Laboratório de Sistemas Operacionais II – Aula prática 01- 09/08/2016

Trabalhando com variáveis do Shell

1. Introdução à **Programação do Shell**. Explicar.

2. Fazer o ***Primeiro Programa Shell***.

Crie uma pasta e chame de ***Aula01Shell***. Faça os exercícios nela.

Você pode usar o comando ***cat*** ou ***vi*** para editar seu texto. Sugiro que vá se acostumando a usar o ***vi***, porque se você errar é fácil fazer a correção.

cat > modelo1.ksh ***ou*** ***vi modelo1.ksh***

```
# Este script posiciona o usuário em seu diretório home e Lista o nome e o
# conteúdo deste diretório
#
cd
pwd
ls –l
```

para executar o programa acima, fazer: ***./modelo1.ksh*** (Permissão de acesso ***u+x***).

3. Fazer o ***Segundo Programa Shell***.

Criar um cadastro de telefones ordenado por nome:

cat > telefones.txt ***ou*** ***vi telefones.txt***

Lázaro Pinto	11	4444-1122
Sérgio Melo	17	5555-1122
Paulo Borges	21	3344-1122
Antonio Souza	35	8123-9988
Maria Sá	41	7311-5566

Obs.: Ao terminar de digitar a última linha, não se esqueça de dar ***ctrl+C*** se estiver usando o ***cat***, porque você pode perder esta linha.
Esta lista de telefones será utilizada no script abaixo.

cat > DuLoren.ksh ou vi DuLoren.ksh

```
#
# Meu Segundo script em Shell conta quantos telefones tem na minha lista
# e mostra ordenado por nome esses telefones
#
echo Eu tenho `cat telefones.txt | wc –l` telefones cadastrados
echo "Que são:"
cat telefones.txt | sort
```

Obs.:
(1) Quando você tem um comando dentro de outro, por exemplo, **cat** dentro de **echo**, o apóstrofe correto é feito com ***shift*** + ***acento craseado*** e ***<enter>*** antes do ***cat*** e após completar o ***cat***.
(2) O comando ***cat telefones.txt*** pega todo arquivo e manda para o ***wc*** porque ele encontrou um ***pipe***. O comando ***wc*** conta quantas linhas tem e então entrega de volta para o comando ***echo***.

4. Treinar o uso de ***variáveis locais*** usando o comando ***set*** e ***typeset***. Veja a diferença entre ambos e o que cada um oferece.

5. Trabalhando com ***variáveis locais e constantes***:

cat > ps01.ksh ou vi ps01.ksh

```
#!/bin/bash
#
# Este programa trabalha com variáveis locais e constantes
#
CLUBE="Flamengo"
printf "Sou torcedor do $CLUBE !"
```

Obs.:

(1) Sempre que você usar atribuições à variáveis, não coloque espaço nem antes nem depois do sinal de atribuição (**=**).

(2) O primeiro comando do seu programa deverá ser sempre ***#!/bin/bash***, porque ele mostra ao linux onde está a biblioteca especial do ***Shell***. Caso este comando não esteja presente, comandos como ***printf***, ***read*** e outros não serão reconhecidos.

6. Trabalhando com ***variáveis locais e entrada de dados***:

cat > ps02.ksh ou vi ps02.ksh

```
#!/bin/bash
#
# Este programa trabalha com variáveis locais e entrada de dados
#
typeset  CLUBE=" "
printf  "Informe qual é o seu Clube."
read  CLUBE
printf  "Sou torcedor do $CLUBE !"
```

Obs.:

(1) O comando ***typeset*** definiu uma variável caractere. Este comando até não precisaria estar definido. Na sua ausência, o linux autodefine variáveis tipo ***string***.

(2) O comando ***read*** causa uma pausa no programa (sai de execução e entra em estado de espera) aguardando que o usuário digite algo e dê <***enter***>.

7. Trabalhando com ***variáveis locais e variáveis do sistema.***

cat > ps03.ksh ou vi ps03.ksh

```
#!/bin/bash
#
# Este programa trabalha com variáveis locais e variáveis do sistema
#
mensagem=”Boa Noite!”
disciplina=”Programação Shell”
printf “ ”
printf “$mensagem, $USER !”
printf “ ”
printf “ Bem Vindo à”
printf “ $disciplina !”
printf “ ”
```

Obs.:

(1) Se você der um comando ***typeset*** no seu ***prompt*** você verá todas as variáveis do sistema. No nosso caso, usamos a variável ***USER*** (sempre maiúscula) que armazena o nome do usuário que logou-se no seu terminal.

