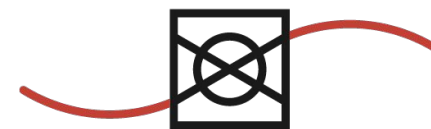




PPGCC



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
CEARÁ



LAPISCO

Comandos LINUX

Prof. Msc. Aldisio Medeiros

aldisio.medeiros@lapisco.ifce.edu.br

Introdução

- **Programa de Aula - Parte 1:**
 - **Apresentação do sistema baseado em Linux**
 - **Conhecendo os comandos**
- **Programa de Aula - Parte 2:**
 - **Conhecendo o git**
 - **Conhecendo o github**

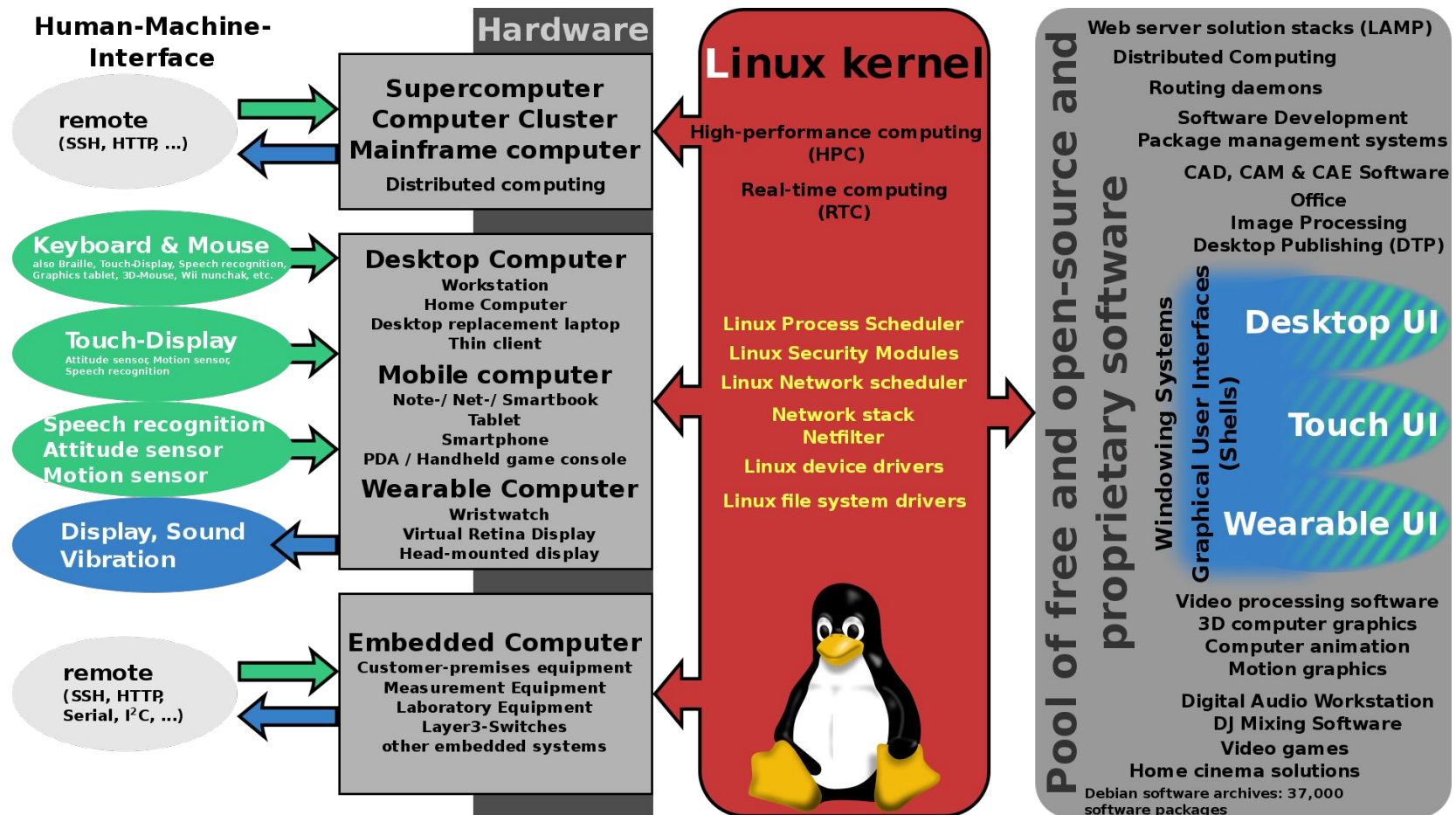
Comandos em sistemas Linux

Comandos em sistema Linux

- **Apresentação do sistema Linux**
 - O termo “Linux” não é um sistema operacional mas um kernel.
 - Desenvolvido iniciado por **Linus Torvald** no início da década de 90.
 - Linux = Linus e Unix (1984 - Richard Stallman) => **GNU/Linux** (SO)
 - O kernel **traduz as informações que recebe ao processador e aos demais elementos eletrônicos do computador.**
 - Agenda processos, gerencia a memória, controla o acesso a arquivos e a dispositivos de hardware
 - Parte destas funções é disponibiliza para os aplicativos por chamadas de sistema disponibilizadas via linguagem C.
 - É considerado um software livre
 - Lembre-se: **Livre não é pelo motivo de ser gratuito! (GPL v2)**

Comandos em sistema Linux

○ Apresentação do sistema linux



Comandos em sistema Linux

- **Apresentação do sistema Linux**

- Distribuições

- **Debian:**

- Ubuntu

- KUbuntu

- Linux Mint

- Kurimin (BR)

