

Resolução da ficha 2

1. Crie um ficheiro de nome “info.txt” com o seguinte conteúdo:

```
Coimbra, 1 de Abril de 2014.  
Hoje e dia de festa.  
Fim de ficheiro.
```

Resolução

```
$ cat > info.txt  
Coimbra, 1 de Abril de 2014  
Hoje e dia de festa.  
Fim de ficheiro.  
^D
```

O comando cat é usado para transpor o input recebido na entrada standard (neste caso, o teclado) para a saída standard. O redireccionamento especificado desvia a o conteúdo enviado para a saída standard para o ficheiro info.txt. Assim, o texto introduzido é armazenado no ficheiro, tal como pretendido. A combinação de teclas control-D causa o carácter fim-de-ficheiro, fazendo o comando terminar.

Resolução alternativa

```
$ nano info.txt  
Coimbra, 1 de Abril de 2014  
Hoje e dia de festa.  
Fim de ficheiro.  
^O  
^X
```

O comando nano corresponde a um editor de texto. O nome do ficheiro a editar é fornecido na linha de comandos. Após invocar o editor, será necessário escrever o conteúdo pretendido e no fim gravar (teclas control-O) e sair (teclas control-X).

2. Adicione os seguintes grupos ao sistema:

```
GRUPO  
so  
coimbra
```

Resolução

```
$ sudo groupadd so  
$ sudo groupadd coimbra
```

O comando `groupadd` é usado para adicionar grupos ao sistema. O nome do grupo é dado como argumento na linha de comandos, sendo o comando invocado uma vez para cada grupo adicionado. O comando suporta mais argumentos e opções que convém ver nas páginas de manual (comando `man`) e na documentação online. O comando `groupadd` é de utilização restrita ao superuser, pelo que é necessário usar o comando `sudo` em conjunto com o comando `groupadd`.

3. Adicione os seguintes utilizadores ao sistema:

LOGIN	NOME COMPLETO	GRUPO PRIMÁRIO
manuel	Manuel Antonio Pires	so
ze	Jose Manuel Santos	so
ana	Ana Santos	coimbra
antonio	Antonio Manuel	so
beto	Alberto Joao Santos	so

Resolução (apresenta-se apenas a resolução para o utilizador manuel como exemplo para os restantes)

```
sudo useradd -g so -c "Manuel Antonio Pires" -m manuel
```

O comando `useradd` é usado para adicionar utilizadores ao sistema. Este comando suporta vários argumentos e opções que convém ver nas páginas de manual (comando `man`) e na documentação online. Neste caso é indicado o nome do grupo (opção `-g nome-do-grupo`), o nome completo/comentário (opção `-c nome-completo/comentário`), e é dada a indicação para criar a homedir (opção `-m`). O nome do novo utilizador é o último argumento.

4. Visualize, página a página, o conteúdo do ficheiro '`passwd`', existente na directoria '`/etc`'.

Resolução

```
more /etc/passwd
```

O comando `more` permite visualizar um ficheiro, apresentado o seu conteúdo página a página. O ficheiro a apresentar é indicado como argumento na linha de comandos.

Resolução alternativa

```
cat /etc/passwd | more
```

Não sendo indicado nenhum ficheiro ao comando `more`, este irá apresentar o conteúdo daquilo que lhe for fornecido na entrada de dados standard. Essa característica permite usar o comando `more` como recipiente de redireccionamento de outro comando. Neste caso, usa-se o comando `cat` para transpor o conteúdo do ficheiro `/etc/passwd` para a saída standard, e, usando um pipe para redireccionar a saída standard para a entrada standard do comando `more`, consegue-se o mesmo efeito da resolução anterior.

A primeira resolução é melhor na medida em que usa menos recursos.

5. Mostre o *login*, número do grupo, nome completo e pasta pessoal dos utilizadores do sistema que tenham com grupo primário 'so', guardando essa informação num ficheiro de nome 'alunos_so'.

Resolução

```
cat /etc/group | grep ^so: | cut -d: -f3
1000
cat /etc/passwd | cut -d: -f1,4-6 | grep :1000: > alunos_so
```

A solução é composta por dois passos. A informação pretendida acerca dos utilizadores encontra-se no ficheiro `/etc/passwd`. No entanto, neste ficheiro, o grupo é dado pelo seu ID (número) e não pelo seu nome. Assim, é necessário obter primeiro no ficheiro `/etc/group` o número do ID correspondente ao grupo “so”.

