Ficha Prática nº1

Programação III

Paul Crocker e Simão Melo de Sousa
Departamento de Matemática/Informática da UBI

19 de Fevereiro de 2001

Sessão Unix

Exercício 1: (*Abertura de uma sessão Unix*)
Abra uma sessão Unix indicando ao seu terminal o seu username e a sua password.

Resolução:

Exercício 2: (*Senha*)
Mude a sua password.

Resolução:

Exercício 3: (*Who, Rusers*)
Quantos utilizadores estão utilizando a sua máquina? a rede?

Resolução:

Exercício 4: (*Quem sou eu?*)
Como verificar quem é o proprietário da sessão Unix activa?

Resolução:

Exercício 5: (*ls*)
Qual é o contéudo do directório de trabalho?

Resolução:

Exercício 6: (*Agenda*)
qual é a hora e a data actual usada no sistema? Quais são as opções do commando usado? exemplifique o uso do comando `cal`.

Resolução:

Exercício 7: (*Agenda Bis*)
Peça ao computador que verifique o dia em que nasceu. Qual é a particularidade do ano 1752?

Resolução:

Exercício 8: (*finger*)
Qual é o estado e a informação pública que disponibiliza? Como mudar essas informações?

Resolução:

Exercício 9: (*Procura de ajuda*)
Descreva os comandos `echo, head, tail, diff, wc, chfn, chsh`.

Resolução:

Sistema de Ficheiros Unix

Exercício 10: (*Directórios*)
Crie na sua "home directory"dois directórios chamados respectivamente `publico` e `privado`. Desloca-se para o directório `publico`, crie um directório `tmp`. Remove o directório `tmp`.

Resolução:

Exercício 11: (*Ficheiros*)

1. executa os comandos seguintes `man ls > ls.txt` e `man wc>> ls.txt`. Qual é o resultado? vizualiza o resultado com os comandos `cat` e `more`.
2. Usando o comando `echo` e a redirecção, crie um ficheiro com o seu nome contendo o seu numero de estudante.
3. Faça uma copia desse ficheiro para a sua home directory. Move o ficheiro para "~/privado".

Quais são os comandos para verificar a localização do ficheiro criado e para verificar o conteúdo do ficheiro (ou seja, como usá-los).

Resolução:

Exercício 12: (*pwd*)
Em que directório está?

Resolução:

Exercício 13: (*Permissões*)
Tornar o directório `publico` público (escrita e leitura para o grupo, e leitura só, para os outros), o directório `privado` privado. Tenta deslocar-se para o directório `privado` de um colega seu, tenta copiar para esse directório o ficheiro com o seu nome.

Resolução:

Exercício 14: (*Caminho relativo, Caminho absoluto*)
Como diferenciar dois ficheiros homónimos?

Resolução:

Exercício 15: (*Hard links, symbolic links*)
Que faz o comando `ln`? Quais são as vantagens de uso de um tal comando? Faça um link de um ficheiro presente no seu directório `privado` para a sua home directory.

Resolução:

Processos

Exercício 16: (*Pipes*)
Como ver o conteúdo de um ficheiro da linha i até a linha j?

Resolução:

Exercício 17: (*Lista de processos*)
Como se usa o comando `ps`? Dê a lista de todos os processos que correm e dos quais o `root` é proprietrário?

Resolução:

Exercício 18: (*Filtros & Pipes*)
Dê a lista ordenada e página por página de todos os processos activos na sua máquina.

Resolução:

Exercício 19: (*Redirecção & Filtros*)
Crie um ficheiro com a lista completa e ordenada de todos os ficheiros da sua home directory.

Resolução:

Commandos UNIX

Exercício 20: (*Correio electrónico*)
Envie um mail ao seu vizinho, leia e Salve o mail recebido. Edita o ficheiro `.mailrc`.

Resolução:

Exercício 21: (*Comandos úteis*)
Exemplifique o uso dos comandos seguintes: `clear`, `du`, `df`, `which`, `whereis`, `tar`, `gzip`, `gunzip`, `lp`, `lpstat`, `cancel`, `alias`, `find`, `awk`.

