

SEMINÁRIOS I

LINUX: SHELL BÁSICO

EDITORES VIM E NANO

Profs. Alexei Machado e João Caram

Localizando programas

2

□ Execute:

which more

whereis more

locate more

Criando diretórios

3

- mkdir: cria diretórios
 - ▣ Caminhos relativos e absolutos
 - ▣ Execute:
`mkdir meuDir`
`mkdir seminarios/trabalhos`
`mkdir -p seminarios/trabalhos`

Manipulando arquivos

4

- cp: cópia de arquivos

cp <origem> <destino>

- Atenção para uso de coringas: * ?

- *Auto-complete* com *tab*

Manipulando arquivos - cp

5

□ Execute

```
cp /var/log/boot.log ./
```

```
mkdir testevar
```

```
cp -r /var ./testevar
```

□ Opção `-u` : copia apenas os arquivos atualizados (*updated*)

Manipulando arquivos

6

- mv: mover (renomear) arquivos e diretórios

- Execute:

```
mv boot.log logDeBoot.txt
```

```
mv meuDir dirTeste
```

Manipulando arquivos

7

- rm: remover (apagar) arquivos
- Execute:
rm logDeBoot.txt
- ATENÇÃO
 - rm -i → pergunta sempre
 - rm -f → nunca pergunta

Executando comandos em série

8

- ❑ Podemos conectar a resposta de um comando à entrada de outro usando o conceito de pipe
comando1 | comando2
- ❑ Muito útil quando da execução de *shell scripts*
- ❑ Execute:

```
ls -la /var/log
```

```
ls -la /var/log | less
```


Redirecionando entrada e saída

9

- Por padrão, a entrada de dados do *shell* é o teclado
 - ▣ Ou algum argumento especificado
- Por padrão, a saída de dados do *shell* é o terminal
- Mas se quisermos utilizar estes dados posteriormente... o que fazer?

Redirecionando saídas

10

- Os operadores `>` e `>>` redirecionam as saídas de comandos para arquivos ao invés da tela
 - ▣ `>` - criação ou sobrescrita de arquivo
 - ▣ `>>` - criação ou anexação à arquivo existente
- Execute

```
ls /etc > teste.txt
more teste.txt
```

Redirecionando saídas

11

□ Execute

```
ls /etc > teste.txt
```

```
more teste.txt
```

```
ls > teste.txt
```

```
more teste.txt
```

```
ls /etc >> teste.txt
```

```
more teste.txt
```

Redirecionando entradas

12

- Analogamente, `<` pode ser usado para indicar o redirecionamento de entrada de dados

- Execute

```
ls
```

```
ls | sort
```

```
sort < teste.txt
```


- *V*isual editor *I*Mproved
- Editor padrão dos sistemas da família UNIX
- Opera em modo de inserção (tradicional) ou modo de comandos, comutado através de ESC
- Modo de inserção ativado com a tecla *i*

- Modo de comandos é iniciado com a tecla de dois pontos - :
- :q – sair do editor
- :w – salvar o arquivo
- :wq – salvar e sair
- :q! – sair sem salvar

□ Comandos:

- `:w <novoarquivo>` – salva em um novo arquivo
- `:0` – mover para a primeira linha
- `:$` – mover para a última linha
- `:n` – sendo n um número, vai para a linha n

nano

17

- nano é um editor de textos criado pelo projeto GNU para fornecer um editor livre similar ao pico (um editor clássico de emails do UNIX)
- Interface amigável e, em geral, auto-explicativa
- F1 ou ctrl-G (^G) mostram a ajuda do nano

OBRIGADO.

Dúvidas?