

## Conteúdo: **Conceitos de Linux**

## CONTEXTUALIZAÇÃO

**Nesta aula teremos como **objetivo** aprender o seguinte:**

- Gerenciar bibliotecas compartilhadas;
- Gerenciar pacotes do Debian;
  - dpkg;
  - apt.
- Comandos Linux

## GERENCIAR BIBLIOTECAS COMPARTILHADAS

- Programas compilados podem ter funções vinculadas a bibliotecas do sistema;
- Ex: quando o **printf()** é usado em um programa, o programador não escreve o código do **printf()**, ele utiliza essa função que deve estar no sistema.
  - **Utiliza uma biblioteca do sistema.**

## GERENCIAR BIBLIOTECAS COMPARTILHADAS

- **Programas estaticamente vinculados:** copiam o código da biblioteca para dentro de si.
  - É um programa independente, pois não precisa de nenhum outro código no momento da execução;
  - **Ao copiar o código, deixa de utilizar a biblioteca compartilhada.**
- **Programas dinamicamente vinculados:** utilizam as bibliotecas, mas não incorporam o código.
  - A biblioteca é vinculada ao executável no momento da execução;
  - Utiliza a biblioteca de forma compartilhada.

## GERENCIAR BIBLIOTECAS COMPARTILHADAS


- Qualquer programa dinamicamente vinculado vai precisar de uma ou mais biblioteca;
  - O programa não executa sem elas.
- Se as bibliotecas não estiverem no sistema, elas precisam ser instaladas;
  - Por isso que muitas vezes quando estamos compilando um programa, obtemos um erro avisando as dependências faltantes.
- O utilitário **ldd** é usado para saber quais bibliotecas são necessárias/utilizadas por um executável qualquer.

## COMANDOS

### **ldd**

- **Sintaxe:**
  - **ldd** programas
    - O resultado indica o nome da biblioteca e o local onde ela se encontra.
- **Exemplo:**
  - **ldd** /bin/bash
    - libtinfo.so.5 => /lib/x86\_64-linux-gnu/libtinfo.so.5
    - libdl.so.2 => /lib/x86\_64-linux-gnu/libdl.so.2
    - libc.so.6 => /lib/x86\_64-linux-gnu/libc.so.6

## VINCULANDO BIBLIOTECAS COMPARTILHADAS

- Os **executáveis dinamicamente vinculados** são examinados, no momento da execução, pelo vinculador dinâmico de objetos compartilhados:
  - **ld.so** 
  - Ele procura pelas bibliotecas compartilhadas no executável e tenta encontrá-las no sistema.

## VINCULANDO BIBLIOTECAS COMPARTILHADAS

- Para informar ao **ld.so** novos lugares para procurar as bibliotecas (diferente dos locais padrão), pode-se usar uma das seguintes formas:
  - Alterando a variável de ambiente **LD\_LIBRARY\_PATH** (adicionar a lista de diretórios separados por vírgulas).
    - Ex: `export LD_PATH="home/teste, /root/teste"`
  - Altera o arquivo **/etc/ld.so.conf**
    - Incluir nele os diretórios onde estão as bibliotecas compartilhadas;
      - Ou onde esse arquivo indicar colocar
    - Se o arquivo não existir, basta criar .



## COMANDOS

- Comando **ldconfig**.
  - Sintaxe: **ldconfig** [opções] [diretórios\\_de\\_bibliotecas](#)
  - Atualiza o cache do ld.so com as bibliotecas compartilhadas especificadas na linha de comando em [diretórios\\_de\\_bibliotecas](#), nos diretórios confiáveis, (/usr/lib e /lib) e nos diretórios encontrados em /etc/ld.so.conf.

## COMANDOS

- Exemplo 1: **Idconfig -p**
  - Exibe o conteúdo do cache atual.
- Exemplo 2: **Idconfig -p | grep ncurses**
  - Procura por uma biblioteca específica no cache.
- Exemplo 3: **Idconfig**
  - Recria o cache.

## COMANDOS

### Exercitando junto com o professor

- Crie um arquivo (.conf) dentro da pasta /etc/ld.so.conf.d e nele inclua um novo local para as bibliotecas. Fazemos assim devido as instruções no arquivo /etc/ld.so.conf.
  - Ex: joilson.conf com o conteúdo /lib/joilson
- Crie a pasta /lib/joilson e dentro dela copie a biblioteca: /lib/joilson/libncurses.so.6.
- Execute o comando **ldconfig** (recria o cache).
- Execute novamente o comando **ldconfig -p | grep ncurses** e veja o resultado.