- **Slackware:**

- Suse

- OpenSUSE

- Arch Linux

- **RedHat:**

- Mandriva

- CentOS

- Fedora

Fedora

```
> sudo yum install gcc
```

OpenSUSE

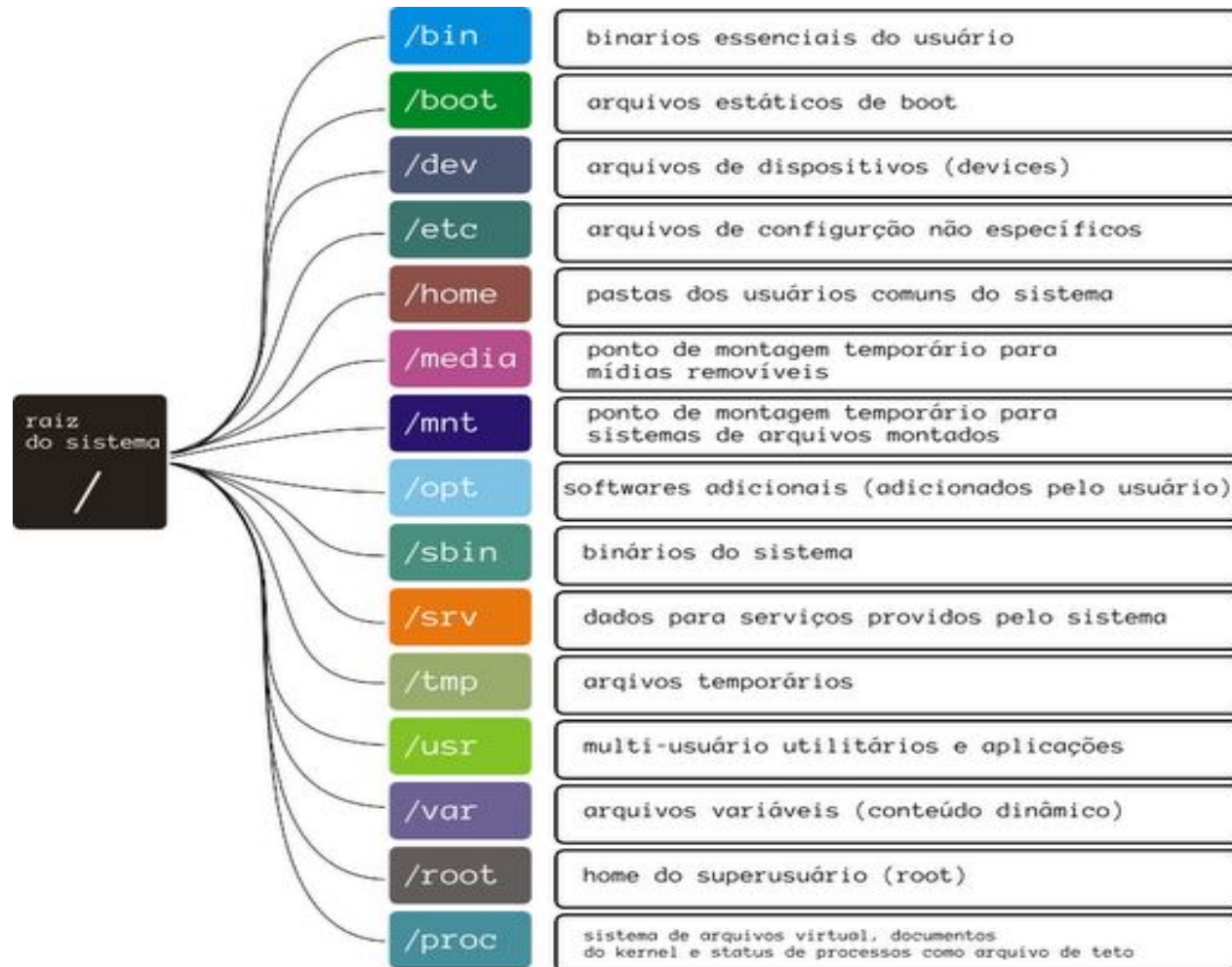
```
> sudo zypper in gcc
```

Ubuntu

```
> sudo apt-get update  
> sudo apt-get install gcc-4.4
```

Comandos em sistema Linux

○ Conhecendo os principais diretórios



Comandos em sistema Linux

- **Apresentação do terminal**
 - Antes de iniciar:
 - **Shell x Bash x Terminal** são a mesma coisa?
 - Agora sabemos o que é o terminal, quais informações iniciais podemos extrair?
 - Usuário logado;
 - Máquina onde está logado (host);
 - Tipo de usuário logado: \$ (comum) ou # (admin)
 - Diretório corrente: ~ padrão para /home/usuario



```
aldisio@Avell-A62-MUV: ~  
File Edit View Search Terminal Help  
aldisio@Avell-A62-MUV:~$
```

Diagram illustrating the components of the terminal prompt:

- usuário (points to `aldisio`)
- host (points to `Avell-A62-MUV`)
- tipo do usuário (points to `$`)
