

L:EIC / SO2122:

A Shell de Comandos do Unix (Linux)

Para a resolução destes exercícios sugere-se que leia e faça os exercícios desta excelente introdução à shell Bash da autoria de Ryan Chadwick. Os comandos (e respectivas opções), operadores e símbolos especiais da shell que deve conhecer e saber como funcionam são:

caminhos absolutos e relativos:

- * directórios (/, . e ..)

símbolos especiais da shell:

- * globs (^, ?, *, [], {})
- * redirecionamento de I/O (>, >>, <, |)
- * desligar processo da shell (&)

comandos básicos:

- * navegação (pwd, cd, tree)
- * criação/remoção de ficheiros/directórios (touch, mkdir, rmdir, rm, rm -rf)
- * copiar e mover ficheiros/directórios (cp, cp -a, cp -r, mv)
- * manipulação de ficheiros (cat, more, less, head, tail, chmod, file)
- * gestão de processos (ps, ps -aux, ps -u, jobs, top, kill -9)
- * pesquisa de ficheiros (find, find -size -name -print, ls, ls -l, ls -lR)
- * pesquisa de comandos e manual (which, man)
- * informação em ficheiros (grep, grep -v, cut, cut -d -f, sort)
- * edição de informação em ficheiros (sed s/X/Y/g ou s/X/Y/n, n=1,2...)
- * redirecionamento de input/output e pipes ("<", ">", ">>", "|")
- * arquivo (tar, zip, gzip)

Abra um terminal de shell no seu computador e resolva os seguintes exercícios. Para cada um deles deve primeiro tentar antever o resultado com base no que sabe sobre a shell Bash e os comandos em causa. Confirme depois o seu palpite executando as sequências apresentadas.

1. Em que directório se encontra depois de executar cada um dos seguintes comandos? Verifique o seu palpite executando o comando `pwd`.

```
$ cd ~  
$ cd
```

2. Use o comando `mkdir` para criar uma sub-árvore de directórios com a forma seguinte:

```
dir1  
|-- dir2  
|   |-- dir4  
|   |   |-- dir6  
|   |-- dir5  
|-- dir3
```

3. A partir do directório inicial, em que directório estaria após executar os comandos que se seguem?

```
$ tree dir1  
dir1  
|-- dir2  
|   |-- dir4  
|   |   |-- dir6  
|   |-- dir5  
|-- dir3  
$ cd dir1/dir2/dir4/dir6/../../..
```

4. Use o comando `touch` para criar ficheiros vazios com os nomes seguintes dentro da subárvore que acabou de criar.

```
dir1  
|-- dir2  
|   |-- dir4  
|   |   |-- g22.doc  
|   |-- f1.txt  
|   |-- h22.txt  
|       |-- g368.pdf  
|-- dir3  
    |-- f3a.txt  
    |-- g56.doc  
    |-- g3x.pdf
```

5. O que imprimem os dois últimos comandos desta sequência?

```
$ tree dir1  
dir1  
|-- dir2
```

```

|   |-- dir4
|   |   |-- g22.doc
|   |   |-- f1.txt
|   |   |-- h22.txt
|   |   |-- g368.pdf
|-- dir3
|   |-- f3a.txt
|   |-- g56.doc
|   |-- g3x.pdf
$ find dir1/dir2 -name "[fg][35][4-7a-z].txt" -print
$ find dir1/dir3 -name "[fg][35][4-7a-z].txt" -print

```

6. Qual a estrutura da subárvore com raiz no directório dir1 depois de executado o último comando desta sequência?

```

$ tree dir1
dir1
|-- dir2
|   |-- dir4
|   |   |-- g22.doc
|   |   |-- f1.txt
|   |   |-- h22.txt
|   |   |-- g368.pdf
|-- dir3
|   |-- f3a.txt
|   |-- g56.doc
|   |-- g3x.pdf
$ rm -rf dir1/dir2

```

7. Assuma agora que em dir3 tem o seguinte cenário. Escreva (em octal) as permissões dos 3 ficheiros, os donos dos ficheiros as suas datas de criação e os seus tamanhos em bytes.

```

$ ls
f3a.txt g3x.pdf j52.docx
$ ls -l
total 28712
-rw-r--r--  1 lblopes  staff      1412 Dec 29 15:43 f3a.txt
-rw-r--r--@ 1 lblopes  staff 13923695 Dec 29 15:44 g3x.pdf
-rw-r--r--@ 1 lblopes  staff   13793 Dec 29 15:47 j52.docx

```

8. Que permissões têm "user", "group" e "others" sobre um ficheiro doit depois de executados cada um dos comandos seguintes?

```

$ chmod 755 doit
$ chmod u-wx doit

```

```
$ chmod go-rx doit
$ chmod 644 doit
```

9. O que imprimem os últimos dois comandos desta sequência?

```
$ cat > trees.txt
pine:253:221:1.2
oak:144:123:0.9
birch:92:83:1.6
yew:65:63:4.3
alder:12:5:9.6
^D
$ cat trees.txt | cut -d ':' -f 1,4
$ cat trees.txt | cut -d ':' -f 1,4 | sort
```

10. Considere o seguinte ficheiro com uma citação escrita numa só linha (sem mudanças de linha). Qual o “output” dos últimos três comandos desta sequência? Explique.

```
$ cat > q1.txt
Three Rings for the Elven-kings under the sky,
Seven for the dwarf-lords in their halls of stone,
Nine for Mortal Men doomed to die,
One for the Dark Lord on his dark throne,
In the Land of Mordor where the Shadows lie.
One Ring to rule them all, One Ring to find them,
One Ring to bring them all and in the darkness bind them
In the Land of Mordor where the Shadows lie.
^D
$ cat q1.txt | sed 's/Ring/Sword/g' > q2.txt
$ cat q2.txt | grep -v Mordor
$ wc -l q2.txt
```

11. Qual o resultado dos comandos diff na seguinte sequência?

```
$ cat > f1.txt
I don't know half of you
half as well as I should like;
and I like less than half of you
half as well as you deserve
^D
$ sed 's/half/two\ thirds/g' < f1.txt > f2.txt
$ diff f1.txt f2.txt
$ sed 's/like/do\ not\ like/g' < f1.txt > f2.txt
$ diff f1.txt f2.txt
```

12. Qual o output do último comando desta sequência? Explique.

```
$ cat > numbers1.txt
```

```
66
```

```
43
```

```
77
```

```
22
```

```
91
```

```
^D
```

```
$ cat > words1.txt
```

```
Rivendell
```

```
Gondolin
```

```
Lothlorien
```

```
Angband
```

```
Gondor
```

```
Moria
```

```
Shire
```

```
$ sort -n < numbers1.txt > numbers2.txt
```

```
$ sort -d < words.txt > words2.txt
```

13. O seguinte comando permite conhecer com uma boa aproximação algo sobre os processos actualmente geridos pelo sistema operativo. O quê?

```
$ ps -A | wc -l
```

14. Qual o resultado da execução do último comando desta sequência?

```
$ emacs &
```

```
$ emacs &
```

```
$ emacs &
```

```
$ ps -A | grep emacs
```

```
3577 ttys001    0:00.02 /usr/local/bin/emacs
```

```
3578 ttys001    0:00.02 /usr/local/bin/emacs
```

```
3579 ttys001    0:00.02 /usr/local/bin/emacs
```

```
3583 ttys001    0:00.00 grep emacs
```

```
$ kill -9 3577 3578 3579
```