

Laboratório 2: Redirecionamento e caracteres curinga

Professor: Diego da Silva de Medeiros

diegomedeiros@ifsc.edu.br

1 Objetivos

1. Conceituar e trabalhar entradas e saída padrão, saída de erros padrão e pipes;
2. Conceituar e trabalhar o uso de caracteres curinga para trabalhar com arquivos e diretórios no Linux
3. trabalhar comandos cut, find, wc

2 Redirecionamento de entrada e saídas

Sempre existem 3 arquivos “abertos” por “default” para um determinado processo. São eles a entrada padrão (stdin, normalmente associada ao teclado), stdout (stdout, normalmente associada a tela) e a saída padrão de erros (stderr, normalmente associada a tela). Estes “arquivos” podem ser modificados resultando no que se chama de redirecionamento.

Cada arquivo aberto, associado a um processo possui um descritor identificado por um número que se constitui em um índice de uma tabela de arquivos abertos associados a um processo. Segue-se um resumo da forma de realizar redirecionamento.

Redireção de stdout (arquivo lista é sobreescrito)

```
ls > lista
```

Redireção de stdout fazendo um *append* ao arquivo de saída

```
ls >> lista
```

Criar/sobreescrever um arquivo de tamanho zero, similar ao touch

```
> lixo
```

Nota: Resumo de comandos de redireção que afetam a linha de comando

1>filename # Redireciona stdout para o arquivo “filename”.

1>>filename # Redireciona e acrescenta a stdout para o arquivo “filename”

2>filename # Redireciona stderr para o arquivo “filename”.

2>>filename # Redireciona e append stderr para o arquivo “filename”.

&>filename # Redireciona ambos stdout and stderr para o file “filename”.

1. Listar o diretório corrente, de forma detalhada e em ordem crescente de modificação, redirecionando a saída para um arquivo *ListaOrdenada*

Resposta:

```
aluno$ ls -l > ListaOrdenada
```

2. Acrescentar ao final do arquivo (“append”) do arquivo *ListaOrdenada*, a data atual

Resposta:

```
aluno$ date >> ListaOrdenada
```

3. Criar um arquivo chamado teste1, com a frase “Alo Mundo”, usando o comando `cat`. Lembre que se você não fornecer o arquivo de entrada a ser visualizado, será utilizado a entrada pelo terminal. Use `<CTRL+D>` para encerrar a entrada de dados.

Resposta:

```
aluno$ cat > teste1
Alo Mundo
<CTRL+D>
```

4. Repita o exercício anterior usando o comando `echo` e redirecionamento de saída;

Resposta:

```
aluno$ echo Alo Mundo > teste1
```

5. Criar um arquivo vazio usando somente o comando `> arquivo`;

Resposta:

```
aluno$ > Arquivo
```

6. Aplicar um comando `ls` ao diretório `/eetc` (intencionalmente errado) redirecionando a saída para `lixo1`;

Resposta:

```
aluno$ ls /eetc > lixo1
```

7. Conferir o conteúdo de `lixo1`

Resposta:

```
aluno$ cat lixo1
```

8. Repetir o comando `ls` ao diretório `/eetc` (intencionalmente errado) redirecionando a saída de erros para `lixo2`;

Resposta:

```
aluno$ ls 2> lixo2
```

9. Conferir o conteúdo de `lixo2`

Resposta:

```
aluno$ cat lixo2
```

3 Pipes

A filosofia do unix/linux é resolver pequenos problemas com comandos simples. É possível no entanto resolver problemas mais complexos encadeando saídas em entradas, usando pipes

1. Procurar arquivos com terminação `.conf` no diretório `/etc`, usando o comando `find` e redirecione a saída de erro para `/dev/null` (equivalente a uma lixeira);

Resposta:

```
aluno$ find /etc -name .conf 2> /dev/null
```

2. Repetir o exercício anterior fazendo um pipe para o comando `wc` (com flag `-l` para contar o número de linhas) a fim de verificar o número de ocorrências dos arquivos procurados.

