

Internet e Protocolos

Prof. Jean Carlos Lourenço Costa

Roteiro da aula

- Shell
- Bash
- Variável
- Man
- Uname
- Gerenciadores de Pacotes
- Instalação de Programas

Shell

- O ambiente em modo texto em distribuições GNU/Linux é conhecido com Shell, e é o local onde os comandos digitados pelos usuários são interpretados. É possível configurar esse ambiente automatizando pequenas tarefas do dia a dia, e ainda fazendo de comandos internos que o próprio ambiente nos trás.
- *Mas afinal o que é um shell?*

Shell

- O shell é um programa que permite ao usuário interagir com o sistema operacional através de comandos digitados no teclado. No MS-DOS o shell era o command.com, que permitia executar alguns comandos como cd, dir, etc.
- O shell mais conhecido no mundo GNU/Linux é o bash, o padrão para novos usuários quando são criados no sistema.

Bash

- O bash traz muitas funcionalidades como comandos internos, histórico de comandos, autocompletar, variáveis de ambiente, etc. Para você exibir quais comandos são internos use help como no exemplo abaixo:

```
$ help
```

Histórico de comandos

- Uma das funções mais uteis no dia a dia é trabalhar com comandos do histórico. Você pode acessar esses comandos usando as teclas de navegação para cima e para baixo, ou através do comando history. Vamos a prática:

```
$ history
```

```
1  ls
2  date
3  cal
4  clear
5  history
```

History

- A lista é uma ordem de comandos que estão guardados no histórico. Para executar um comando da lista use exclamação + numero do comando. Exemplo:

```
$ !4
```

- Para limpar a lista de comando use o comando history com a flag -c:

```
$ history -c
```

Variável

- Variáveis
- No GNU/Linux as variáveis são muito usadas na criação de shell scripts, mas também podem ser declarada diretamente no terminal, e assim gerenciadas pelo shell.

Mas o que é variável?

Variável

- Variável pode ser definida como um objeto, ou uma posição localizada na memória, que guarda um valor ou expressão. Vamos ver na prática como declarar uma variável.

```
$ a=10
```

- Veja que em nosso exemplo a variável “a” foi declarada com o valor 10. Para exibir o conteúdo da variável use o comando “echo”, cifrão e o nome da variável.

```
$ echo $a
```

Variável

- É possível somar o conteúdo de 2 ou mais variáveis usando alguns comandos, vamos a prática:
- Declare um conteúdo para 2 variáveis:

```
$ b=20
```

```
$ c=30
```

- Para somar use o comando expr ou o próprio echo. Exemplo:

```
$ expr $b + $c
```

Variável

```
$ echo $(($b+$c))
```

- Variáveis locais x globais
- Quando você declara uma variável ela pode ser considerada pelo sistema como local ou global. A diferença está na maneira de declarar a variável.
- Exemplo:
 - Variável local

```
$ curso=ETECDS
```

Variável

- Variável global

```
$ export curso=ETECDS
```

- Veja que o comando export foi usado antes de declarar a variável.
Para listar os tipos de variáveis use os comandos env e set:
- Para exibir variáveis locais use o comando set:

```
$ set
```

- Para exibir variáveis globais use o comando env

```
$ env
```

Variável

- Para deletar uma variável da memória use o comando unset:

```
$ unset curso
```

- O shell guarda informações do ambiente dentro de algumas variáveis, chamadas de variáveis de ambiente. Veja a descrição abaixo:

Variável

HOME - Exibe o diretório do usuário logado;

SHELL - Exibe qual shell está sendo usado;

TERM - Exibe o tipo de terminal que está sendo usado;

USER - Exibe o nome do usuário logado;



Não esqueça que os nome das variáveis de ambiente são apresentadas em maiúsculas!

uname

- O comando `uname` exibe informações do sistema como a versão do kernel, processador, sistema operacional, entre outros. Veja a descrição das opções usadas com o comando `uname`:

`-i` - Tipo de processador.

`-m` - Arquitetura da máquina.

`-n` - Nome da máquina na rede.

`-p` - Processador.

`-o` - Sistema operacional.

`-r` - Versão do código fonte do kernel.

