

Ministério da Educação
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca
UNED Nova Friburgo
Curso Técnico em Informática Integrado ao Ensino Médio

Arquivos e redirecionamento

Sistemas Operacionais

Prof. Bruno Policarpo Toledo Freitas
13 de maio de 2020
bruno.freitas@cefet-rj.br

Objetivos

- **Aprender a salvar resultados de comandos e operações em arquivos**
- **Aprender a executar comandos com parâmetros lidos de arquivos**

Arquivos-padrão

- **Tudo em um sistema GNU/Linux é baseado em arquivos**
- **Sistemas Operacionais GNU/Linux possuem 3 canais de comunicação com o usuário:**
 - Entrada padrão (*stdin*)
 - Identificador **0**
 - Saída padrão (*stdout*)
 - Identificador **1**
 - Saída de erros (*stderr*)
 - Identificador **2**

Redirecionamento para arquivos

- **Existem 3 maneiras de se redirecionar saídas e entradas de programas para arquivos:**
 - comando **>** arquivo : salva a saída de *comando* em *arquivo*
 - comando **<** arquivo : *comando* recebe como entrada o conteúdo de *arquivo*
 - comando **>>** arquivo : anexa o conteúdo de *comando* em *arquivo*

Redirecionamento *comando* → *arquivo*

- **Salva a saída de comandos em arquivos**
- **Trunca o arquivo-destino**
- **Exemplos:**

`ls /etc/*.conf > arquivos.txt`

`pwd > pasta.txt`

Redirecionamento *comando* ← *arquivo*

- A entrada do *comando* é o conteúdo em *arquivo*
- Exemplo:
 cat < arquivos.txt
 cd < pasta.txt

Redirecionamento *comando* → *arquivo*

- **Anexa a saída de um comando em um arquivo existente**
- **Cria o arquivo se ele não existir**
- **Exemplo:**

`ls *.txt > arquivos.txt`

`ls /etc >> arquivos.txt`

`ls ~/Downloads >> arquivos.txt`

Redirecionamento saída → entrada

- **Utiliza a saída de um programa como parte de outro programa**
 - *comando1* ``comando2`` : executa *comando2*, substitui sua saída e então executa *comando1*

Exercício

- 1)Salve todos os processos em execução pelo seu usuário no arquivo *“processos_bruno.txt”*
- 2)Salve todos os processos do usuário root no arquivo *“processos_root.txt”*
- 3)Usando o cat, concatene os arquivos acima e salve-os no arquivo *processos.txt*
- 4)Repita o exercício 2 e 3 mas agora usando o >>

Pipes

- **Filosofia UNIX:**

“Tarefas complexas são realizadas pela concatenação de várias tarefas simples”

- **Pipes: utilizadas para redirecionar a saída de um programa para ser entrada de um outro**

- **Sintaxe:**

programa1 | programa2

Pipes

- **Exemplos:**

`cat /etc/passwd | less`

`apt-cache search codeblocks | grep 'codeblocks'`

`ps -e | wc -l`

`cat /etc/passwd | head -7 | tail -5`

Programas para Pipes

- Pipes são parte essencial de sistemas GNU/Linux
- Muitos programas são feitos para se utilizar junto com pipes:

head/less

sort

uniq

cut

join / paste

grep / egrep

sed

awk

Pipes & redirecionamentos

- **Pipes podem ser combinados com redirecionamentos.**

```
ps -e | grep 'root' > processos_root.txt
```

O programa tee

- **Usado quando queremos:**

- Redirecionar a saída de um programa mas ainda ver sua saída.
- Redirecionar a saída e salvar em arquivos com permissão de usuários / superusuários

- **Exemplo:**

`ls -lR $HOME > /home/bruno/allMyFiles`

`ls -lR $HOME > /etc/allMyFiles`

`ls -lR $HOME | sudo tee /etc/allMyFiles`

Redirecionamento do stderr

- **Se redirecionarmos o o stdout com o `>`, mensagens de erro ainda irão aparecer no terminal**
 - `ls /eunaoexisto > /dev/null`
- **Para redirecionar o stderr, devemos utilizar:**
 - `ls /eunaoexisto 2>/tmp/errors`

Redirecionamento do stderr para o stdout

- Para capturar o stdout e o stderr em um mesmo arquivo, devemos utilizar a seguinte sintaxe:
 - ls *.bak > listfile **2>&1**

O dispositivo nulo

- **/dev/null**
- **Utilizado quando queremos descartar a saída de programas**

`ls -l > /dev/null`

O comando exec

exec 3>/tmp/thirdfile

exec 4>/tmp/fourthfile

echo "drib" >&3

echo "drab" >&4

echo "another drib" >&3

echo "another drab" >&4

exec 3>&-

exec 4>&-

Exercícios

- 1) Salve os nomes de todos os arquivos .conf da pasta /etc no arquivo *\$HOME/todasConfiguracoes.conf*
- 2) Quantos arquivos .conf há na pasta /etc?
- 3) Procure as informações do seu usuário no arquivo /etc/passwd. (Dica: utilize grep com pipes)
- 4) Quantos usuários existem no sistema? (Dica: conte quantas linhas possui o arquivo /etc/passwd com pipes)

Referências

- FILHO, João Eriberto Mota. **Descobrimdo o Linux: entenda o sistema operacional GNU/Linux**. 3ª. ed. São Paulo: Novatec Editora, 2012.
 - Capítulo 10: seção 10.22
- <http://write.flossmanuals.net/command-line/stand-ard-files/>
- <http://write.flossmanuals.net/command-line/file-st-structure/>