



Título do Curso:

Administração de Sistemas
GNU/Linux



Conteúdo do curso

ADMINISTRAÇÃO DE SISTEMAS GNU/LINUX

- ✓ Tópico 1: Introdução ao sistema operacional GNU/Linux.
- ✓ Tópico 2: Introdução ao Shell e comandos básicos.
- ✓ Tópico 3: Manipulação de conteúdos com comandos no Shell.
- ✓ Tópico 4: Comandos para gerenciamento do sistema e do Hardware.
- ✓ Tópico 5: Editor de Texto VI.
- ✓ Tópico 6: Administração de usuários e grupos.
- ✓ Tópico 7: Gerenciamento de permissões.
- ✓ Tópico 8: Gerenciamento de processos.
- ✓ Tópico 9: Sistemas de arquivos e particionamento.
- ✓ Tópico 10: Expressões regulares.
- ✓ Tópico 11: Introdução ao Shell Script.
- **Tópico 12: Gerenciamento de Pacotes.**
- Tópico 13: Agendamento de tarefas (cron) e Backup.





DGP

Tecnologia da Informação

Tópico 12

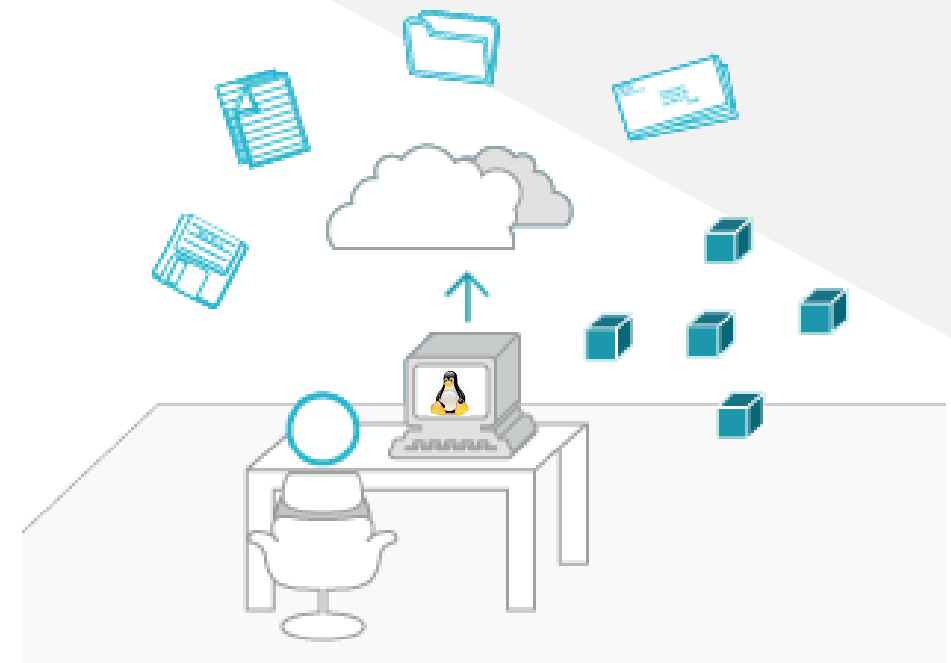
Gerenciamento de Pacotes



Gerenciamento de Pacotes

APT, YUM, YAST...

- Neste slide teremos comandos que nos auxiliam no gerenciamento de pacotes, bem como as diferenças entre as principais distribuições:



Conceitos e aplicabilidade

Tópico 12: Gerenciamento de Pacotes.



Introdução ao Gerenciamento de Pacotes – História!

- Inicialmente (desde os primórdios do mundo **UNIX** até os dias atuais), os pacotes eram distribuídos através de arquivos empacotados ou compactados (extensões “**tar.gz**” ou “**tar.bz2**”), contendo o código fonte do programa.
- Este formato pode ser utilizado para instalação em qualquer “**distro**” e permite que um programador modifique e realize adaptações no código fonte do pacote, visando a implementação de melhorias, customizações entre outras modificações.
- Com o passar do tempo e a criação do **GNU/Linux**, diversas distribuições foram criadas, cada uma delas com suas particularidades e com o “**Gerenciamento de Pacotes**” não foi diferente.