- diretório atual (points to `~`)

Comandos em sistema Linux

○ Comandos de manipulação de diretório: Listar diretório (ls)

- `$ ls`
- `$ ls -l`
- `$ ls -lah`
- `$ ls *.txt`
- `$ ls photo*.j`

-rw-r--r--	1	ppinto	ppinto	68	2007-01-31 22:36	teste.c
-rw-r--r--	1	ppinto	ppinto	55	2007-01-31 22:35	teste.c~
-rw-r--r--	1	ppinto	ppinto	12288	2007-01-31 22:35	teste.c.swp
-rw-r--r--	1	ppinto	ppinto	27	2007-01-31 22:07	teste.txt
drwx-----	2	ppinto	ppinto	4096	2006-12-23 22:24	.flash

Nome do ficheiro / directório

Data/hora de modificação

Tamanho do ficheiro (em bytes)

Grupo a que pertence o ficheiro/Directório

"Dono" do ficheiro / directório

Nº de links existentes no sistema para o ficheiro ou directório

Permissões do ficheiro/directório (permissões para user, grupo e others)

Indica se é ficheiro ou directório ('-' ou 'd')

- Vale destacar que o comando `ls` reconhece expressões regulares para filtragem de arquivos pelo nome. Vamos ver na prática.
 - `$ ls -l *\ \([1-9]\)*.png` - O que esse comando retornaria?