Resolução:

Redirecção do I/O

Exercício 22: (*stdin e stdout*)
Experimente a seguinte sequência de comandos no seu terminal e dê uma explicação do que está, a cada passo, acontecendo. (O comando ctrl-d indica ao sistema que se atingiu o fim do input (ou seja o caracter `eof`=end of file)

```
ciunix> cat
algumas palavras
ctrl-d
ciunix> cat  > gato
Miau, brrrr
ctrl-d
ciunix> cat gato
ciunix> cat >> gato
miau
ctrl-d
ciunix> cat gato
ciunix> cat < gato
```

Resolução:

Exercício 23:
Explique a diferença entre os comandos `who | sort` e `who > sort`

Resolução:

Exercício 24:
Descreve e explique o que acontece quando se executa o comando `wc`, mal escrito, seguinte:

```
ciunix> wcll > temp
```

Resolução:

Exercício 25:
Explique o conteúdo do ficheiro criado pelo comando:

```
ciunix> ls > ls.out
```

Resolução:

Exercício 26:
Explique o output do comando:

```
ciunix> wc temp > temp
```

Resolução:

Exercício 27:
Crie os seguintes ficheiros com, por exemplo, o comando `touch`.
`cap1.txt cap2.txt cap3.txt cap4.txt cap?.txt`

1. Faça uma listagem dos ficheiros que começam com "cap"

2. Faça uma listagem dos ficheiros "cap1", "cap2"e "cap3"(use o modelo `[]`)
3. Faça uma listagem apenas dos ficheiros cujo nome é de 4 letras, que começam por "cap"que têm por extensão "txt".
4. Apagar os ficheiros que seguem o padrão "cap?.txt".

Resolução:

Ficheiros

Exercício 28:
Leia a página de ajuda on-line do comando `ls`.

Resolução:

Exercício 29:
Quais são as opções e uso do comando `cat`. Introduza texto no ficheiro cap1.txt, incluindo vários "tabs"e "newlines". Execute a seguir os seguintes comandos e explique o que acontece.

```
cat cap1.txt
cat -t cap1.txt
od -c cap1.txt
od -cb cap1.txt
```

Resolução:

Exercício 30:
Faça uma listagem de todos os ficheiros do seu directório actual (ls -a). Examine o ficheiro do directório (o ficheiro de nome ".") com o comando

```
od  -cb  .
```

(od = octal dump. Faça um `man od` para mais informação.).
Explique o output deste comando.

Resolução:

Exercício 31:
Quais são as permissões do ficheiro "."(`ls -ld`)? Será que se consegue executar um comando do género `who > .` ou `who > ..`?
Explique.

Resolução:

Exercício 32:
Crie um link simbólico e um link hard ao ficheiro "cap1.txt"no mesmo directório deste. A seguir faça uma listagem dos ficheiros usando os seguintes comandos: `ls -l` e `ls -i`.
Explique o output em relação aos i-nodes e links.

Resolução:

Pipes e os Filtros Grep, Sed e Awk

Exercício 33:
Juntando os comandos who e wc com um pipe crie um linha de comando que escreverá o número actual de pessoas "logged on".

1. Escreva esta linha num ficheiros chamado **nu** e a seguir "execute"este ficheiro (se a execução falhar, verifique as permissões do ficheiro).
2. Verifique que o seu caminho (path) inclua o seu directório bin ˜aluno/bin. Fazer isto examinando a variável do ambiente PATH com o comando `echo $PATH`.
3. Mude o ficheiro nu para o seu directório bin. Ainda consegue executar este ficheiro ? Porquê ?

Resolução:

Exercício 34:
A criação do ficheiro `nu` podia ter passado pelo comando `echo "who | wc -l≫ nu`. (i)Verifique e (ii) O que aconteceria se o comando for escrito sem as aspas ?

Resolução:

Exercício 35:
Partindo do seu directório raiz e usando os comandos `ls -R` (opção `R` $\Longrightarrow$ Recursividade) e `grep`, faça uma listagem de todos os seus ficheiros executáveis (os ficheiros que têm uma permissão a "x").

