

Exercício Prático #6 – Shell Script

- 1) Escreva um script Shell para:
 - a) Saber informações a respeito do sistema, tais como: horário, ocupação do disco e os usuários que estão “logados”.
 - b) Melhore o script anterior, exibindo mensagens na tela para anunciar cada comando antes de executá-lo.
 - c) Para o script ficar mais completo, introduzir uma interação mínima com o usuário, pedindo uma confirmação antes de executar os comandos.
 - d) Para facilitar manutenções futuras, deixe o código visivelmente mais agradável e coloque comentários que esclareçam as atividades a serem efetuadas.
- 2) Elaborar um script que pede ao usuário para digitar um nome de arquivo e testa se o arquivo existe. Se sim, diz se é um arquivo ou um diretório.
- 3) Para saber se uma variável é maior ou menor do que um valor qualquer inteiro (por ex.: 10) e mostrar uma mensagem na tela podemos escrever o seguinte script:

```
if test "$VARIABEL" -gt 10
then
    echo "é maior que 10"
else
    echo "é menor que 10"
fi
```

OU

```
if [ "$VARIABEL" -gt 10 ]
then
    echo "é maior que 10"
else
    echo "é menor que 10"
fi
```

Escreva um script que, dados dois números como parâmetro, mostra a relação entre eles. Uma possível saída seria:

```
prompt$ ./relacao.sh 3 5
3 é menor que 5
prompt$ ./relacao.sh 5 3
5 é maior que 3
prompt$ ./relacao.sh 5 5
5 é igual a 5
prompt$
```

- 4) Escreva um script que recebe um número como parâmetro e o diminui até chegar a zero, mostrando na tela cada passo em uma mesma linha. Uma possível saída seria:

```
prompt$ ./zerador.sh 5
5 4 3 2 1 0
prompt$ ./zerador.sh 10
10 9 8 7 6 5 4 3 2 1 0
prompt$
```

- 5) Escreva um script que receba duas palavras como parâmetro e verifica se a primeira palavra está contida na segunda. O script mostra uma mensagem informativa em caso de sucesso, caso contrário não exibe nada.

Um exemplo de saída seria:

```
prompt$ ./substring.sh ana banana
ana está contida em banana
prompt$ ./substring.sh banana maria
prompt$ ./substring.sh banana
prompt$ ./substring.sh
prompt$
```

- 6) Escreva um script que mostra na tela todos os parâmetros recebidos na linha de comando “grudados”, ou seja, como uma única palavra.

Dica: investigue o comando *tr*, que traduz uma cadeia de caracteres em outra (*translate*).

Um exemplo de saída seria:

```
prompt$ ./juntaparms.sh a b c d e f verde azul
abcdefverdeazul
prompt$
```

- 7) Escreva um script que mostra o usuário e o nome completo de cada usuário do sistema (campos 1 e 5), separados por TAB. Obter as informações no arquivo */etc/passwd*.

Uma possível saída:

```
prompt$ ./users.sh
ftp FTP User
nobody Nobody
named Domain name server
xfs X Font Server
mysql MySQL server
prompt$
```

- 8) Escreva um script que mostra os shells de cada usuário do sistema (último campo), não apresentando linhas repetidas. Obter as informações no arquivo */etc/passwd*.

Dica: utilize o comando *uniq*.

Uma possível saída:

```
prompt$ ./shells.sh
/bin/bash
/bin/false
/bin/sync
/sbin/halt
/sbin/shutdown
prompt$
```

9)Escreva um script que mostra todos os parâmetros recebidos na linha de comando, contando-os.

Uma possível saída:

```
prompt$ ./parametros.sh a b c d e f
Parâmetro 1: a
Parâmetro 2: b
Parâmetro 3: c
Parâmetro 4: d
Parâmetro 5: e
Parâmetro 6: f
prompt$
```