

Programação em Shell (Exercícios)

Princípios da Computação
2008/2009

Novembro de 2006

Luis Nogueira
Luis Lino Ferreira
Berta Batista
Bertil Marques
Maria João Viamonte
Paulo Ferreira

Sugestões e participações de erros para:

llf@dei.isep.ipp.pt

1 Programação em Bash – Exercícios

1. Crie um script que apresente no ecrã a frase “Estes são os ficheiros da área de trabalho do utilizador X na data Y:”, substituindo X pelo nome do utilizador corrente e Y pela data actual do sistema, seguida de uma listagem dos ficheiros existentes na área de trabalho do utilizador. Use as variáveis de ambiente adequadas
2. Altere o script anterior permitindo:
 - a. Perguntar ao utilizador qual o nome a apresentar no ecrã, bem como o directório a listar
 - b. Receber o nome e o directório como parâmetros
3. Crie um script que, recebendo como parâmetro o nome de um ficheiro, mostre no ecrã o seu conteúdo caso ele exista, ou a frase “Erro: ficheiro não encontrado” se o ficheiro não existir
4. Crie um script com o nome *chkdir* que apresenta informação sobre o directório actual:
 - a. Conta o número de ficheiros e subdirectórios existentes
 - b. Para cada um dos ficheiros, conta o número de palavras e linhas.

Obs: o comando *wc* se usado na forma “*wc -l ficheiro*” para além de devolver o nº de linhas do ficheiro mostra também o nome desse ficheiro. Para obter apenas o nº use o comando na forma “*wc -l < ficheiro*”

5. Crie um script com o nome *chkspc* que
 - a. Usando o comando *du* mostra o espaço ocupado em disco pelo directório actual
 - b. Liste o nome e conte o nº dos ficheiros com mais de 1 MB
6. Crie um script com o nome *daily_backup* que
 - a. Cria um directório *backups* (caso ainda não exista) na sua área
 - b. Cria um subdirectório de *backups* com a data do dia actual (*date +%F*)
 - c. Copie todos os seus ficheiros existentes na sua área e subdirectórios (com a excepção do subdirectório backups) com cada uma das extensões passadas por parâmetro

Obs: Para excluir o directório *backups* do caminho de pesquisa do comando *find* deve ser usada a opção *-prune*. No exemplo seguinte *-o* tem o valor OR

```
find $HOME -path "$HOME/backups" -prune -o -name ...
```

7. Crie um script que encontre todos os seus ficheiros com permissões de escrita para os restantes utilizadores do sistema e retire essas permissões

8. Considere um excerto do ficheiro */etc/passwd*

```
joao:x:100:100:Joao Silva:/home/joao:/bin/bash
joana:x:101:101:Joana Santos:/home/joana:/bin/bash
jose:x:102:102:Jose Silva:/home/jose:/bin/bash
...
```

- a. Crie um script que receba um nome de um utilizador como argumento e apresente no ecrã se o utilizador existe ou não no sistema
- b. Altere o script anterior para no caso do utilizador existir, apresentar no ecrã o seu nome completo, caminho para a área de trabalho e tipo de shell usada por esse utilizador
- c. Crie um script que apresente uma lista dos utilizadores que possuem cada um dos apelidos passados como argumento

9. Crie um script que forneça ao utilizador 3 opções:

- a. (V) Ver o conteúdo de um ficheiro
- b. (U) Listar o nome do último ficheiro acedido
- c. (C) Copiar um ficheiro para um determinado directório
- d. (M) Mudar de directório corrente
- e. (D) Listar o nome do directório corrente
- f. (E) Encontrar e mostrar o conteúdo do ficheiro com mais linhas do directório corrente
- g. (S) Sair

10. Reorganize o código do script anterior usando funções. Experimente passar valores por parâmetro às funções, usar variáveis globais, retornar um inteiro como valor de saída da função, etc.