Comandos em sistema Linux

- **Comandos de manipulação de diretório**
 - `$ mkdir`
 - Cria um diretório vazio exemplo;
 - `$ rmdir`
 - Exclui um diretório (se estiver vazio).
 - `$ pwd`
 - Exibe o local do diretório atual.
 - `$ du -h [diretório ou arquivo]`
 - Mostra o tamanho do diretório/arquivo.
 - `$ whereis nomebinario`
 - Mostra onde se encontra determinado arquivo (binário)
 - Não aceita símbolos curingas como `*` ou `?`

Comandos em sistema Linux

- **Comandos de manipulação de diretório:**
 - **Trocar de diretório (cd)**
 - `$ cd [endereço do diretório]`
 - Existem alguns “atalhos” úteis:
 - `~` : diretório /home/usuário
 - `..` : diretório acima do endereço atual (pode ser combinado)
 - `#` : diretório /home do root (somente quando logado com root)
 - **Mover/ Renomear arquivos ou diretórios (mv)**
 - `$ mv endereco_de_origem endereco_de_destino`
 - **Pergunta:** Posso mover diretórios que estão em níveis acima do diretório corrente?

Comandos em sistema Linux

- **Comandos para manipulação de arquivos**
 - `$ cat filename` (`tac` para ordem inversa)
 - Mostra o conteúdo de um arquivo binário ou texto
 - `$ tail [-2] filename`
 - Mostra as últimas linhas de um arquivo. Útil para visualizar arquivos de log continuamente.
 - `$ head [-10] filename`
 - Mostra as primeiras linhas de um arquivo.
 - `$ less filename`
 - Mostra o conteúdo de um arquivo de texto com controle
 - `$ vim filename` ou `$ nano filename`:
 - Editores de texto em linha de comando
 - `$ chown novodono filename`
 - Altera o dono de arquivos ou diretórios;
 - `$ chmod [dono grupo outros] filename`
 - Altera permissões de arquivos ou diretórios (+x para execução)

Comandos em sistema Linux

- **Comandos para manipulação de arquivos**
 - `$ comando > log.txt`
 - Cria um novo arquivo com o resultado do comando
 - `$ comando >> txt`
 - Adiciona ao fim do arquivo (modo append)
 - `$ > arquivo.txt`
 - Cria arquivo vazio (similar ao “`$ touch arquivo.txt`”)
 - `$ comando | comando`
 - Operador chamado de pipe, direciona o fluxo de saída do comando para outro (tubo de conexão). Ex: `cat log.txt | wc -w`
 - `$ rm [-r] [arquivo ou diretório]`
 - Remoção de arquivos (também remove diretórios, mas com o parâmetro `-r`, que significa recursividade)
 - `$ cp [-r] [arquivo ou diretório]`
 - cópia de arquivos

Comandos em sistema Linux

- **Comandos para manipulação de arquivos de texto**
 - `$ cat arquivo1.txt arquivo2.txt`
 - Concatena verticalmente arquivos de texto
 - `$ paste arquivo1.txt arquivo2.txt > concat_H.txt`
 - Concatena horizontalmente arquivos de texto
 - `$ cut arquivo1.txt --delimiter=',' -f2`
 - Separa campos por delimitador
 - `$ sed 's/encontre_por/substitua_por/g' arquivo1.txt`
 - O sed é um editor de texto com diversas opções para manipulação de arquivos de texto.
 - `$ sed '1d' arquivo1.txt`
 - Remoção de linhas
 - `$ wc -m [w, l]`
 - Contador de caracteres, palavras e linhas
 - `$ grep -n 'log' arquivo1.txt (-inv: inverte a saída)`