(2) O comando ***disciplina=”Programação Shell”*** armazena na variável disciplina uma constante, conforme vimos anteriormente.

8. Trabalhando com ***variáveis locais string, locais numérica e do sistema***

cat > ps04.ksh ou vi ps04.ksh

```
#!/bin/bash
#
# Este programa trabalha com variáveis locais string, locais numérica e
# variáveis do sistema
#
mensagem=”Boa Noite!”
```

disciplina="Sistema Operacional"
typeset –i anos=0 formatura=0
printf " "
printf "$mensagem, $USER !"
printf "Informe quantos anos você tem:"
read anos
formatura=$anos+2
printf " "
printf " Bem Vindo à"
printf " $disciplina !"
printf " "
printf "Quando formado, você $USER terá $formatura anos"
printf " "

Obs.:

(3) O comando ***typeset –i anos=0 formatura=0*** definiu duas variáveis numéricas inteiras inicializadas em 0.

(4) O comando ***formatura=$anos+2*** armazena na variável formatura a operação de soma entre a variável anos e a constante 2. Veja que o $ antes da variável somente é usado quando você for referenciar-se à ela. Quando ela for receptora de algo, não é necessário.

9. Trabalhando com ***variáveis locais e globais***

cat > ps05.ksh ou vi ps05.ksh

```
#!/bin/bash
#
# Este programa trabalha com variáveis locais e globais
#
```

typeset LOCAL="Var Local"***
typeset –x GLOBAL="Var Global"***
printf " "

printf "*** Estamos no programa ps05.ksh **********"***
printf " "
printf "A variável LOCAL tem como conteúdo: $LOCAL"
printf "A variável GLOBAL tem como conteúdo: $GLOBAL"
./ps05a.ksh

cat > ps05a.ksh ou vi ps05a.ksh

```
#!/bin/bash
#
# Este programa trabalha com variáveis locais e globais recebidas do
# programa ps05.ksh
#
printf " "
printf "****** Estamos no programa ps05a.ksh **********"
printf " "
printf "A variável LOCAL tem como conteúdo: $LOCAL"
printf "A variável GLOBAL tem como conteúdo: $GLOBAL"
```

Obs.:

(1) Não se esqueça em de habilitar a execução dos dois programas usando o ***chmod***.

(2) Você vai executar apenas o programa ps05.ksh. Ele, internamente, vai chamar e executar o programa ps05a.ksh.

(3) Veja a diferença do comando ***typeset***. Um informa ***–x*** como opção e outro não. O ***–x*** indica que a variável será ***global***, ou seja, pode ser referenciada em todos os programas que você chamar.

(4) O que se espera nestes dois programas é que: no ps05.ksh você deve ver o conteúdo das duas variáveis. No ps05a.ksh você verá apenas o conteúdo da variável ***global***.

10.Trabalhando com ***parâmetros***:

Treinando um novo tipo de busca em arquivo:

Procurar nomes no arquivo de telefones:

- Usar mesmo arquivo de ***telefones.txt***

- Procurar por um certo Nome : ***grep Paulo telefones.txt***
- Procurar por outro Nome: ***grep "Paulo Borges" telefones.txt***

Veja como se comportam os comandos acima.

Agora crie um programa para testar essas buscas.

cat > ps06.ksh ou vi ps06.ksh

```
#!/bin/bash
#
# Este programa trabalha com parâmetros
#
grep $1 telefones.txt
```

Obs.:

(1) Execute o programa desse tipo informando sempre uma variável. Tente fazer assim: ***./ps06.ksh Cardoso***. Neste caso, o programa nada responderá, porque Cardoso não foi encontrado no arquivo ***telefones.txt***.

(2) Execute ***./ps06.ksh Maria***. O programa vai mostrar o telefone e os dados de Maria.

(3) Se você tentar procura um nome composto, por exemplo "Maria Sá", então o comando ***grep*** deverá ser ***grep "$1" telefones.txt***.

11.***Desafio:*** Desenvolver um exercício Extra.

Desenvolva um programa ***PS07.ksh – Mensagem***
- Solte a seguinte mensagem:

Identificação (Antes peça para ser digitado que tipo de tratamento se quer dar à pessoa – Sr, Dr, Ilmo, etc). ***Fulano*** (Substitua o Fulano pelo Usuário), você está localizado no diretório ***dd*** (Diretório base, o diretório do usuário) e está logado deste as ***nn*** horas (Hora em que ele se logou no sistema).

Obs.:
As palavras em negrito são variáveis locais ou do sistema.

"Um otimista vê uma oportunidade em cada calamidade. Um pessimista vê uma calamidade em cada oportunidade."

Winston Churchill