`-s` - Nome do kernel.

`-v` - Versão de compilação do kernel.

man

- Exibe o manual de um comando. Exemplo de como trazer o manual do comando ls

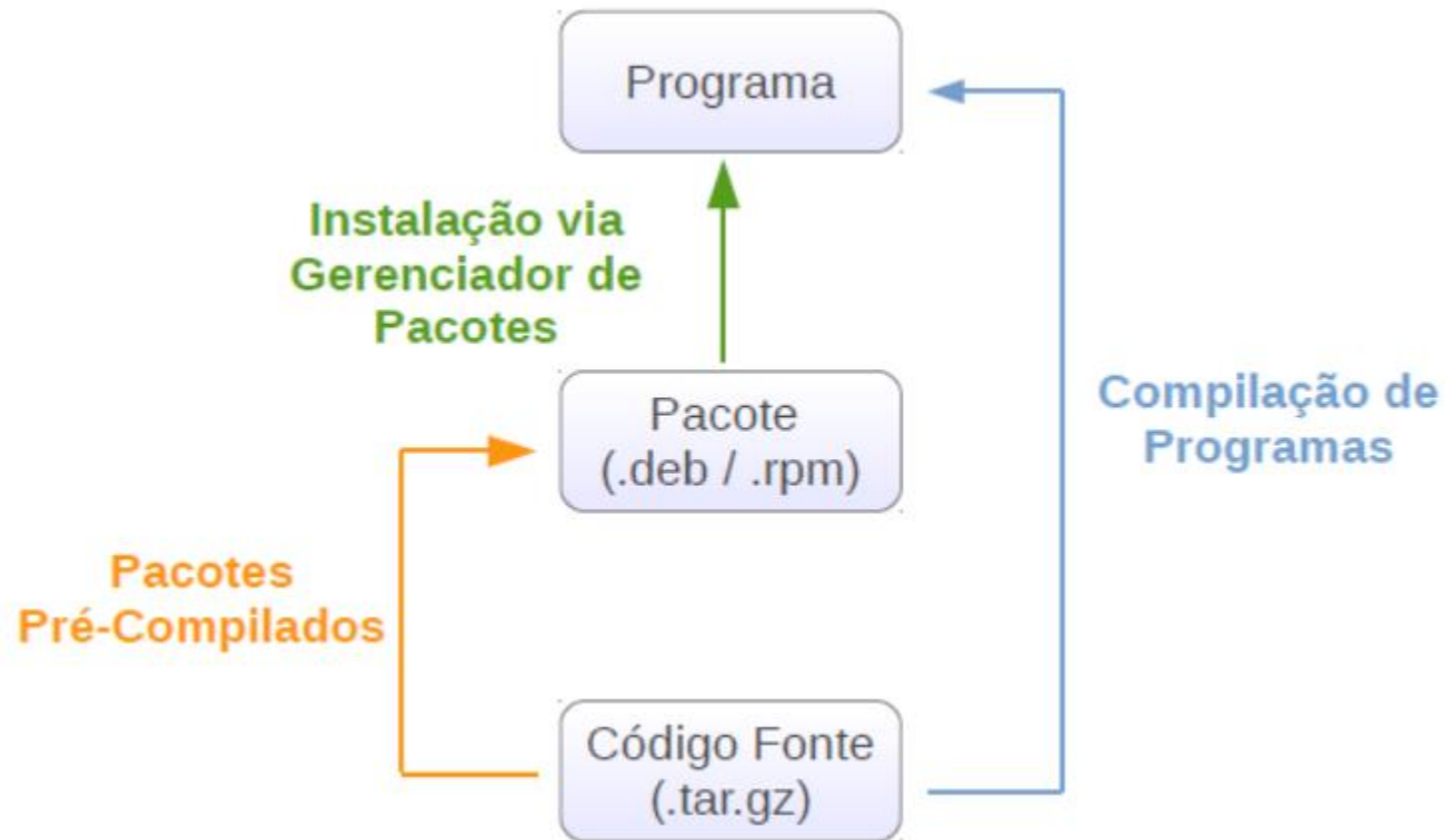
```
$ man ls
```


Instalação de Programa



