



Usar streams, pipes e redirecionamentos

Sumário

Capítulo 1

Usar streams, pipes e redirecionamentos.....	3
1.1. Mãos a obra.....	4

Capítulo 2

Gerenciando	9
2.1. Objetivos.....	9
2.1. Troubleshooting.....	9

Índice de tabelas

Índice de Figuras

Capítulo 1

Usar streams, pipes e redirecionamentos

- Criar scripts que são úteis no dia-a-dia.

1.1. Mãos a obra

While

O while testa continuamente uma expressão, até que ela se torne falsa. Vamos aos exemplos:

```
while [ $variavel = "valor" ]; do
    comando1
    comando2
done
```

Exemplo 1: Verifica se um site esta no ar.

```
#!/bin/bash
clear
echo "Digite o endereço de um site"
read SITE
while ping -c1 $SITE > /dev/null 2>&1
do
    echo "O site $SITE está no ar."
done
```

Exemplo2:Cria quantos arquivos você indicar com uma determinada extensão.

```
#!/bin/sh
clear
echo "Digite o nome do arquivo"
read ARQ
clear
echo "Digite a extensão do arquivo"
read EXT
clear
echo "Digite o numero de arquivos criados"
read NUM
i=1
```

```
while [ $i -le $NUM ]
do
    touch $ARQ$i.$EXT
    i=`expr $i + 1`
done
```

For

O laço for vai substituindo uma variável por um valor, e vai executando os comandos pedidos. Vamos aos exemplos:

```
for variavel in *
do
    comando 1 $variavel
    comando 2 $variavel
done
```

Exemplo1: Compactar todos os arquivos do diretório atual.

```
#!/bin/bash
for i in `ls -l`
do
    tar -cvzf $i.tar.gz $i
done
```

Exemplo2: Apaga todos os arquivos de uma determinada extensão.

```
#!/bin/bash
clear
echo "Digite a extensão dos arquivos que você quer apagar"
read ARQ
for i in *.$ARQ ; do
    rm $i
done
```

if

Com o *if* é possível usar as operações lógicas, que permitem que o script tome decisões. Vamos aos exemplos:

```
If[ condição ]; then
    Faça
Else
    Faça
fi
```

Exemplo 1: Verifica se um determinado usuário está logado no sistema.

```
#!/bin/bash
clear
echo "Digite o nome do usuário"
read USER
if who | grep $USER
then
clear
echo $USER está logado
else
clear
echo $USER não está logado
fi
```

Exemplo 2: Pesquisa uma palavra dentro de um arquivo.

```
#!/bin/bash
clear
echo "Escreva o nome do arquivo e a palavra a ser pesquisada:"
read file word
if grep $word $file > /dev/null
then
clear
echo "A palavra $word existe no arquivo $file."
```

```
fi
```

Case

O case é para comando de fluxo, tal como é o if. Mas enquanto if testa expressões não exatas, o case vai agir de acordo com resultados exatos. Vamos aos exemplos:

```
case $1 in
    parametro1) comando1 ; comando2 ;;
    parametro2) comando3 ; comando4 ;;
    *) echo "Você tem de entrar com um parâmetro válido" ;;
esac
```

Exemplo1: Executar comandos dependendo do usuário digitado.

```
#!/bin/bash
clear
echo "Digite um nome de usuário"
read Usuario
case $Usuario in
    aluno )
        clear ; ls /etc ; cal ; date
        ;;
    root )
        clear ; whoami
        ;;
    *)
        clear
        echo $Usuario não existe
        ;;
esac
```

Exemplo2: Script que trás informações do sistema.

```
#!/bin/bash
clear
echo "Escolha uma opção para informações da maquina (Digite o numero)"
echo "1-Horario do sistema"
echo "2-Tempo que o servidor esta ligado"
echo "3-Quantidade de usuário logados"
echo "4-Sair"
read ESC
case $ESC in
  1)
    H=$(uptime | awk -F" " '{ print $1 }')
    echo "Agora são $H"
    ;;
  2)
    T=$(uptime | awk -F" " '{ print $3 }')
    echo "O sistema esta $T ligado"
    ;;
  3)
    U=$(uptime | awk -F" " '{ print $4 }')
    echo "Existem $U atualmete logados"
    ;;
  4)
    echo "Bye ..."
    ;;
  *)
    echo "Opção invalida"
    ;;
esac
```


Capítulo 2

Gerenciando

2.1. Objetivos

- Troubleshooting criar script que envia email;
- Troubleshooting transformar shell script em binário.

2.1. Troubleshooting



Como eu faço para enviar a cada uma hora, um e-mail com o conteúdo de um arquivo?

Isso é possível usando o comando mail nos scripts e configurar o crontab para automatizar as tarefas.

Primeiro crie um script e aponte o arquivo que sera enviado o conteúdo, e o email de quem vai receber. Como exemplo as ultimas 30 linhas de um arquivo de log sera enviado ao Administrador do sistema.

mail.sh

```
#!/bin/bash
tail -30 /var/log/auth.log >> /tmp/auth.log
mail -s "Logs de autenticação do Sistema" admin@dominio < /tmp/auth.log
rm /tmp/auth.log
```

Para automatizar esta tarefa você precisa configurar o crontab, adicionando no arquivo `/etc/crontab` a seguinte linha:

```
* * * * * root /root/mail.sh
```



Mas como eu posso transformar meu script em binário?

Para compilar seu shell script e transforma-lo em um binário, use o programa shc criado por *Francisco Rosales*. Obtenha o arquivo compactado no site usando o comando `wget`:

```
# wget -c http://www.datsi.fi.upm.es/~frosal/sources/shc-3.8.6.tgz
```

Descompacte o arquivo:

```
# tar -xvzf shc-3.8.6.tgz
```

Copie o binario o shc para `/usr/local/bin`

```
# cp shc-3.8.6/shc /usr/local/bin/
```

Compile seu shell script usando o comando:

```
# shc -v -r -f script.sh
```

Opções de linha de comando:

- v - Modo verbose (mostra o que esta fazendo)
- r - Gera um binário compatível com mais de um sistema.
- f - Opção para o compilador buscar o arquivo.

Copia o arquivo binário para `/bin` (assim todos os usuários terão acesso)

```
# cp script.sh.x /bin/script
```

Acerte as permissões do arquivo para que todos os usuários tenham acesso.

```
# chmod 755 /bin/script
```

Acerta as permissões para todos os usuários.