Introdução ao Gerenciamento de Pacotes

- Atualmente temos três possibilidades para a instalação de pacotes:
 - Através do **código fonte**;
 - Através de um **pacote pré-compilado**;
 - Através de um **gerenciador de pacotes** (que utiliza pacotes pré-compilados);
- A instalação de um pacote através do **código fonte** é a única forma idêntica em qualquer distribuição, ou seja, o processo pode ser realizado em qualquer *distro*.
- A instalação através de **pacotes pré-compilados** apresenta diferenças de sintaxe e comandos entre distribuições **e o pacote deve ser específico** para a distribuição utilizada.
- A instalação com “**Gerenciadores de Pacotes**” é considerada a forma mais simples, porém, **depende de configuração prévia dos repositórios de pacotes** (conteúdo que veremos nos próximos slides).



Conceitos Importantes!

- **Pacote** → Arquivo (comprimido ou não) contendo dados de um determinado software (**código-fonte**, arquivos do software, ícones, documentações, scripts de inicialização, entre outros).
- **Código fonte** → Programa escrito pelo programador (linguagem de alto nível);
- **Compilar** → Ato de “transformar” o código fonte de um programa (linguagem de alto nível) em um programa executável (binário - linguagem de máquina);
- **Compilador** → Programa que converte o código fonte (escrito pelo programador), para a linguagem de máquina (programa executável);
 - Após a compilação, o programa estará pronto para ser executado como um arquivo binário.
 - Temos diversos compiladores no ambiente **GNU/Linux**, cada um com suas particularidades. Dentre eles, o mais usado é o “**gcc**” (linguagem “C”);

Conceitos Importantes!

Dependências...



- Ao instalar um pacote, podemos nos deparar com o seguinte problema:
 - *“Erro – A instalação do pacote XPTO requer que o XYZ esteja instalado.”*
- Este tipo de problema é chamado de “**dependência**”, ou seja, quais pacotes são “**pré-requisitos**” para a instalação de outro pacote.
- Portanto, **dependências** são pacotes requeridos para a instalação de um determinado pacote.

Instalação de um pacote através do código fonte

Tópico 12: Gerenciamento de
Pacotes.



Instalação através do código fonte

- A instalação através do **código-fonte** é considerada por muitos, uma das formas mais complexas de instalação de um pacote, devido a necessidade de executar uma sequência de comandos específica.
 - 1º Passo → **descompactar** o “**pacote.tar.gz**”, através do comando “**tar**”;
 - 2º Passo → **Preparar a instalação do pacote** com as opções desejadas.
 - Ao descompactar, temos acesso ao **código fonte** do pacote, ou seja, podemos alterar o software conforme a necessidade de uso antes de realizar a **compilação** do código fonte, **gerando os binários** e demais arquivos necessários para **instalação** do pacote no sistema.
 - Devemos **preparar o pacote** com os parâmetros/opções desejadas **antes do código fonte ser compilado**. Na maioria das vezes este segundo passo (preparação) é feita através do comando “**./configure**”.



Instalação através do código fonte

- Continuação....:
 - 3º Passo → **Compilar o pacote**, gerando os binários e demais arquivos necessários para a instalação do pacote a partir do código fonte e opções utilizadas na “preparação”, descrito no passo anterior.
 - Comando para compilar o pacote → “**make**”;
 - 4º Passo → **Instalar o pacote**, copiando os arquivos gerados durante a compilação para o diretório previamente definido na 2ª etapa (com o comando “**./configure**”).
 - A instalação é realizada através do comando → “**make install**”.



