



## Comandos

### de Redirecionamento e Pipe

Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro

Prof. Edwar Saliba Júnior

Agosto / 2017



## Comando >

- Redireciona a saída padrão de um programa/comando/script para algum dispositivo ou arquivo ao invés do dispositivo de saída padrão (tela). Quando é usado com arquivos, este redirecionamento cria ou substitui o conteúdo do arquivo;
- Por exemplo, você pode usar o comando `ls` para listar arquivos e usar `ls >listagem.txt` para enviar a saída do comando para o arquivo `listagem.txt`. Use o comando `cat` para visualizar o conteúdo do arquivo `listagem.txt`;
- O mesmo comando pode ser redirecionado para o segundo console `/dev/tty2` usando: `ls >/dev/tty2`, o resultado do comando `ls` será mostrado no segundo console (pressione `ALT` e `F2` para mudar para o segundo console e `ALT` e `F1` para retornar ao primeiro). O mesmo resultado pode ser obtido com o comando `ls 1>/dev/tty2`, sendo que o número 1 indica que será capturada a saída padrão do comando;
- Para redirecionar somente a saída de erros do comando `ls`, use a sintaxe: `ls 2>/tmp/erros-do-ls`



## Comando >>

- Redireciona a saída padrão de um programa/comando/script para algum dispositivo ou adiciona as linhas ao final de arquivo ao invés do dispositivo de saída padrão (tela). A diferença entre este redirecionamento duplo e o simples, é se caso for usado com arquivos, adiciona a saída do comando ao final do arquivo existente ao invés de substituir seu conteúdo;
- Por exemplo, você pode acrescentar a saída do comando `ls` ao arquivo listagem do capítulo anterior usando `ls / >>listagem.txt`. Use o comando `cat` para visualizar o conteúdo do arquivo `listagem.txt`.



## Comando <

- Direciona a entrada padrão de arquivo/dispositivo para um comando. Este comando faz o contrário do anterior, ele envia dados ao comando;
- Você pode usar o comando `cat <teste.txt` para enviar o conteúdo do arquivo `teste.txt` ao comando `cat` que mostrará seu conteúdo (é claro que o mesmo resultado pode ser obtido com `cat teste.txt` mas este exemplo serviu para mostrar a funcionalidade do `<`).



## Comando <<

- Este redirecionamento serve principalmente para marcar o fim de exibição de um bloco. Este é especialmente usado em conjunto com o comando `cat`, mas também tem outras aplicações. Por exemplo, digite o comando a seguir e, onde tiver `<Enter>` aperte a tecla "Enter":

```
cat << final<Enter>
    este arquivo<Enter>
    será mostrado<Enter>
    até que a palavra final seja<Enter>
    localizada no início da linha<Enter>
    final<Enter>
```

- Qual foi o resultado?



## Comando | (Pipe)

- Envia a saída de um comando para a entrada do próximo comando para continuidade do processamento. Os dados enviados são processados pelo próximo comando que mostrará o resultado do processamento;
- Por exemplo: `ls -la | more`, este comando faz a listagem longa de arquivos que é enviado ao comando `more` (que tem a função de efetuar uma pausa a cada 25 linhas do arquivo);
- Outro exemplo é o comando `locate find | grep "bin/"`, neste comando todos os caminhos/arquivos que contém `find` na listagem serão mostrados (inclusive man pages, bibliotecas, etc.), então enviamos a saída deste comando para `grep "bin/"` para mostrar somente os diretórios que contém binários. Ainda assim, se a listagem ocupar mais de uma tela, podemos acrescentar o `more`: `locate find | grep "bin/" | more`
- Podem ser usados mais de um comando de redirecionamento (`<`, `>`, `|`) em um mesmo comando.



## Diferença entre `|` e `>`

- A principal diferença entre o "`|`" e o "`>`", é que o Pipe envolve processamento entre comandos, ou seja, a saída de um comando é enviado a entrada do próximo e o "`>`" redireciona a saída de um comando para um arquivo/dispositivo;
- Você pode notar pelo exemplo acima (`ls -la | more`) que ambos `ls` e `more` são comandos porque estão separados por um "`|`". Se um deles não existir ou estiver digitado incorretamente, será mostrada uma mensagem de erro;
- Um resultado diferente seria obtido usando um "`>`" no lugar do "`|`"; A saída do comando `ls -la > more` seria gravada em um arquivo chamado `more`.



## Comando tee

- Envia ao mesmo tempo o resultado do programa para a saída padrão (tela) e para um arquivo. Este comando deve ser usado com o pipe "|";

```
comando | tee [arquivo]
```

- Exemplo: `ls -la | tee listagem.txt`, a saída do comando será mostrada normalmente na tela e ao mesmo tempo gravada no arquivo `listagem.txt`.





## Referências

- GUIA FOCA GNU/Linux. Iniciante. Disponível em: <http://www.guiafoca.org/cgs/guia/iniciante/ch-redir.html>. Acesso em: 06 ago. 2017.