Aula 07

Redirecionamento de conteúdo

&

Agendamento de Tarefas

Prof. Roitier Campos

debian

Parte I

Redirecionamento de conteúdo

Redirecionamentos e Pipe

Redirecionamentos e pipe são recursos utilizados no Linux para que as saídas de um determinado comando possa ser utilzada como entrada em um outro comando.

Sendo assim, podemos visualizar esses dois recursos como operações, similares às matemáticas, que assumem os resultados de uma operação como variável para outras.

>

Redireciona a saída 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 para enviar a saída do comando para o arquivo listagem. Use o comando cat para visualizar o conteúdo do arquivo listagem.

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).

>>

Redireciona a saída de um programa/comando/script para algum dispositivo ou 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. Use o comando cat para visualizar o conteúdo do arquivo listagem.

<

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 <).

<<

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:

cat << final

Nota: Este arquivo será mostrado até que a palavra final seja localizada no inicio da linha final

| (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).

| (pipe)

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.

Mesmo assim a listagem ocupe 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 o "|" e o ">"

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 seria gravada em um arquivo chamado more.

tee

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

Sintaxe:

“comando“ | tee [arquivo]

Exemplo:

ls -la | tee listagem.txt,

Nota: saída do comando será mostrada normalmente na tela e ao mesmo tempo gravada no arquivo listagem.txt.

Parte II

Agendamento de Tarefas

Crontab

O **Crontab**, é um programa do Unix que edita o arquivo onde são especificados os comandos a serem executados e a hora e dia de execução pelo cron, um programa que executa comandos agendados nos sistemas operacionais do tipo Unix (como o Linux ou o MINIX, por exemplo).

O c**ron** se encarregará de verificar a hora e determinar se existe ou não algum programa a ser rodado. Caso exista ele o rodará na hora e data solicitada

Estrutura do Crontab

Para a maioria das tarefas pouco importa a hora que vai ocorrer mas sim a frequência em que ela vai ser executada, como diariamente ou semanalmente. Para isso já existe 4 diretórios especiais, que basta o administrador botar o script lá dentro, eles já serão executados na periodicidade desejada.

- /etc/cron.daily ---- agendamentos diários
- /etc/cron.hourly ----agendamentos a cada hora
- /etc/cron.monthly ---- agendamentos mensais
- /etc/cron.weekly ---- agendamentos semanais

Agendamento específico

Mas caso você mesmo queira fazer um periodo especifico, com hora e tudo mais, basta editar o arquivo /etc/crontab:

```
# m h dom mon dow user command
17* * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
```

/etc/contrab

A cada espaço, se avança um campo e os campos seguem o padrão a seguir:

minuto	hora	diadomês	mês	diadasemana	usuário	comando
m	h	dom	mon	dow	user	command
17	*	*	*	*	root	cd / && run-parts --report /etc/cron.hourly

Observações

Cada campo pode receber valores segundo a tabela a seguir:

Caractere	Exemplo	Siginificado
Hífen	2-4	intervalo de 2 a 4
Virgula	2,4,6,8	os numeros 2, 4, 6 e 8
Barra	*/10	de dez em dez
asterisco	*	todas as opções possíveis

Nota: O campo Mês recebe valores de 1 a 12 e o campo Semana recebe valores de 0 a 7, onde zero é domingo, 1 é segunda-feira, 2 terça-feira e assim por diante.

Atividade Complementar

Simule agendamentos de tarefas semanais, mensais, diárias, etc:

1) backup semana;
2) Cópia do arquivo /var/log a cada 10 minutos;
3) Use sua imaginação e exercite

Sugestão: Crie scripts e coloque nos agendamentos.