No primeiro passo vai-se obter o ID correspondente ao grupo “so”. Essa informação encontra-se no ficheiro `/etc/group`. O comando `cat` permite abrir esse ficheiro, colocando o seu conteúdo na saída standard. Através de um pipe, redirecciona-se a saída standard do comando pipe para a entrada standard do comando `grep`, o qual é usado para filtrar a informação recebida pelo `cat`, transpondo para a sua saída apenas as linhas que começam por “so” (expressão regular `^so`). O uso de um novo pipe permite redireccionar o output do `grep` para o comando `cut`, o qual é usado para filtrar apenas o campo 3 da linha recebida pelo `grep` e que descreve o grupo “so”. O campo 3 é precisamente o que tem o ID do grupo, sendo indicado ao `cut` pela opção `-f3`. A informação no ficheiro `/etc/group` está organizada em campos delimitados pelo carácter “:”, pelo que é necessário indicar esse carácter delimitador ao comando `cut` (opção `-d:`). No exemplo, o grupo tem o ID 1000 (noutras máquinas poderá ser diferente).

No segundo passo, tendo o valor do ID do grupo, é necessário obter os nomes que têm esse grupo com o grupo primário. A informação pretendida encontra-se no ficheiro `/etc/passwd`. Usa-se o comando `cat` para obter o conteúdo desse ficheiro. O uso de um pipe redirecciona o output do comando `cat` (o conteúdo do ficheiro `/etc/passwd`) para a entrada do comando `cut`. Este permite filtrar os campos (colunas) excluindo as que não interessam. Assim, da informação recebida pelo comando `cat`, pretende-se transpor para o output do comando `cut` apenas as colunas 1,4,5 e 6 (opção `-f1,4-6`), as quais correspondem à informação pedida. A utilização de um novo pipe redirecciona a saída do comando `cut` para o comando `grep`. Este último filtra a informação recebida, apresentando no seu output apenas as linhas que cumprem o critério indicado, que é conterem o ID do grupo “so” (neste caso 1000. Tal como no ficheiro anterior, o carácter delimitador é o “:”. Por fim, redirecciona-se a saída do comando `grep` para o ficheiro `alunos_so`, tal como pedido.

Nota: algumas das opções dos comandos podem ser dadas de forma ligeiramente diferente (por exemplo, `-d:”` em vez de `-d:`, ou `-f1,4,5,6` em vez de `-f1,4-6`, “`^so`” em vez de `^so`). Esta observação também se aplica às restantes resoluções desta ficha.

Resolução alternativa – uso de ``

```
cat /etc/passwd | cut -d: -f1,4-6 | grep :` cat /etc/group | grep ^so: | cut -d: -f3
`: > alunos_so
```

Esta solução é a mesma que a anterior. No entanto, tira partido da possibilidade de se executar um (ou mais) comandos dentro dos caracteres `` , sendo o output resultante desse(s) comando(s) usado no resto da linha de comandos como se tivesse sido escrito explicitamente. Assim, o ID do grupo “so” é obtido através de `` cat /etc/group | grep ^so: | cut -d: -f3`` e o valor resultante usado como argumento no resto da expressão, sendo o efeito final o mesmo que na resolução anterior, mas usando apenas uma única linha de comandos.

A sintaxe \$() é semelhante a `` , o que permite escrever a última solução da seguinte forma:

Resolução alternativa – sintaxe \$()

```
cat /etc/passwd | cut -d: -f1,4-6 | grep :$( cat /etc/group | grep ^so: | cut -d: -f3): > alunos_so
```

6. Mostre o 3º, 4º e 5º caracteres de todas as linhas de 'alunos_so1'. Mostre de seguida os 8 primeiros caracteres.

Resolução

```
$ cut -c3,4,5 alunos_so
$ cut -c1-8 alunos_so
```

Usa-se o comando cut para abrir o ficheiro indicado e extrair os caracteres pretendidos. OS caracteres podem ser indicados através da opção -c.

7. Mostre a coluna correspondente aos *logins*, no ficheiro 'alunos_so'.

Resolução

```
$ cut -f1 -d: alunos_so
```

A informação pretendida encontra-se na primeira coluna do ficheiro (este ficheiro é resultante das perguntas anteriores). Usa-se o comando cut para abrir o ficheiro e extrair a primeira coluna, não esquecendo de indicar que o caracter delimitador é o ":".

