 - rpm -Va

```
rpm -Va  
prelink: /usr/lib/libGL.so.1.2.#prelink#.h7cPUg Could not trace symbol resolving  
S.?..... /usr/lib/libGL.so.1.2  
S.5....T. c /var/log/mail/statistics  
S.5....T. c /etc/ldap.conf
```

Instalação de Software

Instalar software: Freshening Vs Upgrading

- A manutenção diária de um sistema vai requerer a instalação de actualizações e correcções em .rpm
- Aplicar correctamente estas actualizações requer algum tacto quando se tem um sistema que precisa de se manter o mais estável possível
- As opções -U e --upgrade estão desenhadas para **instalar** e/ou actualizar pacotes para as ultimas versões
- Tem como consequência que apenas a versão mais recente do pacote fica instalada, removendo todas as restantes

```
rpm -U *.rpm
```

- Actualiza o sistema para as ultimas versões dos RPMs existentes num directório particular
- Se no directório existirem ficheiros .rpm de aplicações **NÃO INSTALADAS** no servidor, estes serão instalados.

Instalação de Software

Instalar software: Freshening Vs Upgrading

- Freshening é diferente num aspecto importante
 - Se o pacote está instalado e um novo é invocado com a opção -F (ou –freshen) é actualizado para a ultima versão
 - Se o pacote não está instalado, não é actualizado ou instalado

```
rpm -F *.rpm
```

- Actualiza o sistema para as ultimas versões dos RPMs existentes num directório particular
- Se no directório existirem ficheiros .rpm de aplicações **NÃO INSTALADAS** no servidor, **NÃO SERÃO INSTALADOS**

Instalação de Software

Instalar software: Remover pacotes

- › Remover um pacote ou um conjunto de pacotes é apenas uma questão de usar a opção `-e` ou `-erase`
- › Requer acesso de *root*
- › Remover pacotes pode ser mais complicado que instalar, pois com o tempo, o pacote pode estar “enterrado” numa lista de dependências
- › Remover um pacote sem dependências é simples
 - › `rpm -e rdesktop`
- › O problema começa quando temos pacotes com várias versões instaladas

```
rpm -e rdesktop
```

```
rpm -e kernel-devel  
error: "kernel-devel" specifies multiple packages:  
kernel-devel-2.6.29.5-191.fc11.i586  
kernel-devel-2.6.29.4-167.fc11.i586
```

```
rpm -e kernel-devel-2.6.29.4-167.fc11.i586
```

```
rpm -e kernel-devel --allmatches
```

Instalação de Software

Instalar software: questionar pacotes

- Questionar pacotes por dados é apenas um dos passos quando gerimos pacotes
- É importante saber a diferença sobre pacotes que estão instalados (aparecem na base de dados RPM) e pacotes que existem no sistema como ficheiros ou outro recurso
- A diferença é adicionar a opção -p que indica que o pacote questionado existe como um ficheiro no disco
- A query mais simples e comum, que devolve se um pacote existe ou não no sistema
 - rpm -q iperf
- Para obtermos mais informação sobre o pacote, adicionamos a opção i (info)

```
rpm -q iperf  
iperf-2.0.4-2.fc11.i586
```

Instalação de Software

Instalar software: questionar pacotes

```
rpm -qi iperf
Name       : iperf                Relocations: (not relocatable)
Version    : 2.0.4                Vendor: Fedora Project
Release    : 2.fc11              Build Date: Wed 25 Feb 2009 07:47:27 AM WET
Install Date: Wed 22 Jul 2009 03:34:09 PM WEST   Build Host: x86-6.fedora.phx.redhat.com
Group      : Applications/Internet   Source RPM: iperf-2.0.4-2.fc11.src.rpm
Size       : 114881               License: BSD
Signature  : RSA/8, Wed 11 Mar 2009 10:49:59 PM WET, Key ID 1dc5c758d22e77f2
Packager   : Fedora Project
URL        : http://iperf.sourceforge.net
Summary    : Measurement tool for TCP/UDP bandwidth performance
Description :
Iperf is a tool to measure maximum TCP bandwidth, allowing the tuning of
various parameters and UDP characteristics. Iperf reports bandwidth, delay
jitter, datagram loss.
```