Comandos em sistema Linux

- **Comandos para administração**
 - `$ man comando`
 - Mostra o manual do comando.
 - `$ passwd`
 - Modifica senha (password) de usuários
 - `$ htop`
 - Monitora os processos em execução além da memória principal e swap que está ocupada.
 - Possibilita filtros e monitoramento por hierarquia
 - `$ kill -9 PID`
 - Mata um processo pelo identificador (PID)
 - `$ killall -9 nome_processo`
 - Mata os processos com o nome informado
 - `$ nvidia-smi -l`
 - Monitora consumo da placa de vídeo
 - `$ watch -n tempo comando`

Comandos em sistema Linux

- **Comandos para administração de rede**
 - `$ip address`
 - Exibe as interfaces de redes ativas e as informações relacionadas a cada uma delas.
 - `$ /etc/init.d/networking restart`
 - Reinicializa o serviço de rede
 - `$ ping endereco` (alternativa: `traceroute -T, -U`)
 - Via protocolo `icmp` envia um pacote de teste de conectividade. Atenção: ICMP pode estar bloqueado.
 - `$ ssh pi@10.110.2.111`
 - Abre uma conexão criptografada em um host remoto.
 - `$ scp file.txt`
`remote_username@10.10.0.2:/remote/directory`
 - Envio/recebimento de arquivos via conexão `ssh`.

Comandos em sistema Linux

- **Comandos para produtividade**
 - **0 - Executar vários comandos em cascata de acordo com o resultado (I: Independentes; II: Dependentes; III: Alternativo)**
 - **\$**
 - **1- Encontrar todas as versões instaladas de um binário:**
 - Exemplo: Encontrar e exibir em modo de lista os binários do python 3.X
 - **\$**
 - **2- Somar linhas de um arquivo de texto**
 - Exemplo: Somar o número de blocos ocupados por arquivos em um diretório.
 - **\$**

Comandos em sistema Linux

- **Comandos gerais para produtividade:**
 - **0 - Executar vários comandos em cascata de acordo com o resultado (I: Independentes; II: Dependentes; III: Alternativo)**
 - `$ comando1 ; comando2 ; comando3`
 - `$ comando1 && comando2 && comando3`
 - `$ comando1 || comando2 || comando3`
 - **1- Encontrar todas as versões instaladas de um binário:**
 - Exemplo: Encontrar e exibir em modo de lista os binários do python 3.X
 - `$ whereis python | sed 's/\ /\\n/g' | grep python3`
 - **2- Somar linhas de um arquivo de texto**
 - Exemplo: Somar o número de blocos ocupados por arquivos em um diretório.
 - `$ ls -la | cut --delimiter='-' -f1 | sed '1d' | paste -sd+ | bc`

Comandos em sistema Linux

- **Comandos gerais para produtividade:**
 - **3 - Ordenar arquivo pela coluna**
 - `$`
 - **4 - Contar arquivos em uma pasta com base no nome**
 - `$`
 - **5 - Montar partição em memória RAM**
 - Montar:
 - `$ mount -t tmpfs none /mnt/ramdrive -o rw,size=100m`
 - Desmontar:
 - `$ umount /mnt/ramdrive`
 - **6 - Deletar arquivos em massa com base em um padrão**
 - `$`

Comandos em sistema Linux

- **Comandos gerais para produtividade:**
 - 3 - Ordenar linhas pela coluna
 - `$ sort file.txt --field-separator=',' --key=2 -n`
 - 4 - Contar arquivos em uma pasta com base no nome
 - `$ find minhapasta -type f \(-name '[1-3]'.txt \) | wc -l`
 - 5 - Montar partição em memória RAM
 - Montar:
 - `$ mount -t tmpfs none /mnt/ramdrive -o rw,size=100m`
 - Desmontar:
 - `$ umount /mnt/ramdrive`
 - 6 - Deletar arquivos em massa com base em um padrão
 - `$ find minhapasta -type f \(-name '[1-3]'.txt \) -delete`

