Oficina Linux --/--/----

Aula __: Entradas e Saídas Padrão

Professor: Eraldo eraldo.silveira@gmail.com

1 Objetivos

- Conceituar e trabalhar entradas e saída padrão, saída de erros padrão e *pipes*;
- Trabalhar os comandos *cut*, *find*, *wc*, *grep*,*head* e *tail*.

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*(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 das entradas e saídas do processo.

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 ao processo. Segue-se um resumo da forma de realizar redirecionamento.

Redireção de *stdout* (arquivo lista é sobrescrito):
ls > lista
Redireção de *stdout* fazendo um *append* ao arquivo de saída:
ls >> lista
Cria/sobreescreve um arquivo de tamanho zero, similar ao *touch*
> lixo
Resumo das redireções das saídas:

- 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 faz *append* da *stderr* para o arquivo *filename*
- &>filename # Redireciona ambos *stdout* and *stderr* para o arquivo *filename*

3 Exercícios

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

   ls -l > ListaOrdenada

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

   date >> ListaOrdenada

3. Criar um arquivo chamado *teste1*, com a frase "Alo Mundo", usando o comando *echo* e redirecionamento de saída;

   echo Alo Mundo > teste1

4. Criar um arquivo vazio usando somente > arquivo;

   > ArquivoVazio

Nota: Acho que esta é a forma mais rápida de criar um arquivo vazio!

5. Aplicar um comando ls ao diretório /eetc (intecionalmente errado) redirecionando a saída para *lixo1*;
   ls /eetc > lixo1
6. Conferir o conteúdo de *lixo1*
   cat lixo1
7. Repetir o comando *ls* ao diretório */eetc* (intecionalmente errado) redirecionando a saída para *lixo2*;
   ls /eetc 2> lixo2
8. Conferir o conteúdo de *lixo2*
   cat lixo2
9. Ordenar o arquivo passwd:
   sort < /etc/passwd > PasswdOrdenado

4 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. Listar de forma controlada (rolando a terra) o diretório */etc*:
   ls -l /etc | more
2. Procurar arquivos com terminação .conf no diretório /etc, usando o comando find e redirecionar a saída de erro para */dev/null* (equivalente a uma lixeira mas sem volta!);
   *find /etc -name *.conf 2> /dev/null*
3. 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.
   *find /etc -name *.conf | wc -l*
4. Eliminar a saída de erros do comando anterior:
   *find /etc -name *.conf 2> /dev/null | wc -l*
5. Usar um pipe com *cat* e *wc* para determinar quantas linhas existe no arquivo */etc/passwd*:
   cat /etc/passwd | wc -l
6. Repetir o exercício anterior para verificar quantos caracteres contem o arquivo:
   cat /etc/passwd | wc
7. Listar o campo 1 do arquivo *passwd*:
   cat /etc/passwd | cut -f 1 -d :

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

5 Outros exercícios

1. O comando grep pode ser usado para verificar ocorrências de palavras em (padrões) em um arquivo. Por exemplo: *grep aluno /etc/passwd*. Componha um comando usando *cut*, *grep* e *wc* para verificar quantos *userid* possuem a cadeia "un";
2. Gere um arquivo contendo somente os campos *userid* e nome completo de usuários que constam do arquivo *passwd*;
3. Pesquisar o comando *head* e gerar um arquivo chamado DezLinhas contendo as 10 primeiras linhas do arquivo */etc/passwd*;
4. Repetir o exercício anterior para as 5 primeiras linhas;
5. Pesquisar o comando *tail* e gerar um arquivo chamado *DezUltimas* com com as 5 últimas linhas do arquivo */etc/passwd* mas ordene estas linhas com o comando *sort* (use *pipes*);

6 Exercícios Adicionais

1. Criar um arquivo *TimesFutebol_1.txt* com a linha:

   Internacional:Porto Alegre:Brasil:America do Sul:vermelho

   Use o comando *echo* e redireção da saída. Verifique o conteúdo do arquivo com *cat*.

2. Repetir o exercício com o *Figueirense* concatenado a saída. Faça um *cat* para verificar o conteúdo;

3. Coloque pelo menos 5 linhas adicionais no arquivo usando a abordagem anterior;

4. Crie outro arquivo *TimesFutebol_2.txt* com 6 linhas e usando a mesma abordagem. Coloque times internacionais. Não se esqueça de separar os campos com ":";

5. Concatene os dois arquivos em um único chamado *TimesFutebol.txt*. Confira o resultado;

6. Use o comando *grep* para visualizar as linhas de ocorrência da palavra Brasil;

7. Use o comando *wc* para contar o número de linhas do arquivo;

8. Use um *pipe* com comandos *grep* e *wc* para contar o número de ocorrências de linhas onde aparece *Brasil*;

9. Melhore a solução do exercício anterior verificando a ocorrência de *Brasil* somente no campo associado ao país;

10. Conte o número de times que usam *vermelho* e que são da *America do Sul*;

11. Gere um arquivo *DuasColunas.txt* com duas colunas: o time e país;

12. Capture em um arquivo LinhasCapturadas todas as linhas entre a linha 3 e 7 inclusive. Use head e tail;

13. Criar uma árvore de diretório *Times* com subdiretórios associados aos continentes. Dentro dos continentes criar diretórios associados aos países e dentro dos países criar arquivos regulares com os times terminados em *.txt*. Dentro de cada arquivo coloque a cor predominante.

14. A partir do diretório de entrada procure em todos os diretórios a partir do seu diretório *Times* todos arquivos com terminação .txt;

15. Repita a operação e coloque o resultado da listagem em um arquivo *Resultado*;

16. Repita a procura mas agora conte o número das ocorrências de arquivos que começam com I;

17. Liste os arquivos que possuem o texto vermelho. Use o *grep* recursivo;

18. Pesquise e implemente o comando anterior fazendo o comando tratar de forma transparente maiúsculas e minúsculas.

19. Retire o direito de leitura de usuário proprietário de um destes arquivos listados. Aplique um comando para procurar todas as ocorrências de arquivos terminados em *.txt*. Verifique a saída.

20. Para evitar a mensagem de erro que aparece devido ao não permissionamento, redirecione a saída de erro para o /dev/null;

Nota: OBSERVE

O grep pode ser usado com os seguintes caracteres especiais

1. O "^" indica o início de uma linha.
2. O "$" o final de linha.
3. O "." qualquer carater
4. O "$*$" indica zero ou mais ocorrências do caracter anterior;
5. O "[a-b]" qualquer expressão entre a e b.