Instalação de Software

Instalar software: questionar pacotes

- Saber informação sobre os ficheiros num pacote, usamos a opção `-q` (query) e `-l` (file listing)

```
rpm -ql iperf
/usr/bin/iperf
/usr/share/doc/iperf-2.0.4
/usr/share/doc/iperf-2.0.4/AUTHORS
/usr/share/doc/iperf-2.0.4/COPYING
/usr/share/doc/iperf-2.0.4/ChangeLog
/usr/share/doc/iperf-2.0.4/README
/usr/share/doc/iperf-2.0.4/dast.gif
/usr/share/doc/iperf-2.0.4/index.html
/usr/share/doc/iperf-2.0.4/ui_license.html
```

- Recordar que, para pacotes não instalados, temos que adicionar a opção `-p`

Instalação de Software

Instalar software: questionar pacotes

- › Saber quais as dependências de um pacote, usamos a opção -R ou -requires
- › Saber em que pacote vem determinado ficheiro, usamos a opção -f e o ficheiro (caminho completo)

```
rpm -qf /bin/ls  
coreutils-7.2-1.fc11.i586
```

```
rpm -qR tree  
libc.so.6  
libc.so.6(GLIBC_2.0)  
libc.so.6(GLIBC_2.1)  
libc.so.6(GLIBC_2.2)  
libc.so.6(GLIBC_2.3)  
libc.so.6(GLIBC_2.3.4)  
libc.so.6(GLIBC_2.4)  
rpmllib(CompressedFileNames) <= 3.0.4-1  
rpmllib(FileDigests) <= 4.6.0-1  
rpmllib(PayloadFilesHavePrefix) <= 4.0-1  
rtld(GNU_HASH)
```

Instalação de Software

Instalar software: yum

- › Yellowdog Updater, Modified (YUM)
- › Usado em Red Hat e Fedora Linux
- › Sistema de gestão de software completo
- › Desenhado para usar rede informática/internet
- › Uso de repositórios
 - › Directoria num servidor/web site que contém pacotes e ficheiros de index
 - › yum automaticamente localiza esses repositórios e obtêm os pacotes correctos para o sistema
- › Automaticamente resolve problemas de dependências
- › Instalado num sistema Fedora base
- › Necessário instalar num sistema Red Hat

Instalação de Software

Instalar software: yum

- O yum modifica o software no sistema em quatro formas
 - Instalar novo software desde repositórios
 - Instalar novo software desde um pacote individual
 - Actualizar software existente no sistema
 - Remover software do sistema
- O yum apenas mostra e usa as versões mais recentes de cada pacote, excepto se for especificada uma versão mais antiga

Instalação de Software

Instalar software: yum.conf

- Yum é configurado usando um ficheiro de texto chamado yum.conf
- Este ficheiro encontra-se em /etc/yum.conf
- Nele estão definidas algumas opções que modificam o comportamento do yum e os repositórios (servidores) onde irá buscar o software

- Existem dois tipos de secções no yum.conf
 - A secção [main] define todas as opções globais de configuração
 - A secção [server] define entradas para cada servidor
- As secções [server] também são afectadas pelas opções em [main], mas também possuem as suas próprias opções

```
[main]
cachedir=/var/cache/yum
debuglevel=2
logfile=/var/log/yum.log
pkgpolicy=newest
distroverpkg=redhat-release
```

```
option=value
```

Instalação de Software

Instalar software: yum.conf

- Cada secção [server] deverá conter pelo menos
 - O nome do servidor
 - O URL base

```
[google]
name=Google - i386
baseurl=http://dl.google.com/linux/rpm/stable/i386
enabled=1
gpgcheck=1
```

Instalação de Software

Instalar software: yum

- › Para cada operação, o yum faz o download da última informação sobre o pacote dos repositórios configurados
- › Utiliza depois essa informação para determinar as melhores acções para produzir o resultado desejado
- › Mostra a transacção para aprovação por parte do administrador
- › A transacção pode incluir
 - › Instalação
 - › Actualização
 - › remoção
- › Resolução de dependências

