

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
```

A variável alterada foi a \$PATH, que é o caminho relativo ou absoluto em que a shell procura os programas, como foi alterado a shell não encontra os programas para executá-los.

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
```

A variável alterada foi a \$HOME, que apresenta o caminho do diretório do usuário que é expandida pelo ~. O valor dela foi alterada de /home/rimsa para /tmp.

c) Terceiro problema.

```
$ /usr/bin/cd /usr/local/bin
$ /bin/pwd
/usr/local/bin
$ /bin/echo ${/bin/pwd}
usr local bin
```

A variável alterada foi o \$IFS que armazena o carácter que sera usado como separador de argumentos na linha de comando foi substituído por / , então o echo recebeu como entrada 3 valores usr, local e bin separados.

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

```
#!/bin/bash
```

```
echo "$2, $1"
```

- 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* e 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
```

```
#!/bin/bash
```

```
set $(uptime)  
echo "Current time: $1"  
echo "Uptime: ${3/,/}"  
echo "Users $4"
```

- 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      $ ./dice.sh      $ ./dice.sh  
2                  6                  3
```

```
#!/bin/bash
```

```
echo $((($RANDOM % 6)+1))
```

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]\$

```
$ PS1='[\u@\H:\w]\$ '
[andreia@cefetmg.br:/tmp]$ _
```

- b) O *prompt* secundário no formato next_line>

```
$ PS2="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 $?
0
```

- b) Ler o arquivo de senhas.

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

- c) Listar o conteúdo do diretório de configurações.

```
$ ls -l /etc/ 1>/dev/null 2>&1
$ echo $?
0
```

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

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

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

```
$ echo -n `unme`
-bash: unme: command not found
$ echo $?
0
```

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          $ ./show_age  
Você tem 10 anos.        Você tem 0 anos.
```

```
#!/bin/bash  
echo "Você tem ${1:-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          $ ./show_age  
Você tem 10 anos.        ./show_age: line 3: 1: Idade inválida.
```

```
#!/bin/bash  
echo "Você tem ${1:?Idade inválida.} anos."
```