

Introd. Software Livre

A Interface de comando

Prompt de comando

- Comumente é representado por
- [usuario]@[maquina]:[diretorio_atual][#|\$]
 - Ex.: tadeu@servidor:/home\$
- Login do usuário atual = tadeu
- Nome de host da máquina = servidor
- Diretório padrão onde os comandos serão executados = /home
- \$ identifica um prompt de usuário comum
- # indicaria o prompt de um usuário privilegiado

Permissões

- Ao listar arquivos em uma pasta pode-se ver as permissões de cada um:
- As permissões são:
 - W - Write
 - R – Read
 - X – Execute
- Para cada arquivo as permissões são dadas ao dono, ao grupo e aos outros.
- Assim as permissões aparecem na forma:
- **-wrxwrxwrx** – dono grupo outros
- O primeiro caracter pode ser um **D** o que indica que este arquivo é na verdade um diretório

Permissões

- Há outros tipos de arquivos:
- b – arquivo especial de blocos Ex.: /dev/sda
- c – arquivo especial de caracter Ex.: /dev/tty
- p – canal (criado para comunicação entre processos)
- s – socket (criado para comunicação entre máquinas na rede)

Permissões

- Mudando permissões
- `chmod` – change mode
- `chmod [modo] arquivo`
- Onde modo pode estar no formato numérico ou textual
- Textual
 - `[ugo][+ -=][rwx]`
- Numérico
 - `000 666 777`

Permissões

- Quem
 - u para dono
 - g para o grupo
 - o para outros
- Operadores
 - + Adicionar
 - - Retirar
 - = Igualar
- Ex.:
 - u=rw – Dono tem permissão de leitura e escrita
 - u+x – Adicionar permissão de execução para o dono

Permissões

- Modo numérico
- Mais comumente usado
 - R = 4
 - W = 2
 - X = 1
- Para combinações basta somar os valores
 - $RWX = 4 + 2 + 1 = 7$
 - $R-X = 4 + 1 = 5$
 - $RW- = 4 + 2 = 6$
- chmod pode então ser usado com os valores para u/g/o de uma vez só. Ex.:
 - `chmod 755 arquivo.sh` – dono tem privilegio 7 / grupo 5 / outros 5

Permissão	Valor
---	0
--X	1
-W-	2
-WX	3
r--	4
r-X	5
rw-	6
rwX	7

Permissões

- `chown` – muda o dono e o grupo de um arquivo
- `chown [dono]:[grupo] arquivo`
- `chgrp` também pode ser usado para mudar apenas o grupo do arquivo

Variáveis de ambiente

- São variáveis comumente definidas na inicialização do sistema
- Controlam preferências do usuário ou localização de arquivos
- A variável PATH indica o caminho padrão para buscar programas no sistema
- Importante: para usar o valor contido dentro da variável é necessário adicionar \$ assim para usar o valor de PATH usamos \$PATH

Variáveis de ambiente

- Atribui-se valores a variáveis com um sinal de igual
- `PATH=/usr/bin`
- Para ver o valor de uma variável use `echo` Ex.:
- `echo $PATH`
- Para que a variável seja visível em todo o sistema é necessário exportá-la:
- `export PATH`
- `env` – exibe as variáveis de ambiente atuais

Variáveis de ambiente

- Tarefa comum:
- Adicionar uma pasta ao PATH:
- `PATH = $PATH:/home/nova_pasta`
- `export PATH`
- note que as pastas em PATH são separadas por `:` e não `;`

Editando um arquivo

- Há muitos editores de modo texto
 - nano/pico – interface mais simples, nem sempre presente na instalação padrão
 - emacs – editor avançado com muitas funções e os mais variados plugins
 - vim – editor avançado, presente em quase qualquer instalação linux, recursos para programadores e administradores de rede

Vim

- Dois modos
 - Modo de inserção
 - Modo normal
- Comandos Básicos:
 - Esc – ir para o modo normal
 - i – Ir para o modo de inserção de texto
 - :q! - sair sem salvar
 - :x – sair salvando
 - :w – gravar o arquivo
 - /texto – para buscar texto em um arquivo
 - G – ir para o final do arquivo
 - g – ir para o início do arquivo

nano

- Editor de texto mais simples
- Interface mais parecida com a de editores gráficos
- Funções ativadas através de combinações de teclas CTRL + <TECLA>
- Os atalhos para as funções podem ser vistos na parte inferior da tela

Atalhos úteis

- CTRL + U – apaga a linha de comando atual
- CTRL + D – encerra a sessão atual
- CTRL + L – limpa a tela
- CTRL + R – busca comandos no histórico
- **TAB auto-completa um comando**