8. Visualize o conteúdo do ficheiro 'alunos_so', de forma a mostrar apenas o *login* e o nome completo de cada utilizador.

Resolução

```
$ cut -f1,3 -d: alunos_so
```

Caso em tudo semelhante ao exercício anterior, mas neste caso pretende-se mostrar também a coluna 3 do ficheiro, pelo que se indicam ambas as colunas 1 e 3 (opção -f) ao comando cut.

9. Repita o exercício anterior, mas utilize o caracter *tab* como separador. Experimente com outros separadores.

Resolução

```
$ cut -f1,3 -d: alunos_so | tr ":" "\t"
$ cut -f1,3 -d: alunos_so | tr ":" ",,"
```

Aqui pretende-se ter no output o caracter tab em vez do ":". O exercício incide, não sobre o caracter que o cut vai usar como delimitador (esse caracter é o ":" pois é o que está no ficheiro), mas sim sobre o output que se vai enviar para o ecrã. Neste caso, pretende-se substituir o ":" no output do comando

cut pelo carácter tab. Para este efeito redirecciona-se com um pipe o output do comando cut para o comando tr, instruindo este para substituir todos os ":" pelo carácter "\t" ("\t" é o carácter tab). Na segunda linha efectua-se algo semelhante, mas em vez de tab usa-se uma vírgula.

Resolução alternativa

```
$ cut -f1,3 -d: --output-delimiter " " alunos_so  
$ cut -f1,3 -d: --output-delimiter "," alunos_so
```

Nesta resolução indica-se directamente ao comando cut que se pretende ter, no output, o carácter tab (primeira linha) ou vírgula (segunda linha) em vez do delimitador (":" em ambos os casos). Assim evita-se o uso do comando tr. É preciso ter em atenção que o carácter tab tem que ser escrito da seguinte forma: control-v seguido de TAB.

10. Redireccione o output do exercício 6 para um ficheiro designado 'pessoas'.

Resolução

```
$ cut -c3,4,5 alunos_so > pessoas  
$ cut -c1-8 alunos_so >> pessoas
```

Este exercício é em tudo semelhante ao exercício 6, sendo a única diferença o facto de se redireccionar o output do comando cut para um ficheiro. O redireccionamento usado na segunda linha indica que se deve acrescentar ao ficheiro pessoas em vez de simplesmente reescrever por cima.

11. Visualize o ficheiro 'alunos_so', ordenado alfabeticamente pelo *login*.

Resolução

```
$ sort alunos_so -k1 -t:
```

Vai-se recorrer ao comando sort, o qual, dado um texto, produz na saída standard uma versão ordenada desse texto. O texto tanto pode ser obtido através da entrada standard, como por um ficheiro cujo nome é indicado na linha de comandos (como acontece neste solução). O comando sort suporta vários argumentos e opções (os quais dêem ser consultados), permitindo especificar o critério de ordenação. Neste caso especifica-se que é para ordenar de acordo com o campo 1 (opção -k), sendo o carácter delimitador de campos o ":" (opção -t:).

12. Mostre todas as linhas do ficheiro 'pessoas' que contenham a palavra "Manuel" e em seguida visualize também o número da linha em que elas se encontram.

Resolução

```
$ grep Manuel alunos_so  
$ grep -n Manuel alunos_so
```

Recorre-se ao comando grep, como habitualmente nos casos em que é preciso obter uma versão de um texto contendo apenas algumas das linhas de acordo com um critério (neste caso, o critério é conterem a palavra "Manuel"). A segunda linha usa a opção -n que tem como efeito apresentar também o número de linha em todas as linhas apresentadas.

13. Visualize todas as linhas do ficheiro 'alunos_so' que contenham um "0" seguido do carácter "1" ou "2".

Resolução

```
$ grep 0[12] alunos_so
```

Neste exercício há apenas a notar a expressão regular no comando grep para conseguir o filtro pretendido. A expressão 0[12] indica que as linhas contendo “01” ou “02” cumprem o filtro e devem ser enviadas para o output.

É de recordar que também se poderia ter especificado "0[12]" com as aspas. Como a expressão regular não contém espaços, não foi necessário recorrer às aspas.