## GERENCIADOR DE PACOTES DO DEBIAN

- **Conjunto de ferramentas** usadas para **instalar** e **gerenciar pacotes** no Debian (são pacotes pré-compilados para o Debian);
- Cada **pacote do Debian** possui:
  - Arquivos de programas e de configuração;
  - Documentação;
  - Indicação de dependências em relação a outros pacotes.

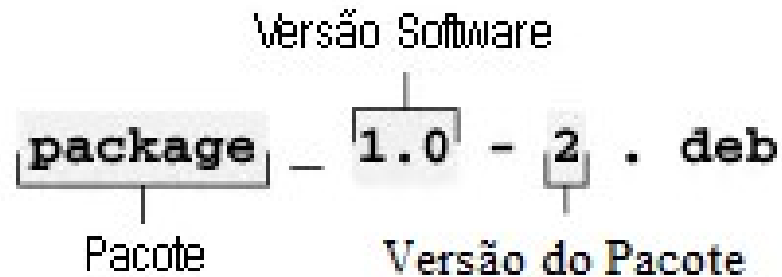
# GERENCIADOR DE PACOTES DO DEBIAN

- Os nomes dos pacotes do Debian possuem quatro partes, que são:
  - **Nome do pacote:** curto e descritivo, quando várias palavras são usadas, elas são separadas por hífens.
  - **Número da Versão do software:** cada pacote tem um número de versão próprio, que pode variar entre versão principal e correções aplicadas;
  - **Número da Versão do pacote:** os pacotes também podem ter versões diferentes de distribuição (pode representar a arquitetura para qual ele foi desenvolvido).
  - **Extensão .deb:** para fácil identificação dos pacotes Debian.

• Exemplo: Pacote-X-Y\_ versão\_do\_software – versão do pacote . deb

## GERENCIADOR DE PACOTES DO DEBIAN

- Exemplo:



- abiword-gtk**1.0.2+cvs.2002.06.05-1\_i386.deb

- Na máquina Linux, execute o comando **dpkg -l** veja alguns pacotes instalados na máquina.

## COMANDOS

- Uma ferramenta usada para gerenciar os pacotes é a **dpkg**
  - Opera sobre pacotes com a extensão **.deb**
  - É usada para automatizar a instalação e manutenção dos pacotes de softwares.
- **Sintaxe: dpkg [opção] ação**
  - **O dpkg mantém informações sobre os pacotes na pasta /var/lib/dpkg**

## GERENCIADOR DE PACOTES DO DEBIAN

- Há dois arquivos que são de particular interesse na pasta:
  - **available:** é a lista de todos os pacotes disponíveis.
  - **status:** contém atributos dos pacotes, tais como se um determinado pacote está instalado ou se está marcado para remoção.
- Na máquina Linux, edite esses dois arquivos em `/var/lib/dpkg` e observe as informações contidas neles.



# COMANDOS

## Opções mais frequentes usadas com o dpkg:

- **-i** package\_name
  - Instala o pacote contido em package\_name.
- **-l** package
  - Listas os arquivos com o nome package e similares.
- **--purge** package\_name
  - Remove o pacote contido em package\_file e todos seus arquivos.
- **--remove** package\_name
  - Remove os binários, mas não remove os arquivos de configurações
- **-s** package\_name
  - Opções sobre o status do pacote, tais como versões, dependência e etc.

## COMANDOS

### Exemplos para fazer junto com o professor

**Faça o download do pacote anacron com o wget:**

```
wget http://ftp.de.debian.org/debian/pool/main/a/anacron/anacron_2.3-30_amd64.deb
```

**Instale o pacote anacron com o dpkg**

```
dpkg -i anacron_2.3-23_i386.deb
```

**Liste o pacote anacron :**

```
dpkg -l |grep anacron
```

**Remova o pacote anacron :**

```
dpkg --purge anacron
```

.

## COMANDOS

### **Exemplos para fazer junto com o professor**

- O Problema é que muitos pacotes possuem dependência, ou seja, necessitam de outros pacotes para serem instalados;
- Por isso é muito trabalhoso fazer a gerência dos pacotes com o dpkg (instalar cada dependência de forma manual).

