Instalação e gerenciamentos de pacotes
Assuntos abordados

- Métodos de instalação de softwares prontos (compilados).
- Ferramentas para administração e manipulação de pacotes.
- Ferramentas para atualização e operações remotas
- Configuração e versionamento
- Criação de pacote
- Conclusão

Instalação de software

- Através de ferramentas de gerenciamentos de pacotes.
- Pacotes contendo binários prontos (i.e.: Já compilados)
- Compilados em arquiteturas semelhantes.

Pacotes binários

- pacote.versao.gz

| /var/lib/pacote.o

| /etc/soft.conf

| /bin/software
| /usr/bin/soft

| /usr/share/man/man1/soft.1.gz

Variando de distribuição para distribuição

– RPM

– DPKG

– Ferramentas auxiliares para instalação online: APT, APTITUDE, YUM e ZIP(YAST).

RPM

– Ferramenta mais popular de administração de pacotes de softwares.

- Utiliza banco de dados para guardar informações pertinentes aos pacotes.

– Através de comandos contras os arquivos de pacotes e contra a base de dados é possível se realizar todas operações de manutenção e administração

RPM
Consultas e arquivos

/var/lib/rpm diretório da base de dados dos RPM

rpmdb (série de ferramentas para manipular o banco de dados de rpm)

rpm –i (instala)

rpm –u (upgrade)

rpm –e (remove)

rpm –qa (exibe pacotes instalados)

rpm –qi (informações sobre um pacote instalado)

rpm –ql (lista os arquivos de um pacote instalado)

rpm –qf (indica a qual pacote instalado é proveniente um arquivo)

rpm –qpi (exibe informações de um pacote não instalado)

rpm –qpl (exibe arquivos que fazem parte de um pacote não instalado)

DPKG

Modelo de gerenciador de pacotes aos moldes do RPM, no entanto, para o Debian. Utiliza uma ferramenta de frontend ---- dselect

Principais arquivos de controle:

- /etc/dpkg/dpkg.cfg (arquivo de log e validação de fontes)

- /var/log/dpkg.log (arquivo de log das transações dpkg)

- /var/lib/dpkg/
  - status
  - available

- /var/lib/dpkg/info/ (info dos pacotes instalados)

Comandos DPKG

DPKG-DEB

#> dpkg -i <arquivo_pacote.deb>

#> dkpg -l <arquivo_pacote.deb>

#> dpkg -c <arquivo_pacote.deb>

DPKG-QUERY

#> dpkg –configure <pacote>

#> dpkg -r <pacote>

#> dpkg –purge <pacote>

#>dpkg -L <pacote>

#>dpkg -s <pacote>

#> dpkg -S arquivo_sistema

DPKG -i

- extrai o info do pacote
- se houver pacote de mesmo nome executa o prerm do antigo
- executa preinst
- descompacta os novos arquivos mantendo backup dos antigos caso necessário
- postrm de arquivos antigos
- configure

DPKG -r

- prerm
- remove arquivos
- postrm:

Informações do pacote

status

not-installed

config-files

half-instaled

unpacked

half-configured

triggers-awaiting

triggers-pending

installed

Informações do pacote

selection state

install

hold

deinstall

purge

Manipulando um pacote deb

→ Link para um arquivo deb

http://ftp.us.debian.org/debian/pool/main/s/sl/sl_3.03-16_i386.deb

Baixar o arquivo e manipulá-lo localmente

Atividade
Com o arquivo .deb faça:

- Crie um arquivo texto que deve conter o caminho completo de todos arquivos pertencentes ao pacote ora obtido
- Descubra se esse pacote já está instalado em seu sistema. É possível garantir com exatidão se tal software está ou não instalado?
- Descubra qual a dependência desse pacote e quem é seu autor
- Por fim, instale e verifique o estado desse pacote

Ferramentas para automatizar DPKG

Como, então, melhorar ou deixar o DPKG mais ágil?

– APT é utilizado para buscar repositório, através de método pré definidos, de forma a automatizar a instalação e conclusão de dependências de alguns pacotes.

- Executa instalação, atualização, remoção de pacotes, etc; mantendo a validade do banco de dados do DPKG.

- Define arquivos de fontes para base de repositórios disponíveis.

APT

Métodos e principais comandos

#> apt-cache search blabla

#>apt-get install blabla

#>apt-get update

#> apt-get upgrade

#>apt-get dist-upgrade

Atividade

- Desligue o cabo de rede de sua estação e explique o que vai acontecer em cada uma das execuções dos comandos abaixo. Por quê?

#> apt-get update

#> apt-cache search sl

APT – Arquivos e diretórios

- /var/lib/apt (diretório base do apt)

- /var/lib/apt/lists (entradas de repositório com as assinaturas e arquivos obtidos)

-/etc/apt/preferences.d/ (configuração personalizada do funcionamento do apt)

-/etc/apt/sources.list (arquivo de definição das fontes de obtenção de pacotes)

Estrutura do arquivo /etc/apt/sources.list

tipo_pacote método://URI distro área/seção1seção2...

Tipo pacote: deb ou deb-src

Método: ftp, http, cdrom, file, copy e rsh

URI: diretório local ou remoto

Distro: stable, testing ou unstable

Componente: main, contrib e non-free

Seção: games, office, multimedia, etc

Estrutura do arquivo /etc/apt/sources.list

sid

wheezy

squeeze

Exemplo de sources.list

deb http://br.archive.ubuntu.com/ubuntu/ karmic-updates main

deb-src http://br.archive.ubuntu.com/ubuntu/ karmic-updates main

Atividade
Alterando configurações apt

- Qual a versão base utilizado em seu sistema pelo APT?

- Altere o conteúdo de sources.lists para conter um repositório de fonte instável

Criando seu próprio pacote .deb

Utilizar o aplicativo dpkg-deb para contrução e edição de pacotes deb

#> mkdir /tmp/pacote

#> mkdir /tmp/pacote/DEBIAN

#> touch /tmp/pacote/DEBIAN/control

Package: pacote
Priority: optional
Version: 0.1
Architecture: i386
Maintainer: Rodrigo Zuolo Carvalho
Depends:
Description: Este é um pacote teste.

#> cd /tmp/pacote

#> mkidr -p usr/bin

#>cp ~zuolo/pacote /tmp/pacote/usr/bin/

#>dpkg-deb -b /tmp/pacote /tmp/

Conclusão

- Facilidade de manter o controle de software
- Centralização na base de dados
- Internet facilita o uso do apt
- Problema de confiabilidade e origem dos pacotes
- Frequência de atualização

FIM

#> apt-get moo

#> aptitutde -v moo
#> aptitutde -vv moo
#> aptitutde -vvv moo

.

.

.