14. Liste a informação de todos utilizadores existentes no ficheiro 'alunos_so' cujo login comece por "a" ou "b".

Resolução

```
$ grep ^[ab] alunos_so
```

Neste exercício há apenas a notar, na expressão regular indicada ao comando grep, a forma como se indica o início de linha. O carácter ^ indica início de linha, e o carácter \$ (não usado neste exercício) indica fim de linha. Assim, a expressão ^[ab] faz com que apenas as linhas que começam por a ou por b sejam enviadas para o output.

15. Liste todos os nomes completos existentes no ficheiro 'alunos_so' que contenham a palavra "Antonio" e de seguida aqueles cujo primeiro nome seja "Antonio".

Resolução

```
$ cat alunos_so | cut -d: -f3 | grep Antonio  
$ cat alunos_so | cut -d: -f3 | grep ^Antonio
```

Neste exercício juntam-se as funcionalidades do comando cat (apresentar o conteúdo de um ficheiro) com o comando cut (apresentar apenas determinados campos/colunas) e do comando grep (apresentar apenas determinadas linhas de acordo com um critério indicado através de um expressão regular). O output dos comandos encontra-se redireccionado à entrada do comando seguinte através de pipes e toda a funcionalidade destes comandos usada aqui foi já vista nos exercícios anteriores. Na segunda linha usa-se o carácter ^ na expressão regular do comando grep para indicar “início de linha” tal como no exercício anterior.

16. Crie um ficheiro com o nome completo dos utilizadores do sistema que tenham como apelido "santos" ou "Santos". Este ficheiro deverá estar ordenado por ordem alfabética.

Resolução

```
$ cat /etc/passwd | cut -d: -f5 | grep [sS]antos$ | sort > ficheiro.txt
```

Vai-se usar o comando cat para obter o conteúdo do ficheiro /etc/passwd, o qual contém a informação pretendida acerca dos utilizadores. Através de um pipe envia-se a informação produzida pelo cat para o

comando cut. O comando cut é necessário para filtrar apenas a coluna 5, que é a que tem o nome completo do utilizador. O comando grep filtra as linhas produzidas pelo cut (recebidas através de redireccionamento com um pipe), enviando para o output apenas aquelas que terminam pro “Santos” ou “santos” (o carácter \$ indica fim de linha). Por fim, o output do grep é redireccionado para o ficheiro ficheiro.txt.

17. Mostre o número total de palavras existentes na penúltima linha do ficheiro ‘info.txt’.

Resolução

```
$ cat info.txt | tail -n 2 | head -n 1 | wc -w
```

Usa-se o cat para obter o conteúdo do ficheiro info.txt. Através de um redireccionamento com um pipe, envia-se o output do comando cat (o conteúdo do ficheiro) para o comando til. Este comando permite obter a parte final (últimas linhas) do texto recebido. Existem diversas formas de expressar quais as linhas pretendidas (as últimas n, da n-ésima em diante, etc.), consoante os argumentos e opções indicadas. É necessário consultar a página de manual deste comando. Neste caso, indica-se que se deve enviar para o output apenas as duas últimas linhas (opção -n 2).

O output do comando tail é enviado, através de redireccionamento com pipe, para o comando head. Este comando é muito semelhante ao tail, mas trabalha coma parte inicial de textos, em vez da parte final. Também é necessário consultar a página de manual deste comando para ver as várias formas como se pode expressar as linhas que se deseja obter. Neste caso está-se a dizer que se pretende ter apenas a primeira linha das que foram recebidas (opção -n 1). As linhas recebidas foram as duas últimas do ficheiro. A primeira destas duas será a penúltima do ficheiro, tal como pretendido no enunciado.

O resultado do comando head é enviado para o comando wc, sendo-lhe indicado que se pretende contar o número de palavras (opção -w), sendo o resultado final o número de palavras existentes na penúltima linha do ficheiro info.txt

18. Mostre todos os grupos existentes no sistema e respectivo GID, separados por um espaço. Esta informação deverá estar ordenada pelo nome do grupo.

Resolução

```
$ cut -d: -f1,3 /etc/group | tr ":" " " | sort
```

A informação pretendida encontra-se nas colunas 1 e 3 do ficheiro /etc/group,oO qual usa o carácter “:” como delimitador de campos. Assim, recorre-se ao comandos cut da forma habitual para obter essas duas colunas a partir do ficheiro /etc/group. O enunciado especifica que se pretende ver a informação separada por um espaço em vez de pelo carácter delimitador existente no ficheiro (“:”). Tal como no exercício 9, pode-se recorrer ao comando tr para substituir o carácter “:” pelo espaço no output produzido pelo cut (obtido por redireccionamento). Note-se que nenhuma substituição ocorre no ficheiro original. Por fim redirecciona-se o output para o comando sort para obter a informação ordenada, tal como pedido no enunciado.