## APT-GET - ADVANCED PACKAGING TOOL

- Usando o dpkg, todas as dependências devem ser instaladas de forma manual. O APT resolve isso (ele instala as dependências de forma automática);
- O APT é um sistema que permite o gerenciamento de pacotes de software em distribuições Debian GNU/Linux e seus derivados de forma automática.
- Permitindo realizar tarefas como:
  - Instalação de pacotes;
  - Remoção de pacotes;
  - Atualização do sistema.
  - Instalação de dependências de forma automática (importante);

## APT-GET - ADVANCED PACKAGING TOOL

### Arquivo de configuração: /etc/apt/sources.list

- O APT usa um arquivo que lista as fontes de onde ele obterá os pacotes.
  - Esse arquivo é o **/etc/apt/sources.list**
    - Na máquina Linux, vamos olhar esse arquivo junto com o professor.
- As entradas desse arquivo são normalmente formadas assim:
  - **deb** http://host/ distribuição seção1 seção2 seção3
  - **deb-src** http://host/ distribuição seção1 seção2 seção3

## APT-GET - ADVANCED PACKAGING TOOL

- **deb e deb-src** indicam o tipo de repositório:
  - se ele armazena pacotes binários (**deb**), ou seja, pacotes pré-compilados prontos para instalação;
  - se guarda pacotes fonte (**deb-src**), que são o código-fonte original do programa.

## APT-GET - ADVANCED PACKAGING TOOL

Exemplo de entrada:

- deb <http://ftp.br.debian.org/debian/> bullseye main contrib non-free
  - **bullseye:** codinome das versões DEBINA
  - **MAIN:** Contém todos os pacotes que estão completamente de acordo com o Debian Free Software Guilines.
  - **CONTRIB:** É um conjunto de programas de código aberto que não podem funcionar sem um elemento não livre.
  - **NON-FREE:** Contém programas os quais não estão (completamente) de acordo com estes princípios do Software Livre, mas que podem, contudo, ser distribuídos sem restrições. Estes pacotes, o qual não é parte oficial do Debian, é um serviço para os usuários que podem precisar de alguns desses programas, entretanto o Debian sempre recomenda dar prioridade aos programas livres.

## APT-GET - ADVANCED PACKAGING TOOL

- Sempre que esse arquivo for modificado, deve-se executar o comando **apt-get update** logo após
- Isso deve ser feito para que o APT obtenha as listas de pacotes atualizadas das fontes especificadas.

→ **Execute o apt-get update na máquina Linux e observe o resultado**



# APT-GET - ADVANCED PACKAGING TOOL

## Acrescentando CD/DVD no arquivo sources.list

- É possível utilizar CDs ou DVDs como repositórios de pacotes do sistema.
- Para isso, deve-se inserir o cd ou dvd da distribuição no drive, e executar o comando a seguir:

**apt-cdrom add**

→ **Tente fazer, e veja que o Linux irá pedir que coloque o CD no drive.**

- Logo em seguida, atualize a lista de pacotes disponíveis:

**apt-get update**

- Agora o conteúdo do CD/DVD também está indexado no sistema, e seus pacotes podem ser instalados por meio do comando **apt-get install**.

# Opções Gerais do apt-get

### Sintaxe:

- **apt-get** [opções] [comando] [nome\_do\_pacote ....]

### Procurar pacotes

- **apt-cache search** nome-pacote
  - Junto com o professor procure o pacote tcpdump


### Instalar o(s) pacotes especificado(s):

- **apt-get install** nomes-pacotes
  - Junto com o professor instale o pacote tcpdump

## COMANDOS

- **Remover os pacotes especificados (desinstalar programas):**
  - **apt-get remove pacote**
- **Remove os pacotes especificados, incluindo seus arquivos de configuração:**
  - **apt-get purge pacote**
  - Junto com o professor remova o pacote tcpdump

## COMANDOS

- Atualizar as listas de pacotes dos repositórios (não instala os pacotes, apenas descobre quais estão disponíveis):
  - **apt-get update**
  - Junto com o professor atualize a lista de pacotes.
- Atualizar os pacotes instalados no sistema. Com a opção -u é possível visualizar os pacotes que serão atualizados.
  - **apt-get upgrade**
  - Junto com o professor atualize os pacotes instalados.
- Atualizar a distribuição para a versão mais recente:
  - **apt-get -u dist-upgrade**  Vai demorar alguns minutos
  - Junto com o professor atualize para a versão mais recente.
  - **IMPORTANTE:** antes faça uma cópia da máquina no VirtualBox (atualizações podem ter problemas em sistemas virtualizados, principalmente com a inicialização(GRUB)).

## COMANDOS

- Verificar a lista de dependências quebradas e tentar corrigi-las, instalando pacotes que sejam necessários:
  - **apt-get -f install**
  - Junto com o professor execute o comando.
- Excluir todos os pacotes que foram baixados e já instalados
  - **apt-get clean**
  - Junto com o professor execute o comando.
- Apagar todos os pacotes que não possam mais ser baixados
  - **apt-get autoclean**
  - Junto com o professor execute o comando.