Exemplo:

Instalando através do código fonte

```
root@darkstar:/var/src# ls
squid-3.0.STABLE25.tar.gz*
root@darkstar:/var/src# tar xzf squid-3.0.STABLE25.tar.gz
root@darkstar:/var/src# ls
squid-3.0.STABLE25/ squid-3.0.STABLE25.tar.gz*
root@darkstar:/var/src# cd squid-3.0.STABLE25
root@darkstar:/var/src/squid-3.0.STABLE25# ls
CONTRIBUTORS  Makefile.in      aclocal.m4      errors/         src/
COPYING        QUICKSTART       bootstrap.sh*   helpers/       test-suite/
COPYRIGHT      README           cfgaux/         icons/         tools/
CREDITS        RELEASENOTES.html configure*       include/
ChangeLog      SPONSORS         configure.in    lib/
INSTALL        TODO             contrib/       scripts/
Makefile.am    acinclude.m4     doc/           snmplib/
root@darkstar:/var/src/squid-3.0.STABLE25# ./configure --prefix=/usr/local/squid
checking for a BSD-compatible install... /usr/bin/ginstall -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking how to create a ustar tar archive... gnutar
checking whether to enable maintainer-specific portions of Makefiles... no
checking for gcc... gcc
checking for C compiler default output file name... _
```

- PASSO 1: Extraíndo arquivos do pacote comprimido através do “**tar**”;
- PASSO 2: Preparando a instalação do pacote com o “**./configure**”



Exemplo:

Instalando através do código fonte

```
config.status: creating config.h
config.status: executing depfiles commands
root@darkstar:/var/src/squid-3.0.STABLE25#
root@darkstar:/var/src/squid-3.0.STABLE25# make ←
Making all in lib
make[1]: Entering directory `/var/src/squid-3.0.STABLE25/lib'
Making all in libTrie
make[2]: Entering directory `/var/src/squid-3.0.STABLE25/lib/libTrie'
make all-recursive
make[3]: Entering directory `/var/src/squid-3.0.STABLE25/lib/libTrie'
Making all in src
make[4]: Entering directory `/var/src/squid-3.0.STABLE25/lib/libTrie/src'
g++ -DHAVE_CONFIG_H -I. -I.. -I../include -Werror -Wall -Wpointer-arith -Wwrit
e-strings -Wcomments -g -O2 -MT Trie.o -MD -MP -MF .deps/Trie.Tpo -c -o Trie.o
```

- PASSO 3: Compilando o pacote com o “**make**”;

```
make[1]: Leaving directory `/var/src/squid-3.0.STABLE25/tools'
make[1]: Entering directory `/var/src/squid-3.0.STABLE25'
Build Successful.
make[1]: Leaving directory `/var/src/squid-3.0.STABLE25'
root@darkstar:/var/src/squid-3.0.STABLE25#
root@darkstar:/var/src/squid-3.0.STABLE25# make install ←
Making install in lib
make[1]: Entering directory `/var/src/squid-3.0.STABLE25/lib'
Making all in libTrie
make[2]: Entering directory `/var/src/squid-3.0.STABLE25/lib/libTrie'
make all-recursive
make[3]: Entering directory `/var/src/squid-3.0.STABLE25/lib/libTrie'
Making all in src
make[4]: Entering directory `/var/src/squid-3.0.STABLE25/lib/libTrie/src'
```

- PASSO 4: Realizando a instalação com o comando “**make install**”



Exemplo: Removendo a instalação realizada através do código fonte

- Ao realizar a instalação (comando “**make install**”), informações sobre a instalação (scripts, libs e outros arquivos gerados durante a compilação) são armazenadas no diretório em que o pacote foi descompactado.
 - Através do comando “**make uninstall**” podemos remover todos os arquivos copiados durante a instalação (realizada pelo comando “**make install**”):

```
root@darkstar:/var/src/squid-3.0.STABLE25# make uninstall ←
Making uninstall in lib
make[1]: Entering directory `/var/src/squid-3.0.STABLE25/lib'
Making uninstall in libTrie
make[2]: Entering directory `/var/src/squid-3.0.STABLE25/lib/libTrie'
```

- Através do comando “**make clean**” podemos remover os arquivos gerados durante a compilação (realizada pelo comando “**make**”):

```
root@darkstar:/var/src/squid-3.0.STABLE25# make clean ←
Making clean in tools
make[1]: Entering directory `/var/src/squid-3.0.STABLE25/tools'
Making clean in .
make[2]: Entering directory `/var/src/squid-3.0.STABLE25/tools'
rm -f squidclient squidclient
rm -f cachemgr.cgi cachemgr.cgi
```

Instalação através de pacotes pré-compilados (“.rpm”, “.deb”...)

Tópico 12: Gerenciamento de Pacotes.