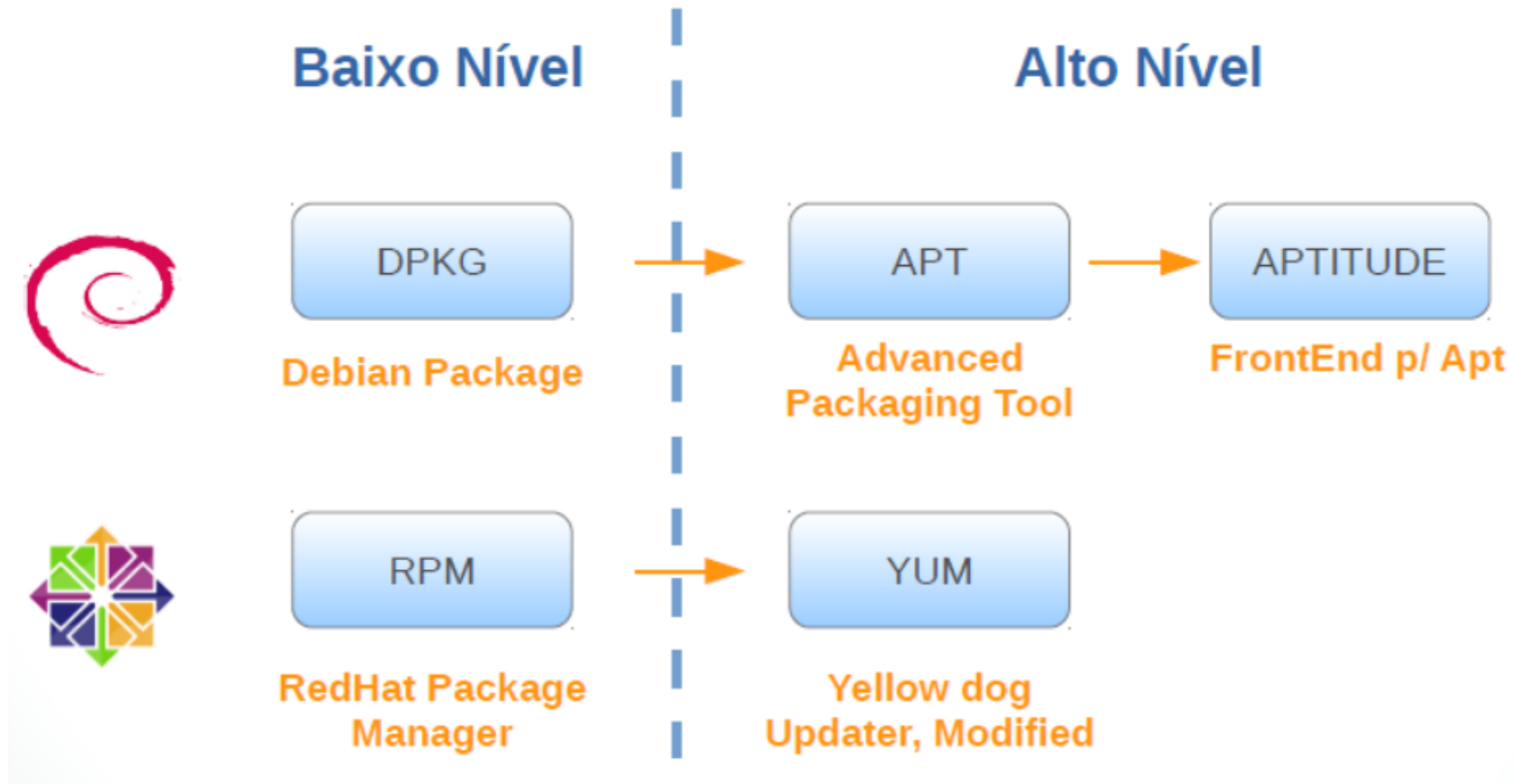
Instalação de Programas

- Métodos de Instalação de Programas do GNU/Linux:



Instalação de Programa

- Gerenciadores de Pacotes Baixo e Alto Nível:



Gerenciadores de Pacotes Baixo e Alto Nível:

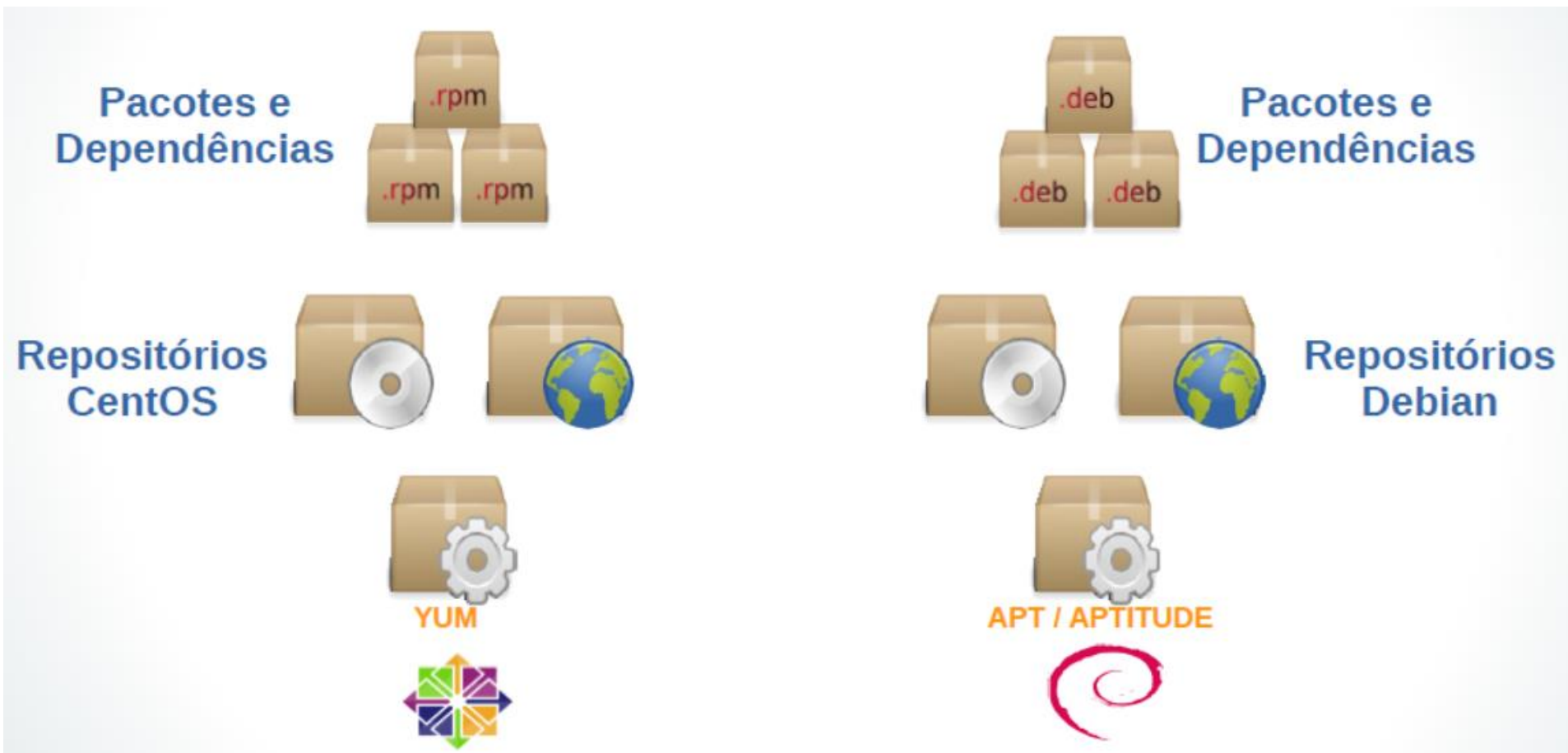
Baixo Nível

- 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.