## COMANDOS

- Remove pacotes órfãos (pacotes sem utilidade e dependências de softwares já removidos)
  - **apt-get autoremove**
  - Junto com o professor execute o comando
- Baixa arquivos apenas, porém sem instalá-los
  - **apt-get -d**
    - Ex: **apt-get -d install apache2**
    - Só vai fazer o download do apache2 (os arquivos ficam em: /var/cache/apt/archives/)
    - Junto com o professor execute o comando
- Baixa o código fonte dos pacotes via APT (baixa na pasta atual)
  - **apt-get source nome\_do\_pacote**
  - **Ex: apt-get source apache2**
  - Junto com o professor execute o comando

## EXERCÍCIOS

**Faça todos os exercícios junto com um relatório para ser enviado ao professor**

## EXERCÍCIOS

1. Verifique quais são as bibliotecas compartilhadas que os seguintes programas utilizam:
  - a) ls;
  - b) dir;
  - c) rmdir.



## EXERCÍCIOS

2. Dado o seguinte arquivo:

`abiword-gtk_1.0.2+cvs.2002.06.05-1_i386.deb`

Identifique qual parte representa o **nome do pacote**, a **versão do software** e a **versão do pacote**.

## EXERCÍCIOS

3. Descreva qual é a função dos arquivos **available** e **status** em **/var/lib/dpkg**;
4. Liste todos os pacotes instalados no sistema com o **dpkg**;
5. Liste todos os pacotes instalados com o **dpkg** e aplique um filtro para ver apenas o pacote **adduser**;
6. Instale novamente, com o **dpkg**, o pacote **anacron** removido anteriormente e comprove que ele está instalado;
7. Remova o **anacron** e comprove que ele foi removido.

## COMANDOS

8. Descreva qual é a função do arquivo `/etc/apt/sources.list`.
9. O que as entradas `deb` e `deb-src` que estão dentro deste arquivo indicam?
10. Sempre que o arquivo `“/etc/apt/sources.list”` for modificado, deve-se executar um comando, qual é esse comando.
11. Execute o comando descrito acima no terminal e veja o resultado.

## EXERCÍCIOS

12. Faça o download via **apt** do pacote **php** (só o download, não instale);
13. Instale o pacote **locate** via **apt** ( após execute o comando **updatedb** para atualizar a base do **locate**);
14. Encontre o pacote **php** (o **.deb**) com o **locate** (**locate nmap**);
15. Tente instalar o **php** via **dpkg**;
16. Verifique as dependências faltantes (provavelmente o pacote não funcionará por falta de dependências);
17. Instale todas as dependências e o **php** com o **apt**;
18. Desinstale o pacote **nmap** com o **dpkg**;
19. Remova o pacote **locate** e todos seus arquivos com o **apt**;
20. Exclua todos os pacotes que foram baixados e já instalados;
21. Baixe via apt, os código fontes do python dentro da pasta desenv, a qual deverá ser criada na pasta home do seu usuário.
22. Atualize todos os pacotes instalados no sistema com o **apt** (mas não atualize a distribuição do sistema).

# PROCESSAR STREAMS DE TEXTO USANDO FILTROS

## Comando CUT

- Corta (ou seja, exhibe) colunas ou campos selecionados a partir de um ou mais arquivos. O arquivo fonte não é modificado.
- **Sintaxe**
  - **cut** opções [arquivos]
- **Seleção de uma coluna**
  - Para selecionar uma coluna de um arquivo, usamos a **diretiva -c**.
  - O comando a seguir seleciona a terceira coluna de cada linha:
  - **\$ cut -c3 exemplos\_cut.txt**
    - Crie o arquivo exemplos\_cut.txt junto com o professor e aplique o **cut -c3**
- Apenas alguns exemplos serão mostrados, mas o cut possui inúmeras outras possibilidades.

a	b	c	d
e	f	g	h
i	j	k	l
m	n	o	p

Conteúdo  
do arquivo  
(junto e sem  
espaço na  
mesma  
linha)

## PROCESSAR STREAMS DE TEXTO USANDO FILTROS

- Seleção de campos específicos de um arquivo
  - O comando cut também aceita delimitadores.
  - Um exemplo clássico é o arquivo `/etc/passwd`, cujos registros contém diversos valores separados pelo delimitador `:`
  - `root:x:0:0:root:/root:/bin/bash`
  - Como podemos ver, o arquivo `/etc/passwd` contém 8 campos.
  - Para seleccionar o nome de todos os usuários definidos no arquivo `/etc/passwd`, fazemos:
    - **cut -d: -f1 /etc/passwd**
- \*o **-d** define o delimitador de campos, e o **-f** qual coluna.