Instalação através de Pacotes pré-compilados



- Para facilitar a administração dos pacotes em um sistema, alguns comandos foram desenvolvidos em algumas distribuições com o objetivo de instalar, remover, atualizar, ou listar os pacotes instalados.
- Porém, temos softwares distintos entre as principais distribuições com esta finalidade.
- Neste slide vamos abordar sobre os pacotes com extensão “DEB”, “RPM” e “TGZ”.
 - OBS.: Nos últimos slides estão disponíveis tabelas de referência com os principais comandos utilizados no gerenciamento de pacotes.

Instalação através de Pacotes pré-compilados



Pacotes RPM

- Red Hat e derivados:
 - CentOS;
 - Fedora;
- OpenSUSE;
- **Comando para instalar:**
 - `rpm -ivh pacote.rpm`

Pacotes DEB

- Debian e derivados:
 - Ubuntu;
 - Kubuntu;
- **Comando para instalar:**
 - `dpkg -i pacote.deb`

Pacotes TGZ

- Slackware;
- **Comando para instalar:**
 - `installpkg pacote.tgz`



Exemplo:

Instalando um pacote pré-compilado

- Para realizarmos a instalação de um pacote pré compilado, devemos primeiramente baixar o pacote da internet ou verificar se o CD de instalação da distribuição possui o pacote desejado.
 - **OBS.: Na maioria das vezes o pacote encontra-se disponível no CD.**
 - O DVD de instalação do **CentOS** por exemplo, possui aproximadamente 4.000 pacotes pré compilados (formato **“.RPM”**).
 - O **“DEBIAN 6”** é constituído de 8 DVDs, porém, apenas o primeiro é necessário para a instalação do sistema e os demais DVDs possuem diversos pacotes pré compilados para a distribuição (formato **“.DEB”**).
- Ao instalar um **pacote pré-compilado**, podemos nos deparar com dependências (pacotes que são pré requisitos para a instalação do pacote desejado), sendo necessário instalar as dependências antes de prosseguir com a instalação do pacote desejado.



Exemplo:

Instalando um pacote pré-compilado

- A seguir temos a instalação de um pacote pré compilado (a mesma sintaxe pode ser utilizada no **Red Hat** e derivados, como o **Fedora**);
 - OBS.: Apesar do **OpenSUSE** não ser derivado do “**Red Hat**”, o comando “**rpm**” é nativo da distribuição;

```
[root@localhost CentOS1# rpm -ivh mysql-5.0.77-3.el5.i386.rpm
aviso: mysql-5.0.77-3.el5.i386.rpm: Cabeçalho V3 assinatura DSA: NOKEY, key ID e
8562897
Preparando... ##### [100%]
 1:mysql ##### [100%]
```

- Na imagem a seguir, temos a instalação do pacote “**nmap**”, disponível no DVD da distribuição (que no caso é a “**CentOS**”, porém, o “**nmap**” também está disponível no DVD de instalação do **DEBIAN**).

```
[root@server ~]# rpm -ivh /mnt/cdrom/CentOS/nmap-4.11-1.1.i386.rpm
Preparing... ##### [100%]
 1:nmap ##### [100%]
[root@server ~]#
```



Exemplo: Removendo um pacote através do comando utilizado na instalação de **pré-compilados**

- Através dos comandos “**dpkg**” e “**rpm**”, utilizados para instalar um pacote pré compilado, também podemos remover pacotes instalados a partir de pacotes pré compilados (instalados com o “**dpkg**” ou “**rpm**”), bem como, pacotes nativos (instalados juntamente com o sistema).
- A seguir temos a remoção do pacote “nmap” instalado no slide anterior:

```
[root@server ~]# rpm -e nmap
[root@server ~]# nmap
-bash: nmap: command not found
```

Instalação através de Gerenciadores de pacotes

Tópico 12: Gerenciamento de Pacotes.

