

Aula 6

6.1 - Redirecionamento e manipulação de entrada e saída

O modelo tradicional computacional determina a existência de um sistema que recebe dados, os processa e posteriormente os devolve para uma saída de análise. Para que essa funcionalidade seja exercida, é necessário que haja uma interface direta de comunicação entre o operador e a máquina.

O conjunto que forma a comunicação com uma máquina é conhecido como um subsistema de entrada e saída. Essas entradas e saídas são, em algum momento do processo, dados; que podem ser interpretados ou processados pelo sistema computacional e posteriormente analisados pelo operador.

Os padrões para o conjunto de entrada e saída são: o teclado, o mouse, o monitor, a impressora, o auto-falante entre outros. Quando se utiliza um sistema Linux, em modo de linha de comando, podemos afirmar que existem alguns itens elementares do sistema de entrada e saída: O terminal e o monitor que poderiam ser classificados como integrantes do sistema de saída padrão e, por outro lado, o teclado e o mouse que representariam as entradas padrões.

Mesmo assim, nada impede que o sistema padrão possa ser alterado de maneira conveniente à aplicação corrente. Para os interpretadores de comando, existem variáveis que convencionam e determinam quais devem ser as saídas e entradas padrões.

6.2 Shell e sua entrada e saída padrão

Quando se está em uma sessão do interpretador de comandos no Linux, todos os comandos executados geram um retorno proveniente desse próprio interpretador. Esse retorno, em geral, é uma resposta da aplicação, que por sua vez pode conter erros ou ainda valores de retorno esperados.

Se um comando é executado, então existem duas possibilidades:

- Saída de valor 1: É o retorno padrão do sistema, chamado de *stdout* e geralmente é a terminal/sessão corrente.

- Saída de valor 2: É o retorno de erro do sistema, chamado de *stderr* e geralmente é o terminal/sessão corrente.

Ainda existe um terceiro valor:

- 0 que identifica a entrada padrão do sistema, também referenciado como *stdin* e geralmente é o teclado.

6.3 Redirecionamento

O redirecionamento de entrada e saída é um recurso muito utilizado em Linux. Em comandos onde há respostas mais complexas ou longas, os retornos podem ser supostamente tratados de maneira mais detalhada através do auxílio e do emprego de redirecionamentos.

Os tipos de redirecionamentos são:

- a) `>` : Redireciona a saída padrão para uma saída desejada (outro comando ou arquivo).
Ex: `cat /var/log/messages > /var/log/messages.copia`
`finger aluno > log_info_aluno.txt`
`ls -la > log_info_aluno.txt` (apaga o conteúdo salvo no comando anterior)
- b) `<`: Redireciona a entrada padrão.
Ex: `cat < arquivotexto.txt`
`sort < /etc/passwd`
- c) `>>`: Redireciona a saída padrão concatenando à saída já existente.
Ex: `cat /etc/group >> arquivo_credenciais.dat`
`cat /etc/passwd >> arquivo_credenciais.dat`
`finger root >> arquivo_credenciais.dat`
- d) `<<`: Redireciona a entrada padrão concatenando-a.
Ex: `<< pare`
- e) `2>` e `1>`: Redireciona a saída de erro e saída padrão.
Ex: `useradd nada 1> /var/log/criauser.log 2>/var/log/criauser_error.log`
Ex: `ls /nada 2>&1`
- f) `2>>` e `1>>`: Redireciona saída de erro e a saída padrão concatenando.
Ex: `useradd nada2 1>> /var/log/criauser.log 2>>/var/log/criauser_error.log`
- g) `|` : Conhecido como pipe, faz a interconexão entre dois comandos, de forma encadear as aplicações. A execução à esquerda encaminha sua saída para o comando à direita do pipe. O resultado final na tela é obtido após a execução do último pipe.
Ex: `ls -la | grep root > /var/arquivos_do_root.txt`
`ls -la | more`
`cat /var/log/messages | more`
`cat /etc/passwd | sort | grep root | wc -l > numero_de_linhas.txt`

6.4 Exercícios

- 1) Qual o conteúdo do arquivo filtro1 após a execução dos seguintes comandos :
 - a) `cat /root/.bashrc > filtro1`
 - b) `cat /home/aluno/.bashrc > filtro1`
 - c) `cat /home/aluno/.bashrc >> filtro1`
 - d) `echo 1 > filtro1`
 - e) `cat /etc/hosts >> filtro1`
 - f) `cat /etc/passwd >> filtro1`
 - g) `cat filtro1 | more`

- 2) Qual a diferença então entre `>` e `>>`

- 3) Execute um comando qualquer salvando saída de erro desse comando no arquivo `log.erro` e saída padrão em `log.executado`

- 4) Execute um comando que vai ordenar as linhas do arquivo `/var/log/mail.log` e filtrar aquelas que contenham a palavra `spool` e salve sua saída em `/var/log/log_de_spool.log` em um único comando.

- 5) O que fazem cada um dos comandos abaixo:
 - a) `ls -al | more`
 - b) `cat /etc/passwd | grep lixo > lixo.tudo`
 - c) `grep tudo /etc/lixo > lixo.tudo`
 - d) `grep nada /etc/lixo >> lixo.tudo`
 - e) `cat < arquivo`
 - f) `sort < /var/log/messages`
 - g) `rmdir /home/aluno 2> remove_diretorio.log`
 - h) `cat << parou`
 - i) `/etc/init.d/networking 2>&1`
 - j) `tail -f /var/log/syslog >> /tmp/lixo 2>&1`