Comandos em sistema Linux

- **Comandos para produtividade:**
 - **7 - Renomear arquivos em massa (rename)**
 - `$ rename [wildcard para arquivos] [padrão de replace]`
 - Neste caso podemos utilizar parte da expressão que aprendemos com o comando `sed`.
 - O primeiro parâmetro refere-se ao padrão de busca dos arquivos que serão afetados
 - O segundo parâmetro é o padrão de substituição da string.
 - **8 - Contar número de linhas em múltiplos arquivos de texto**
 - `$`
 - **9 - Print da saída do comando na tela e em arquivo:**
 - `$`

Comandos em sistema Linux

- **Comandos para produtividade:**
 - **7 - Renomear arquivos em massa (rename)**
 - `$ rename [wildcard para arquivos] [padrão de replace]`
 - Neste caso podemos utilizar parte da expressão que aprendemos com o comando `sed`.
 - O primeiro parâmetro refere-se ao padrão de busca dos arquivos que serão afetados
 - O segundo parâmetro é o padrão de substituição da string.
 - **8 - Contar número de linhas em múltiplos arquivos de texto**
 - `$ wc -l *.txt`
 - **9 - Print da saída do comando na tela e em arquivo:**
 - `$ top 2>&1 | tee output.txt`

Comandos em sistema Linux

- **Comandos para produtividade:**
 - **7- Remover as duas últimas linhas de um arquivo de log**
 - \$
 - **8- Copiar arquivos especificados pelo nome para um diretório**
 - \$
 - **9- Listar arquivos de um diretório desconsiderando os subdiretórios, mas com opção de alternar entre os níveis de subdiretórios**
 - \$

Comandos em sistema Linux

- **Comandos para produtividade:**
 - **7- Remover as duas últimas linhas de um arquivo de log**
 - ?
 - **8- Copiar arquivos especificados pelo nome para um diretório**
 - `$ find . -type f \(-name 'colunas_indices_[1-9].*' \) -exec cp {} temp/ \;`
 - **9- Listar arquivos de um diretório desconsiderando os subdiretórios, mas com opção de alternar entre os níveis de subdiretórios**
 - `$find . -type f -maxdepth 1 \(-name 'colunas_indices_[1-9].*' \)`

Comandos em sistema Linux

- **Executando scripts em segundo plano**
 - **Utilize o comando screen!**
 - `$ screen -S nome_screen`
 - **Listando screens ativas:**
 - `$ screen -ls`
 - **Conectando a screen existente:**
 - `$ screen -r nome_screen`

Comandos em sistema Linux

- **Agendamento de tarefas**

- Tarefas recorrentes podem ser automatizadas via agendamento de tarefas por meio da cron!
 - `$ crontab -e`
 - Padrão:

```
# _____ min (0 - 59)
# |_____ hour (0 - 23)
# | |_____ day of month (1 - 31)
# | | |_____ month (1 - 12)
# | | | |_____ day of week (0 - 6) (0 to 6 are Sunday to
# | | | | Saturday, or use names; 7 is also Sunday)
# | | | |
# | | | |
# * * * * * command to execute
```

```
# @reboot screen -dmS facerec-service bash -c '/home/sspds-server/Documents/facial-recognition/start-face
# 30 06 * * * /home/sspds-server/Documents/facial-recognition/./stop-facerec.sh; screen -dmS facerec-serv
# 00 22 * * * /home/sspds-server/Documents/facial-recognition/./stop-facerec.sh
```

Obrigado pela atenção!
- Dúvidas?