A obtenção do carácter espaço e mvez do “:” no output pode também ser obtida directamente pelo comando cut, tal como já anteriormente visto no exercício 9, dando origem à resolução alternativa seguinte.

Resolução alternativa

```
$ cut -d: -f1,3 --output-delimiter=" " /etc/group | sort
```

Nesta resolução alternativa indica-se ao comando cut que, no output, o caracter delimitador deve ser substituído pelo espaço.

19. Mostre todas as entradas do sistema de ficheiros cujo nome contém a terceira palavra da linha número 2 do ficheiro 'info.txt'.

Resolução temporária

```
$ cat info.txt | head -n 2 | tail -n 1 | cut -d' ' -f3
dia
ls -l | grep dia
```

Esta resolução de carácter temporário serve para perceber os dois passos envolvidos na solução definitiva.

A resolução correcta deste exercício exige saber em primeiro lugar qual a terceira palavra da segunda linha do ficheiro info.txt. A obtenção dessa palavra é relativamente directa, sendo semelhante ao exercício 17, usando-se os comandos cat, head, tail e cut, em sequência, ligados por pipes. O comando cat obtém o conteúdo do ficheiro info.txt. Este conteúdo é passado ao comando head, o qual selecciona as duas primeiras linhas (opção -n 2), passando-as ao comando tail. Este selecciona a última linha (opção -n 1), correspondendo então esta à última linha das duas primeiras do ficheiro, ou seja à segunda, tal como pretendido. A linha é passada ao comando cut, o qual vai considerar a linha recebida como estando formatada em campos separados pelo delimitador espaço (opção -d' ') e seleccionado o terceiro campo, ou seja, a terceira palavra (opção -f3), sendo o resultado final a terceira palavra da segunda linha do ficheiro. No exemplo, a palavra é "dia".

De seguida usa-se o comando ls com a opção -l para obter a listagem da directoria actual. O resultado é enviado ao comando grep que filtra todas as linhas obtidas, apresentando apenas aquelas que têm a palavra "dia".

Resolução definitiva – uso de ``

```
$ ls -l | grep `head -n 2 info.txt | tail -n 1 | cut -d' ' -f3`
```

Esta é a solução que se pretende. Junta os dois passos num só usando da sintaxe `` , através da qual um conjunto de comandos delimitado pelos `` é executado em primeiro lugar, sendo o output resultante usado de seguida para o restante da linha de comandos. Assim, os comandos que obtém a terceira palavra da segunda linha do ficheiro info.txt são colocados dentro de `` , e a palavra resultante (no exemplo, "dia") é usada para o comando grep.

Note-se que é possível que os comandos dentro de `` resultem em texto contendo espaços. Nesta situação é conveniente usar aspas ao redor dos `` (ficando então " comandos ").

A sintaxe \$() é semelhante à `` pelo que se pode também ter a seguinte solução alternativa

Resolução alternativa – sintaxe \$()

```
$ ls -l | grep `head -n 2 info.txt | tail -n 1 | cut -d' ' -f3`
```


Esta alternativa é em tudo semelhante à anterior, variando apenas o uso de \$() em vez de ``. Aplica-se a mesma observação relativamente à hipótese dos comandos dentro de \$() poderem resultar em texto contendo espaços e portanto, ser de encerrar a hipótese de uso de aspas à volta do \$().

20. Que linha de comandos retorna o número actual de vezes que o utilizador “ana” está *logado* no sistema?

Resolução

\$ who grep ana wc -l

O comando who apresenta os utilizadores que estão logados, uma linha por utilizador. Através de um pipe envia-se o output do comando who para o comando grep, indicando a este que deve filtrar e apresentar apenas as linhas contendo “ana”. Se seguida, e também através de um pipe, envia-se este resultado para o comando wc indicando-lhe que deve contar o número de linhas. O resultado final (número de linhas obtidas pelo who contendo ana) corresponde ao número de vezes que o utilizador ana está logado.

Tal como nos outros exercícios, o uso de aspas pode ser importante se o texto usado tiver espaços.