Resposta:

```
aluno$ find /etc -name .conf | wc -l
```

3. Eliminar a saída de erros do comando anterior:

Resposta:

```
aluno$ find /etc -name .conf 2> /dev/null | wc -l
```

4. Usar um pipe com `cat` e `wc` para determinar quantas linhas existe no arquivo `/etc/passwd`:

Resposta:

```
aluno$ cat /etc/passwd | wc -l
```

5. Repetir o exercício anterior para verificar quantos caracteres contem o arquivo:

Resposta:

```
aluno$ cat /etc/passwd | wc
```

6. Listar o campo 1 do arquivo `/etc/passwd`:

Nota: conferir com o comando `man` o comando `cut`

Resposta:

```
aluno$ cat /etc/passwd | cut -f 1 -d:
```

4 Caracteres curinga

No diretório home do seu usuário, crie um diretório de nome “Curinga”, e crie dentro dele os seguintes arquivos:

Nota: arquiva1.doc arquiva1.jpeg arquiva2.doc arquiva3.jpeg arquiva4.jpeg arquiva5.png arquiva7.png arquivo1.doc arquivo1.jpeg arquivo2.doc arquivo3.jpeg arquivo4.jpeg arquivo5.png arquivo7.png file1.doc file1.jpeg file11.doc file12.doc file13.doc file14.doc file2.doc file22.doc file3.jpeg file4.jpeg file5.png file7.png

1. Liste o conteúdo do diretório */etc*.

Resposta:

```
aluno$ ls /etc
```

2. Liste o conteúdo do diretório */etc*, mas somente os arquivos que começam com a letra “s”.

Nota: Caso alguma pasta comece com a letra “s”, o comando *ls* irá listar o conteúdo das pastas. Procure uma forma de evitar isso usando o *man* do *ls*.

Resposta:

```
aluno$ ls -d /etc/s*
```

3. Liste o conteúdo do diretório */etc*, mas somente os arquivos que começam com a palavra “sudo”.

Resposta:

```
aluno$ ls -d /etc/sudo*
```

4. Liste o conteúdo do diretório */etc*, mas somente os arquivos que terminam com os caracteres “.d”.

Resposta:

```
aluno$ ls -d /etc/*.d
```

5. Entre no diretório “*~/Curinga/*”

Resposta:

```
aluno$ cd ~/Curinga
```

6. Liste os arquivos que iniciam com a letra “a”.

Resposta:

```
aluno$ ls -d a*
```

7. Liste todos os arquivos que começam com a letra “a”, mas não os de nome “arquiva*”.

Resposta:

```
aluno$ ls -d arquivo*
```

8. Liste todos os arquivos que contém a extensão “.png”.

Resposta:

```
aluno$ ls -d *.png
```

9. Liste todos os arquivos que contém a extensão “.doc”, mas não os arquivos “file*”.

Resposta:

```
aluno$ ls -d [^f]*.doc
```

10. Crie um diretório chamado “Dir1” dentro do diretório “Curinga”. Copie todos os arquivos com o número “1” no nome para o diretório “Dir1”.

Resposta:

```
aluno$ mkdir Dir1  
aluno$ cp *1* Dir1
```

11. Liste todos os arquivos com o número “1” no nome, mas não os arquivos que começam com a letra “a” ou o diretório “Dir1”.

Resposta:

```
aluno$ ls -d [^aD]*1*
```

12. Liste todos os arquivos com o número “1” no nome, menos os que possuem extensão “jpeg”.

Resposta:

```
aluno$ ls -d *1*.[^j]*
```

13. Liste somente os arquivos “file11.doc, file12.doc, file13.doc, file14.doc”.

Resposta:

```
aluno$ ls -d file1?.doc
```

14. Liste todos os arquivos com o número “1” no nome, menos os arquivos “file11.doc, file12.doc, file13.doc, file14.doc”.

Resposta:

```
aluno$ ls -d *[^1]1.*
```

15. Liste os arquivos “file1.doc, file1.jpeg, file2.doc”.

Resposta:

```
aluno$ ls -d file[12]
```

16. Liste os arquivos “arquiva1.doc, arquiva2.doc, arquivo1.jpeg, arquiva1.jpeg, arquivo1.doc, arquivo2.doc” .

Resposta:

```
aluno$ ls -d arquiv?[12].*
```

17. Liste os arquivos “file13.doc, file14.doc”.

Resposta:

```
aluno$ ls -d file1[34]*
```