Gerenciadores de Pacotes de Alto Nível



Gerenciadores de Pacotes de Alto Nível

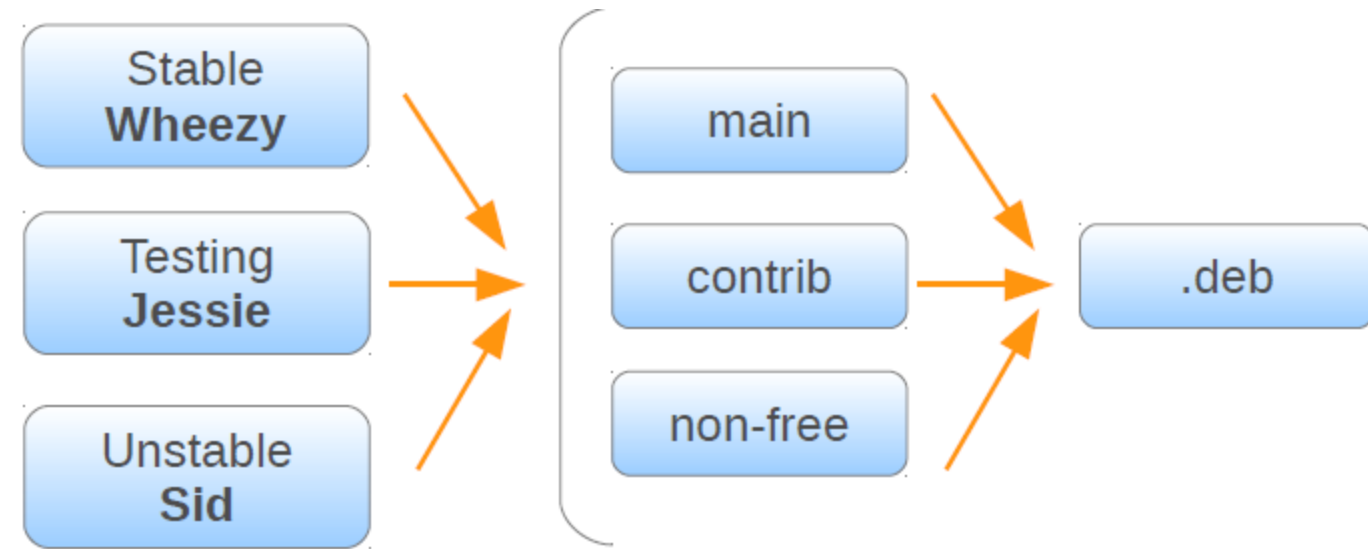
- Diferente do DPKG e RPM, o APT e o YUM trazem agilidade e facilidade no gerenciamento de programas do Linux;
- Ambos necessitam da configuração dos repositórios para que possam automaticamente buscar o pacote e suas dependências, quando acionamos uma instalação de programa;
- A principal função do APT e YUM é realizar a pré-instalação, ou seja, realizar o download dos pacotes e suas dependências e passar as instruções para o DPKG e o RPM realizarem a instalação. Eles auxiliam o DPKG e RPM.

Gerenciador de Pacotes APT

- O ciclo de desenvolvimento do Debian passa por três fases:
- **Unstable** → Últimas versões de programas, porém não homologado;
- **Testing** → Versões recentes e sendo homologados para a próxima versão do Debian;
- **Stable** → Não necessariamente as últimas versões de programas, porém testado e homologado pela Comunidade Debian, ou seja, programas estáveis.

Gerenciador de Pacotes APT

- **wheezy** é a distribuição. Neste campo opções possíveis são stable, testing e unstable.
- **main contrib e non-free** são as secções do repositório.



Gerenciador de Pacotes APT

- Configuração do Repositório Debian/Ubuntu:

```
1# vim /etc/apt/sources.list
```

Gerenciador de Pacotes APT

➤ 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# cat /var/log/auth.log | ccze | more
7# apt-get remove ccze
8# apt-get remove --purge 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 pacotes e suas dependências;

remove --purge → Remove completamente o pacote incluindo arquivos de configuração.

Gerenciador de Pacotes APT

➤ Funcionalidades Avançadas:

```
1# ls /var/cache/apt/archives
2# apt-get clean
3# apt-get dist-upgrade
4# apt-get autoremove
5# apt-get -f install
6# apt-get -f remove
7# apt-get -d install <pacote>
```

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.

Vamos para Prática

- Dúvidas??
- Instalar: vim, htop, ipcalc, net-tools e python3.13