# PROCESSAR STREAMS DE TEXTO USANDO FILTROS

## Comando HEAD

- Exibe as primeiras linhas de um ou mais arquivos (o “head”, ou cabeçalho do(s) arquivo(s)).
- **Sintaxe**
  - **head** [opções] [arquivos]
- Exemplos
  - **head** arquivo1.txt (exibe as 10 primeiras linhas do arquivo, que o padrão);
  - **head -n 15** arquivo (exibe as 15 primeiras linhas do arquivo).



Faça testes junto com o professor no /etc/passwd

# PROCESSAR STREAMS DE TEXTO USANDO FILTROS

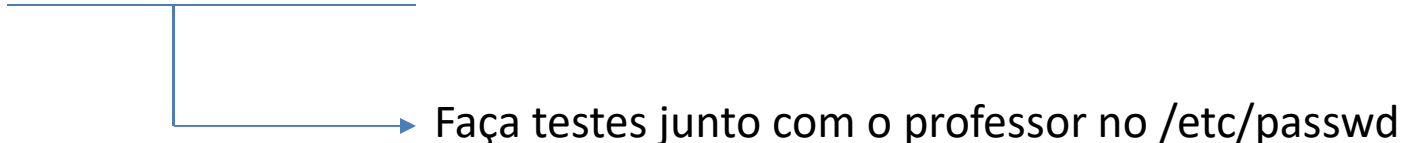
## Comando TAIL

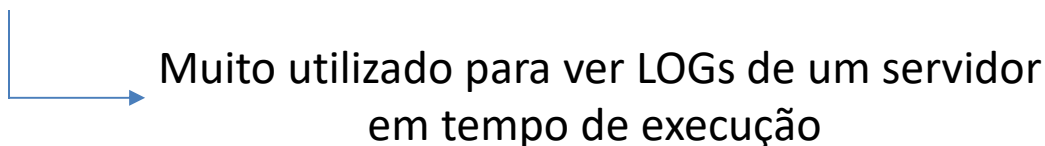
- Exibe as ultimas linhas de um ou mais arquivos (a “cauda” ou “tail” dos arquivos)
- **Sintaxe:**
  - tail [opções] [arquivos]
- **Exemplo:**
  - tail arquivo.txt (exibe as **ultimas** 10 linhas por padrão)



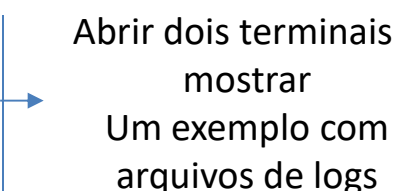
# PROCESSAR STREAMS DE TEXTO USANDO FILTROS

## Comando TAIL

- Opções mais usadas:
  - -n : exibe a quantidade de linha especificada
    - **Exemplo:** tail -n 15 arquivo.txt

Faça testes junto com o professor no /etc/passwd
  - -f : exibe ativamente um arquivo à medida que ele vai sendo escrito por um processo.
    - **Exemplo:** tail -f arquivo.txt

Muito utilizado para ver LOGs de um servidor em tempo de execução



Abrir dois terminais e mostrar  
Um exemplo com arquivos de logs

# PROCESSAR STREAMS DE TEXTO USANDO FILTROS

## Comando SORT

- Ordena de forma alfabética ou numeral.
- **Sintaxe**
  - **sort** [opções] [arquivos]
- **Exemplo:**
  - **sort** texto.txt

# PROCESSAR STREAMS DE TEXTO USANDO FILTROS

## Comando SORT

- Opções mais usadas:
  - -f : classifica insensível ao caso (maiúsculas ou minúsculas);
  - -n: classifica numericamente;
  - -r: classifica em ordem reversa.



Crie um arquivo com vários numero aleatórios  
e outro com letras aleatórias  
junto com o professor  
e teste o comando.

Aplicar: `sort -n` e `sort -nr`

## EXERCÍCIOS

1. Utilizando o CUT, selecione o shell padrão de cada usuário no arquivo `/etc/passwd`;
2. Exiba as 15 primeiras linhas do arquivo `/var/log/messages`;
3. Exiba as 10 últimas linhas do arquivo `/var/log/messages`;
4. Insira 10 números aleatórios dentro de um arquivo e imprima os mesmos ordenados no terminal;
5. Veja instantaneamente os logs em `/var/log/messages` após reiniciar algum dos serviços que ficam em `/etc/init.d`.

**Perguntas ?**