Laboratório de Sistemas Operacionais II – Aula prática 04 - 21/09/2016

Trabalhando com Loops e Comando Condicional FOR

1. Conceito de **blocos de programa**.

   Chamamos de bloco de programa ou bloco de instruções o agrupamento de instruções compreendido entre um sinal de abre chaves (**{**) e um de fecha chaves (**}**). Por exemplo:

   Criar com **cat > *PS31.ksh*** ou ***vi PS31.ksh***

```
#!/bin/bash
#
# Verifica se o diretório existe. Caso não existe, cria o diretório
# e caminha para ele. Por fim, mostra o caminho com pwd e dá
# mensagem que diretório foi criado
#
[ -d "Diretorio" ] ||
{
        mkdir Diretorio
        cd Diretório
        pwd
        echo "Diretório foi criado com sucesso!!!"
}
```

   Execute com ***./PS31.ksh***

   O operador lógico **||** obriga a execução da instrução seguinte, caso a anterior tenha sido mal sucedida. Então, no caso de não existir o diretório contido na variável ***Diretorio***, o mesmo será criado e, então,

faremos um ***cd*** para dentro dele, e em seguida mostraria o caminho (***pwd***) onde estamos e daremos uma mensagem.

Criar com **cat > *PS32.ksh*** ou ***vi PS32.ksh***

```
#!/bin/bash
#
# Verifica se Usuário está cadastrado no servidor Linux.
# Caso não esteja, gera mensagem de erro
#
grep "$1:" / etc/passwd  ||
{
echo "ERRO: Não existe usuário [$1], cadastrado !!!"
exit 3
}
```

Execute com ***./PS32.ksh fsa44444***

O usuário executou o programa ***PS32.ksh*** informando no parâmetro o código de algum Usuário. Caso o 1º Parâmetro ($1) seja encontrado no início de um registro de ***/etc/passwd***, o programa mostrará os dados do usuário que estão no arquivo ***passwd***, encerrando em seguida. Caso contrário, o bloco de comandos após o operando ***ou*** (**||**) será executado, dando uma mensagem de erro e abortando a execução do programa.

Um bloco de programa também pode ser aberto por um ***do***, por um ***if***, por um ***else*** ou por um ***case*** e fechado por um ***done***, um ***else***, um ***fi*** ou um ***esac***.

2. **Criar** diretório ***Aula4Shell*** (***mkdir Aula4Shell***) e Seguir para ela (***cd Aula4Shell***).

3. **Comando For** – Controle de decisão:

Sintaxe:

```
for    var in valor1, valor2,…., valorN
do
       <comando1>
       <comando2>
       < …….   >
done
```

Exemplo 1: Listar uma seqüência de números múltiplos de 11 (de 11 até 99)

Criar com **cat >** ***PS33.ksh*** ou ***vi PS33.ksh***

```
#!/bin/bash
#
# Lista os números múltiplos de 11 (de 11 até 99)
#
for i  in  1 2 3 4 5 6 7 8 9
do
       echo $i$i
done
```

Execute com ***./PS33.ksh***

4. Comando para gerar uma **seqüência numérica**:
   Sintaxe:

seq ultimo
seq primeiro ultimo
seq primeiro incremento ultimo

No primeiro caso, seria gerada uma seqüência numérica de todos os reais começando em **1** e terminando em **último**.

No segundo caso, seria gerada uma seqüência numérica de todos os reais começando em **primeiro** e terminando em **último**.
No terceiro caso, seria gerada uma seqüência numérica de todos os reais começando em **primeiro** e terminando em **último**, porém os reais viriam espaçados de **incremento**.

Exemplo 2: Faça os testes abaixo no prompt do seu Linux:

seq –s " " 10
1 2 3 4 5 6 7 8 9 10

seq –s " " 10 15
10 11 12 13 14 15

seq –s " " 5 2 15
5 7 9 11 13 15

Obs. está sendo usado a opção –s " " para que o separador entre os números gerados fossem um espaço em branco.

Exemplo 3: Listar uma seqüência de números múltiplos de 11 (de 11 até 99)

Criar com **cat > *PS34.ksh*** ou ***vi PS34.ksh***

```
#!/bin/bash
#
# Lista os números múltiplos de 11 (de 11 até 99) usando seq
#
for  i  in  `seq 9`
do
        echo $i$i
done
```

Execute com ***./PS34.ksh***

Obs. o programa acima ***PS34.ksh*** mostrará o mesmo resultado do programa ***PS33.ksh***. Imagine se você quisesse 30 números, teria que digitá-los um a um no comando for. Desta forma não.