Resolução:

Exercício 36:
Partindo do directório raiz da sua área e usando os comandos `ls` e `grep`, faça uma listagem dos seus ficheiros cujos nomes não contêm qualquer número (ou seja, só letras).

Resolução:

Exercício 37:
Utilize os comandos `finger`, `grep` e `sort` para obter uma listagem ordenada dos utilizadores ligados ao servidor sem a primeira linha do output do `finger`.

Resolução:

Exercício 38:
Escreva algum texto nos ficheiros "cap?.txt".

1. Usando o comando `sed` substitua a letra "o"por "A"nos ficheiros considerados.
2. Tente fazer a mesma coisa com o comando `tr`.
3. Crie um script com os comandos da primeira alínea.

Resolução:

Exercício 39:
Neste exercício vamos utilizar o completíssimo comando `awk` usando como input o output dos comandos `ps`, `who` e `finger`.

1. Escolha alguns campos destes comandos, e selecioná-los para o output, usando o `awk`.
2. Um dos campos do comando `ps -auxw` é o `% tempo cpu`. Usando `awk` mande para o output a identificação do processo que está a utilizar o valor máximo.
3. Invente os seus próprios exemplos!

Resolução:

Processos

Exercício 40:
Qual é a diferença entre as três seguintes linhas de comando

```
ciunix> sleep  15; echo "hora do cha"
ciunix> sleep  15 & ; echo "hora do cha"
ciunix> (sleep 15; echo "hora do cha")&
```

Resolução:

Exercício 41:
Executa o comando o comando `sleep 120 &`.
Faça, a seguir, uma listagem dos seus processos (`ps` e `ps -axw`) e mate (`kill ...`) o seu processo `sleep 120`.

Exercício 42:

1. Faça um link simbólico do ficheiro `cicloslink` ao ficheiro `~crocker/prog3/ciclos`.
2. Examine o conteúdo e explique.
3. Executa o programa com : `cicloslink 1`. Termine o programa premindo `ctrl-c`, faça `ps`. Volte a executar `cicloslink 1` e prima, desta vez, em `ctrl-z` e faça `ps`. Qual a diferença ?
4. Se `cicloslink` ainda esteja a correr, executa o comando `fg`. Explique o que acontece.
5. Parar o programa `cicloslink 1` com `ctrl-z` e executa o comando `cicloslink 2` seguido por `ctrl-z` e `cicloslink 3` seguido por `ctrl-z`.
6. Examine o output de `ps -lw` e explique os campos `PID` e `PPID`.
7. Examine o output do comando `jobs`.
8. Experimente o comando `fg %1` ou `fg %2` etc... Explique.
9. O comando `bg` põe a correr um processo parado em "background". Experimente com um dos seus processos parados. Tente parar um processo da sua escolha sem usar `kill -9`.
10. O que acontece ao colocar um processo a correr em fg e ao premir, de seguida, `ctrl-z`?
11. Qual é o `PID` do seu "login shell"? qual é o programa e o dono do pai deste processo ?

Resolução:

O comando find

Exercício 43:
Localize, a partir do seu directório raiz,

1. Os ficheiros com mais de 10 blocos ?
2. Os ficheiros com menos de 10 blocos ?
3. Os ficheiros cujos nome contêm "cap"
4. Os ficheiros cujas extensões são .txt

Resolução:

Exercício 44:
Crie, utilizando o comando `touch`, em vários subdirectórios seus um ficheiro chamado `core`.

1. Localize, a partir do seu directório raiz, estes ficheiros.
2. Apague estes ficheiros utilizando `find` e `exec`.

Resolução:

Exercício 45:
Localize os seus ficheiros que têm:

1. um tempo de modificação inferior a um dia
2. um tempo de acesso inferior a um dia.

Resolução:

Exercício 46:
Invente mais três exemplos de uso do comando `find`, ilustrar as opções (ou flags) `exec` e `-perm`.
