Lista de Exercícios 08

Estruturas de Fluxo Condicionais

Exercício 01) Escreva um *shell script* que recebe um arquivo como argumento em linha de comando e verifica se existe ou não. Caso exista, verifique se é um arquivo, um diretório ou desconhecido mostrando quais as permissões que o **usuário atual** (**não o dono do arquivo**) possui sobre ele. Caso não exista, imprima que é inválido. **Cuidado**, essas informações não são obtidas com comandos como `ls`, `stat` ou `file`.

```
$ ./checkfile
Usar: ./checkfile [Arquivo]
$ ./checkfile /etc/passwd
/etc/passwd: arquivo (r--)
$ ./checkfile /tmp
/tmp: diretório (rwx)
$ ./checkfile /dev/random
/dev/random: desconhecido (rw-)
$ ./checkfile /private
/private: inválido
```

Exercício 02) Escreva um *shell script* que recebe três números inteiros pela **entrada padrão** (e não via argumentos em linha de comando) e mostre qual o maior deles.

```
$ ./higher
Primeiro número: 5
Segundo número: 12
Terceiro número: 8

O maior número é o 12
```

Exercício 03) Escreva um *shell script* para cumprimentar um usuário com `Bom dia`, `Boa tarde` ou `Boa noite` dependendo do horário do sistema. A saudação deve incluir o nome completo do usuário atualmente logado. Dica: use o comando `whoami` para obter o usuário atual e depois consulte o arquivo `/etc/passwd` na quinta coluna para obter o seu nome completo. As horas do dia são divididas da seguinte maneira:

- **Dia**: até 12:00
- **Tarde**: a partir de 12:00 e até 18:00
- **Noite**: a partir das 18:00

Exemplo de utilização do *shell script*.

```
$ date
Sun Nov 30 12:40:01 BRST 2014
$ ./greeting
Boa tarde João do Caminhão, seja bem vindo.
```

Exercício 04) O comando `expr` é capaz de realizar cálculos com expressões aritméticas inteiras. Reescreva um programa similar em *shell script* para calcular uma expressão com apenas um operador com dois números em ponto flutuante e imprima o resultado em ponto flutuante. Os operadores suportados com seus respectivos símbolos são: adição (`+`), subtração (`-`), multiplicação (`x`), e divisão (`/`). **Você deve usar obrigatoriamente a estrutura** `case` **como controle de fluxo condicional.** O programa deve ler os dois números e o operador via parâmetros em linha de comando. Caso não sejam especificados os três parâmetros, deve-se mostrar uma mensagem de ajuda. Se o operador não for um dos quatro suportados deve-se mostrar uma mensagem de erro. Dica: use o comando `bc` (*basic calculator*) para fazer as operações em ponto-flutuante. Exemplo de uso do comando `bc`.

```
$ echo "22 / 7" | bc -l
3.14285714285714285714
```

Exemplo de uso do *shell script*.

```
$ ./calc
Usar: ./calc [Número] [Operador] [Número]
$ ./calc 22 / 7
3.14285714285714285714
$ ./calc 12 - 7.2
4.8
$ ./calc 5 % 4
%: operação inválida
```