Lista de Exercícios 07

Parâmetros e Variáveis

Exercício 01) Um invasor entrou em um sistema e mexeu nos arquivos de inicialização da *shell* (ex.: `/etc/profile`), alterando variáveis especiais causando os mais diversos efeitos colaterais. Para cada um dos problemas a seguir, indique **qual variável** foi afetada e qual o provável **conteúdo** dela.

a) Primeiro problema.

```
$ cat /etc/passwd
-bash: cat: command not found
$ ls /var
-bash: ls: command not found
$ ps aux
-bash: ps: command not found
```

b) Segundo problema.

```
$ /usr/bin/whoami
rimsa
$ /bin/cat /etc/passwd | /usr/bin/grep rimsa
rimsa:*:1000:1000:Andrei:/home/rimsa:/bin/bash
$ /usr/bin/cd ~
$ /bin/pwd
/tmp
```

c) Terceiro problema.

```
$ /usr/bin/cd /usr/local/bin
$ /bin/pwd
/usr/local/bin
$ /bin/echo $(/bin/pwd)
 usr local bin
```

Exercício 02) Escreva *shell scripts* para realizar as tarefas a seguir.

a) Receba `Nome e Sobrenome` via argumentos e imprimir no formato `Sobrenome, Nome` como em passagens de avião.

```
$ ./ticket Paulo Silva
Silva, Paulo
```

```
$ ./ticket "Ana Maria" Braga
Braga, Ana Maria
```

b) Imprima a hora atual, a quanto tempo o servidor está online e quantos usuários estão logados nesse momento. Dica: use o comando `uptime` para obter essas informações. Detalhe: você não pode usar *pipes* ou jogar a saída para um arquivo e você pode executar o comando `uptime` somente uma vez.

```
$ uptime
12:12  up  2:59, 2 users, load averages: 1.39 1.46 1.54
$ ./show_stats
Current time: 12:12
Uptime: 2:59
Users: 2
```

c) Simule o rolamento de dados de seis faces. A cada rodada deve-se sortear um valor entre `1` e `6` e imprimi-lo. Detalhe: você não pode executar nenhum programa auxiliar.

```
$ ./dice.sh
2
```

```
$ ./dice.sh
6
```

```
$ ./dice.sh
3
```

Exercício 03) Mostre o comando para alterar os seguintes *prompts* de usuário de acordo com o formato especificado:

a) O *prompt* primário no formato `[user@hostname:currentdir]$`

```
$ ____________________________________________________
[andrei@cefetmg.br:/tmp]$ _
```

b) O *prompt* secundário no formato `next line>`

```
$ ____________________________________________________
$ echo "Hello
next line> World"
Hello
World
```

Exercício 04) Usando uma **expansão** mostre a **saída** para cada um dos comandos abaixo de forma que o usuário possa perceber se ele foi executado com sucesso ou não.

a) Ler o arquivo de usuário.

```
$ cat /etc/passwd 1>/dev/null 2>&1
$ echo ___
___
```

b) Ler o arquivo de senhas.

```
$ cat /etc/shadow 1>/dev/null 2>&1
$ echo ___
___
```

c) Listar o conteúdo do diretório de configurações.

```
$ ls -l /etc/ 1>/dev/null 2>&1
$ echo ___
___
```

d) Listar o conteúdo do diretório do usuário `root`.

```
$ ls -l ~root/ 1>/dev/null 2>&1
$ echo ___
___
```

e) Mostrar a saída de uma expansão de comando inválido.

```
$ echo -n `unme`
-bash: unme: command not found
$ echo ___
___
```

Exercício 05) Crie *shell scripts* para cada uma das situações a seguir. Não é permitido o uso de qualquer estrutura de fluxo condicional, como `if`.

a) Imprima a idade do usuário recebida via parâmetros em linha de comando. Caso a idade não seja informada, o programa deve assumir o valor padrão `0`.

```
$ ./show_age 10
Você tem 10 anos.
```

```
$ ./show_age
Você tem 0 anos.
```

b) Reescreva o *shell script* acima para abortar com uma mensagem de erro se o usuário não informar sua idade.

```
$ ./show_age 10
Você tem 10 anos.
```

```
$ ./show_age
./show_age: line 3: 1: Idade inválida.
```