Instalação de Software

Instalar software: yum

```
=====
Package           Arch    Version    Repository    Size
=====
Installing:
tsclient          i386    0.132-6    base          247 k
Installing for dependencies:
rdesktop          i386    1.4.0-2    base          107 k

Transaction Summary
=====
Install      2 Package(s)
Update      0 Package(s)
Remove      0 Package(s)
Total download size: 355 k
Is this ok [y/N] :
```

Instalação de Software

Instalar software: yum

- O yum também se preocupa com a autenticidade dos repositórios
- Consoante a nossa resposta, instala automaticamente a chave para o repositório

```
warning: rpmts_HdrFromFdno: Header V3 DSA signature: NOKEY, key ID 4f2a6fd2
public key not available for tsclient-0.132-6.i386.rpm
Retrieving GPG key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora
Importing GPG key 0x4F2A6FD2 "Fedora Project <fedora@redhat.com>"
Is this ok [y/N] :
```

Instalação de Software

Instalar software: yum

- Para instalar uma aplicação, basta indicar no yum, com a opção *install*
 - `yum install deluge`
- Actualizar todo o sistema com o yum
 - `yum update`
- Manter o sistema actualizado, basta colocar o serviço do yum a correr
 - `chkconfig --level 345 yum on`

Instalação de Software

Instalar software: yum

- Para remover uma aplicação, basta indicar no yum, com a opção *remove*
 - `yum remove deluge`
- Actualizar pacotes, também se torna simples
 - `yum update deluge`
- O yum apenas instala sempre a ultima versão do pacote disponível
- Excepção para o kernel
 - Instala a ultima versão
 - Mantém a versão anterior instalada, caso aconteçam problemas

Instalação de Software

Instalar software: yum

- › O yum tem um motor de pesquisa muito completo e poderoso
- › Procura automaticamente por pacotes já instalados ou disponíveis nos repositórios
- › Procurar pelo nome de um pacote específico, usa-se a opção *list*
- › Também é possível procurar por versões específicas

```
yum list tsclient  
yum list tsclient-0.132
```


Instalação de Software

Instalar software: yum

- › Quando não se sabe o nome do pacote, usamos a opção *search*
- › A opção *search* procura
 - › Nomes
 - › Descrições
 - › Sumários
 - › Criadores dos pacotes
- › Esta procura é estendida a todos os pacotes disponíveis

```
yum list torrent  
yum search torrent
```

Instalação de Software

Instalar software: yum

- Por vezes não sabemos o nome do pacote, mas sabemos um ficheiro que ele fornece. Usando a opção *provides* conseguimos saber quais os pacotes que fornecem aquele ficheiro

```
yum provides libnet
```

- Se na eventualidade existirem vários pacotes que fornecem o mesmo ficheiro ou se o yum achar que a palavra é errónea ou que existem mais ficheiros no sistema que são fornecidos por vários pacotes

```
yum provides libneon  
Loaded plugins: downloadonly, refresh-packagekit  
Warning: 3.0.x versions of yum would erroneously match against filenames.  
You can use "*/libneon" and/or "**bin/libneon" to get that behaviour  
No Matches found
```

Instalação de Software

Instalar software: yum

- › A procura do yum também aceita *wildcards*
- › Para os *wildcards*, usamos a opção *list*
- › Algumas regras da ER também se aplicam

```
yum list tsc\*
```

```
yum list shadowutils  
yum list shadow-util  
yum list shadow-util?  
yum list shadow*
```

Instalação de Software

Instalar software: yumdownloader

- › Programa para fazer o download dos pacotes (em vez da instalação)
- › Usa os mesmos repositórios que o yum
- › Parte do pacote yum-utils
- › Opções mais usadas
 - › --destdir : directório onde colocar o pacote após o download
 - › --source : em vez de fazer o download dos binários, efectua do código fonte
 - › --resolve : quando estiver a efectuar o download, resolve as dependências e efectua também o download dos pacotes

```
yumdownloader --source kernel  
yumdownloader tomboy  
yumdownloader --resolve tomboy
```

Curso Linux

bibliografia

- › LPIC I, Exam Cram 2, Brunson - QUE Certification
- › LPI Linux Certification In a Nutshell, Pritchard, Pessanha, Langfeldt, Stranger & Dean – O REILLY
- › Linux Administration Handbook, Second edition, Nemeth Snyder Hein – Prentice Hall