5. **Várias formas** de se usar o comando ***for*** .

Exemplo 4: Duplicando programas na pasta criada

Desenvolva um programa ***PS35.ksh*** em que você faz cópias dos programas existentes começados por PS3, e gere cópias com o mesmo código, colocando um x logo após o número do programa.

Dê um comando ***ls –l*** e veja se seus programas foram copiados corretamente.

Exemplo 5: Listar todos os Programas que estão na lista abaixo

Criar com **cat > *PS35.ksh*** ou ***vi PS35.ksh***

```
#!/bin/bash
#
# Lista programas de uma relação de programas
# Usando parâmetro no comando for
#
for i in PS31.ksh PS32.ksh PS33.ksh PS34.ksh PS35.ksh
do
        ls –l $i
done
```

Execute com ***./PS35.ksh***

Obs. o programa acima ***PS35.ksh*** mostrará via comando ***ls –l*** todos os arquivos que foram relacionados no comando ***for***, um a um.

Exemplo 6: Lista programas do PS31.ksh até PS35.ksh

Criar com **cat > *PS36.ksh*** ou ***vi PS36.ksh***

```
#!/bin/bash
#
# Usando Parâmetros no for
#
for i in PS3[1-5].ksh
do
        ls –l $i
done
```

Execute com ***./PS36.ksh***

Obs. o programa acima ***PS36.ksh*** mostrará o mesmo resultado do programa ***PS35.ksh***. A diferença é que foi colocado um limite mínimo e máximo, e o ***for*** vai usar esta diferença para mostrar, via comando ***ls –l***, todos os arquivos pedidos.

Exemplo 7: Lista programas começados por ***PS3***

Criar com **cat >** ***PS37.ksh*** ou ***vi PS37.ksh***

```
#!/bin/bash
#
# Usando Parâmetros no for
#
for i in `echo PS3*.ksh`
do
        ls –l $i
done
```

Execute com ***./PS37.ksh***

Obs. o programa acima ***PS37.ksh*** mostrará todos os programas que começam por ***PS3***.

6. ***Usando for*** sem os parâmetros.

Exemplo 8: Passando parâmetros para o for

Criar com **cat >** ***PS38.ksh*** ou ***vi PS38.ksh***

```
#!/bin/bash
#
# Não usando parâmetros no for
#
echo O programa $0 Recebeu $# Parâmetros
echo –n "Que são: "
for i
do
        echo –n "$i "
done
```

Execute com ***./PS38.ksh alfa beta gama delta teta iota***

Exemplo 9: Modificando palavras de maiúsculas para minúsculas

****** ATENÇÃO *** Este programa vai mudar de maiúsculo para minúsculo e vice versa todos os arquivos da sua pasta (diretório).***
Tenha certeza de que você esteja trabalhando apenas na pasta Aula4, conforme foi pedido no início desta sessão.

Criar com **cat > *PS39.ksh*** ou ***vi PS39.ksh***

```
#!/bin/bash
#
# Passando maiúscula para minúscula
#
for Maiusc in *
do
    Minusc=$(echo $Maiusc | tr A-Z a-z)
   mv $Maiusc $Minusc 2> /dev/null || echo $Minusc não renomeado
done
```

Execute com ***./PS39.ksh***

Obs.: Não tem espaço após o número 2 do comando ***mv*** acima.

O (*) do ***for*** lista o nome de todos os arquivos do diretório corrente (com ou sem maiúsculas), passando-os para um ***tr*** que transformará, se for o caso, as letras maiúsculas em minúsculas. Caso o nome já estivesse em minúsculas, ou caso existisse um arquivo anterior com o mesmo nome em letras minúsculas, o erro proveniente do ***mv*** seria desviado para ***/dev/null*** e em seguida mandaria a mensagem que o arquivo não fora renomeado.

Para executar o programa, coloque permissão de acesso e execute-o

./PS39.ksh

"Não se permite, nem ao homem mais justo, que seja o juiz de sua própria causa."

Blaise Pascal.