Instalação através de Gerenciadores de Pacotes



- Com o intuito de facilitar ainda mais a instalação de pacotes no sistema Linux, os **gerenciadores de pacotes** resolvem automaticamente as dependências durante a instalação de um determinado pacote.
- Ao utilizar um **gerenciador de pacotes**, temos um arquivo contendo a **lista de repositórios** de pacotes válidos para a sua distribuição;
 - A imagem a seguir ilustra o funcionamento de um gerenciador de pacotes:



Fonte: Documentação OpenSUSE – Gerenciamento de Pacotes. http://pt.opensuse.org/Gerenciamento_de_pacotes



Instalação através de Gerenciadores de Pacotes

- Um fator interessante é que os “Gerenciadores de Pacotes” buscam por **pacotes pré compilados nos repositórios** configurados, ou seja, a grande **vantagem** dos gerenciadores de pacotes é a **resolução automática de dependências**, ao invés de **retornar um erro** informando as **dependências** que devemos instalar previamente, como na instalação realizada a partir de pacotes pré compilados, explicado anteriormente.
- Alguns **gerenciadores de pacotes** permitem a realização de tarefas distintas, como se conectar automaticamente a um repositório, baixar um pacote, identificar e resolver dependências, pesquisar ou listar pacotes disponíveis nos repositórios configurados, adicionar/remover repositórios, verificar a integridade de um pacote através do “checksum” e assinatura digital, fazer atualizações automáticas e remover dependências ao desinstalar programas. Porém, nem todos os gerenciadores realizam todas estas tarefas.



Instalação através de Gerenciadores de Pacotes

- Dentre os principais gerenciadores de pacotes, temos:
 - **APT** (Advanced Packaging Tool) → **Debian e derivados**;
 - **YUM** (Yellow Dog Updater Modified) → **Red Hat e derivados**;
 - **YaST** (Yet another Setup Tool) → **OpenSUSE**;
 - OBS.: Cada um destes gerenciadores de pacotes possui um **arquivo de configuração** com uma listagem dos **repositórios válidos de pacotes** para a respectiva distribuição. Segue abaixo tabela para consulta rápida:

Arquivo de configuração dos repositórios - Gerenciadores de Pacotes	
DEBIAN e derivados	/etc/apt/sources.list
RED HAT	/etc/yum.repos.d/redhat-*.repo
CentOS	/etc/yum.repos.d/centos-*.repo
OpenSUSE	Através da Interface do YAST ou via CLI - Vide Link nas referências

- OBS.: O Slackware não possui um gerenciador de pacotes que resolva dependências até o momento de elaboração deste slide.
 - Vide guia de comandos em: <http://distrowatch.com/weekly.php?issue=20081013#feature>



Gerenciadores de Pacotes

Arquivos de Configuração

- Antes de configurar os repositórios de um “**gerenciador de pacotes**”, devemos definir quais serão os repositórios de pacotes para a respectiva distribuição (Internet ou DVD da distribuição).
- Este arquivo de configuração possui conteúdo distinto de acordo com o gerenciador/distribuição utilizada.
- Portanto, não vamos abordar maiores detalhes sobre o conteúdo destes arquivos, tendo em vista que ao instalar a distribuição o arquivo já vem previamente configurado com os repositórios oficiais da distribuição (seja o “**Red Hat**”, “**Debian**”, “**OpenSUSE**”, “**CentOS**”, entre outras);
 - OBS.: Como os repositórios oficiais estão disponíveis na Internet, o sistema GNU/Linux precisa ter acesso a Internet para possibilitar o download dos pacotes disponíveis nestes repositórios.



Exemplo:

Instalando através de um gerenciador

- Instalando o “**nmap**” através do gerenciador “**apt**” no “**Debian 6**”.

```
root@debian:~# apt-get install nmap
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  nmap
0 upgraded, 1 newly installed, 0 to remove and 33 not upgraded.
Need to get 0 B/1,584 kB of archives.
After this operation, 6,312 kB of additional disk space will be used.
Selecting previously deselected package nmap.
(Reading database ... 120286 files and directories currently installed.)
Unpacking nmap (from .../n/nmap/nmap_5.00-3_i386.deb) ...
Processing triggers for man-db ...
Setting up nmap (5.00-3) ...
root@debian:~#
root@debian:~# nmap
Nmap 5.00 ( http://nmap.org )
```



Exemplo:

Instalando através de um gerenciador

- Instalando o “nmap” através do gerenciador “yum” no “CentOS”.

```
[root@server ~]# yum install nmap
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: ftp.ussg.iu.edu
 * extras: centos-mirror.hostdime.com.br
 * updates: centos-mirror.hostdime.com.br
ftp://ftp.ussg.iu.edu/linux/centos/5.8/os/i386/repodata/repomd.xml: [Errno 4] IO
Error: [Errno ftp error] (111, 'Connection refused')
Trying other mirror.
base                                     | 1.1 kB      00:00
extras                                  | 2.1 kB      00:00
extras/primary_db                       | 171 kB      00:00
updates                                 | 1.9 kB      00:00
updates/primary_db                      | 366 kB      00:02
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package nmap.i386 2:4.11-2 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch      Version      Repository      Size
=====
```



Exemplo: Instalando através de um gerenciador

- Continuação....:

```
=====
Installing:
  nmap                i386                2:4.11-2                base                660 k
Transaction Summary
=====
Install          1 Package(s)
Upgrade          0 Package(s)

Total download size: 660 k
Is this ok [y/N]: y
Downloading Packages:
nmap-4.11-2.i386.rpm                | 660 kB      00:01
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing      : nmap                1/1

Installed:
  nmap.i386 2:4.11-2

Complete!
[root@server ~]# _
```



Exemplo:

Removendo através de um gerenciador

- Da mesma forma que o gerenciador de pacotes resolve as dependências para realizar a instalação de um pacote, as dependências são removidas ao remover um pacote.
- No slide a seguir temos como exemplo a remoção do pacote “sendmail” (Servidor de e-mail padrão da distribuição **CentOS**, instalado juntamente com o sistema).
- Observe que ao solicitar a remoção do pacote “sendmail”, mais 5 dependências são removidas automaticamente, após a confirmação do usuário.



Exemplo:

Removendo através de um gerenciador

- Removendo o pacote “sendmail” e suas dependências através do “yum”

```
[root@server ~]# yum erase sendmail
Loaded plugins: fastestmirror
Setting up Remove Process

Is this ok [y/N]: y
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Erasing      : sendmail                1/5
  Erasing      : redhat-lsb              2/5
  Erasing      : fetchmail               3/5
  Erasing      : mdadm                   4/5
  Erasing      : mutt                    5/5

Removed:
  sendmail.i386 0:8.13.8-2.el5

Dependency Removed:
  fetchmail.i386 0:6.3.6-1.1.el5      mdadm.i386 0:2.6.9-2.el5
  mutt.i386 5:1.4.2.2-3.0.2.el5      redhat-lsb.i386 0:3.1-12.3.EL.el5.centos

Complete!
```




Se liga aí! Hora da **REVISÃO**!





Tabela referência – Instalando Pacotes

Instalando através do código fonte			
DEBIAN	RED HAT	OpenSUSE	Slackware
tar xzf pacote-X.tar.gz cd pacote-X ./configure --options make make install	tar xzf pacote-X.tar.gz cd pacote-X ./configure --options make make install	tar xzf pacote-X.tar.gz cd pacote-X ./configure --options make make install	tar xzf pacote-X.tar.gz cd pacote-X ./configure --options make make install
Instalando através de pacotes pré compilados			
DEBIAN	RED HAT	OpenSUSE	Slackware
dpkg -i pacote.deb	rpm -ivh pacote.rpm	rpm -ivh pacote.rpm	installpkg pacote.tgz
Instalando através de Gerenciadores de Pacotes (utilitário que "resolve" as dependências)			
DEBIAN	RED HAT	OpenSUSE	Slackware
apt-get install pacote	yum install pacote	yast -i pacote	N/D

Tabela referência – Removendo e listando pacotes instalados



Removendo pacotes instalados através do código fonte			
DEBIAN	RED HAT	OpenSUSE	Slackware
cd pacote-X make uninstall make clean	cd pacote-X make uninstall make clean	cd pacote-X make uninstall make clean	cd pacote-X make uninstall make clean
Removendo pacotes pré compilados			
DEBIAN	RED HAT	OpenSUSE	Slackware
dpkg -r pacote	rpm -e pacote	rpm -e pacote	removepkg pacote
Removendo através de Gerenciadores de Pacotes			
DEBIAN	RED HAT	OpenSUSE	Slackware
apt-get remove pacote	yum erase pacote	yast --remove pacote	N/D
Listando Pacotes Instalados			
DEBIAN	RED HAT	OpenSUSE	Slackware
dpkg -l	rpm -qa	yast	pkgtool

