

Introdução ao Linux Aula 3

Abordagem prática para novos usuários

Fernando Borges Whitaker

Editores de texto

- ▶ Editor de texto - Permite editar texto puro sem formatação
 - ▶ Bloco de notas, gedit, nano, **vim**
- ▶ Processadores de texto - Texto com formatação, tabelas, figuras
 - ▶ Word, LibreOffice Writer

VIM (VI Improved)

- ▶ Por quê usar?
 - ▶ É possível usá-lo num terminal (conectando-se a um computador remoto)
 - ▶ É um padrão. Está presente em qualquer sistema Linux
- ▶ Editor de textos muito poderoso e **MUITO COMPLICADO** mas pode salvar seu dia
- ▶ Sugestão: saiba usá-lo apenas o suficiente para salvá-lo de situações de emergência

Vimbook

- ▶ Se você quiser se tornar um Mestre Ninja do VIM
- ▶ <https://github.com/cassiobotaro/vimbook>

Vimtutor

- ▶ Aproximadamente 30 minutos
 - ▶ <ESC> :q! (sai sem salvar)
 - ▶ <ESC> :w salva
 - ▶ <ESC> :wq salva e sai
-
- ▶ Pratiquem o vim nos arquivos que vocês criaram nas aulas passadas.
 - ▶ Vamos praticar até as 10h

Alias

- Comandos alias : apelidos de comandos
- `alias novo_comando=comando`
 - `alias rm="rm -i"`
 - `alias ls="ls -F -color"`
 - `alias ..='cd ..'`
 - `alias ...='cd ../..'`

Redirecionar Entrada/Saída

- No Linux todo acesso é feito por arquivos e temos três arquivos padrões de entrada/saída de informações:
 - stdin: Entrada padrão (teclado)
 - stdout: Saída padrão (tela)
 - stderr: Saída padrão para erros (tela)
- Podemos redirecionar a saída como:
 - `ls > arquivo.ls`
 - `ls | more`

Redirecionar Saída

- Redirecionar stdout para arquivo (zera o arquivo antes)
 - comando > arquivo ou comando 1> arquivo
 - ls > arquivo
 - ls 1> arquivo
- Redirecionar stdout para arquivo (acrescenta no final - append)
 - comando >> arquivo
 - ls >> arquivo

Redirecionar Saída

- Redirecionar stderr para arquivo (zera o arquivo antes)
 - comando 2> arquivo
 - ls arquivo_nao_existe 2> arquivo_erro
- Redirecionar stdout+stderr para arquivo (zera o arquivo antes)
 - comando &> arquivo
 - comando > arquivo 2>&1
 - ls arquivo_existe arquivo_nao_existe 1> saida_ok 2> saida_erro
 - ls arquivo_existe arquivo_nao_existe &> saidas
 - ls arquivo_existe arquivo_nao_existe > saidas 2>&1

Redirecionar Saída

- Redirecionar stdout+stderr para arquivo (acrescenta no final)
 - comando &>> arquivo
 - comando >> arquivo 2>&1
 - comando &>> arquivo
 - ls arquivo_existe arquivo_nao_existe &>> saidas
 - ls arquivo_existe arquivo_nao_existe >> saidas 2>&1

Redirecionar Entrada

- Arquivo para comando
 - `sort < arquivo_in`
 - `sort < arquivo_in > arquivo_ou`
- Redirecionar entrada e saída
 - `sort < arquivo_in > arquivo_out1 2> arquivo_out2`
 - `sort < arquivo_in &>> saida`
 - `sort < arquivo_in >> saida 2>&1`

Exercício

- ▶ Crie um arquivo chamado nomes.txt
- ▶ Escreva o nome de 10 pessoas fictícias, um por linha, com alguns nomes repetidos. Use o vim
- ▶ `cat arquivo.txt | sort`
- ▶ `cat arquivo.txt | sort | uniq`
- ▶ `cat arquivo.txt | sort | uniq -c`
- ▶ `cat arquivo.txt | sort | head -n 3`
- ▶ `cat arquivo.txt | sort | tail -n 3`
- ▶ `cat arquivo.txt | sort | head -n 3 > tres_primeiros.txt`

Exercícios

- Defina um alias para editar um arquivo como:
 - `alias edicao=gedit`
`~/fulano/aula3/exercicios/ex01.txt`
- Execute o alias de comando `edicao`

Exercícios

- Escreva no arquivo
`/home/curso/fulano/aula3/exercicios/ex02.txt`
comandos para:
 - Redirecionar saída do comando `ls` para `arquivo.txt`
 - Redirecionar `arquivo.txt` para o comando `sort`
 - Redirecionar `arquivo.txt` para o comando `sort` e a saída desse comando para `resultado.txt`