Tabela referência

Gerenciadores de Pacotes



Tarefa A Ser Executada	Yum / Rpm CentOS / Fedora	Apt / Dpkg Debian / Ubuntu
Gerenciando o software		
Instalar software usando repositórios	<code>yum install pacote</code>	<code>apt-get install pacote</code>
Instalar software usando arquivo de pacote	<code>yum localinstall pacote.rpm</code> <code>rpm -ivh pacote.rpm</code>	<code>dpkg -i pacote.deb</code>
Atualizar um software	<code>yum update pacote</code> <code>rpm -Uvh pacote.rpm</code>	<code>apt-get install pacote</code>
Remover um software	<code>yum erase pacote</code>	<code>apt-get remove pacote</code>
Atualizando o sistema		
Atualizar lista de pacotes	<code>yum check-update</code>	<code>apt-get update</code>
Atualizar o sistema	<code>yum update</code>	<code>apt-get upgrade</code>

Fonte: <http://www.opcaolinux.com.br/gnulinux/dicas/14-comandos-linux/68-gerenciadores-de-pacotes.html>

Tabela referência

Gerenciadores de Pacotes



Tarefa A Ser Executada	Yum / Rpm CentOS / Fedora	Apt / Dpkg Debian / Ubuntu
Procurando por pacotes		
Procurar pelo nome do pacote	yum list pacote	apt-cache search pacote
Procurar por padrão	yum search padrão	apt-cache search padrão
Procurar pelo nome do arquivo	yum provides arquivo	apt-file search caminho
Listar todos os pacotes instalados	rpm -qa	dpkg -l
Configurando o acesso a repositórios de software		
Listar repositórios	yum repolist	cat /etc/apt/sources.list
Adicionar repositório	(adicionar à /etc/yum.repos.d/)	(editar /etc/apt/sources.list)
Remover repositório	(remover de /etc/yum.repos.d/)	(editar /etc/apt/sources.list)

Fonte: <http://www.opcaolinux.com.br/gnulinux/dicas/14-comandos-linux/68-gerenciadores-de-pacotes.html>



No próximo slide...

- Tópico 13: Agendamento de tarefas (cron) e Backup.
- FINISH HIM!!!



Referências

- BONAN, Adilson Rodrigues. **LINUX – Fundamentos, Prática & Certificação LPI**. Editora: Alta Books. RJ. 2010;
- PEREIRA, Guilherme Rodrigues. **Slides para aula expositiva**. Centro Universitário UNA.
- SILVA, Gleydson Mazioli. **Guia Foca GNU/Linux**. Disponível em: <https://guiafoca.org/>
- SHIELDS, Ian – IBM – Aprenda Linux, 101: Gerenciamento de pacote RPM e YUM. Disponível em: <http://www.ibm.com/developerworks/br/linux/library/l-lpic1-v3-102-5/index.html>



Referências

- FREY, John – Gerenciamento de Pacotes (Traduzido por: Roberto Bechtluft). Disponível em: <http://www.hardware.com.br/dicas/gerenciamento-pacotes.html>
- Documentação OpenSUSE – Gerenciamento de pacotes. Disponível em: http://pt.opensuse.org/Gerenciamento_de_pacotes
- Documentação OpenSUSE – Adicionar Repositórios de Pacotes. Disponível em: http://pt.opensuse.org/SDB:Adicionar_reposit%C3%B3rios_de_pacotes#Reposit.C3.B3rios_de_softw
[are_do_YaST](http://pt.opensuse.org/SDB:Adicionar_reposit%C3%B3rios_de_pacotes#Reposit.C3.B3rios_de_softw)
- Documentação OpenSUSE – Repositórios de Pacotes. Disponível em: http://pt.opensuse.org/Reposit%C3%B3rios_de_pacotes



DGP

Tecnologia da Informação

Obrigado!



Guilherme Rodrigues