

# SISTEMAS OPERATIVOS

Multi utilização

# LINUX – CLI – STANDARD INPUT, STANDARD OUTPUT E STANDARD ERROR

O standard input é a origem (entrada), por omissão, dos dados lidos pelos programas. Normalmente está associado ao teclado. O standard output é o destino (saída), por omissão, dos dados escritos pelos programas. Normalmente está associado ao monitor. É habitual chamar, abreviadamente, ao standard input e ao standard output respectivamente stdin e stdout. I/O é normalmente usado para referir a entrada(I-Input) e/ou a saída (O-Output) de um programa/processo.

A forma mais abrangente de um comando em Unix pode ser expressa como

**xpto**

em que o comando espera a entrada dos dados vindos do stdin(teclado), processa-os e envia os para o stdout (monitor).

Na realidade, muitos dos comandos do Unix (eventualmente a maioria) acabam por não seguir exactamente este comportamento mas, mesmo esses, podem ser encarados como casos particulares da forma indicada.



**cat**

O comando cat comporta-se como um filtro neutro entre stdin e stdout. A introdução do EOF(End Of File) faz-se usando Ctrl+D (e não Ctrl+Z, como no DOS! Em Unix Ctrl+Z desencadeia uma acção completamente diferente). A terminação (forçada) do processo consegue-se com Ctrl+C (e não Ctrl+Z (parece equivalente mas é muito diferente)).

```
quental@quental-VirtualBox:~$ cat
Estou a digitar no teclado (stdin) e quando clicar em <enter> esta mensagem vai aparece
r no monitor (stdout).
Estou a digitar no teclado (stdin) e quando clicar em <enter> esta mensagem vai aparece
r no monitor (stdout).
```



**wc**

```
quental@quental-VirtualBox:~$ wc
Quantas linhas, palavras e caracteres estão presentes na mensagem que eu estou a digita
r?
Vamos deixar que seja o wc (word count e não o "Water Closet, toilet") a fazer esta con
tagem.
      2      32     186
quental@quental-VirtualBox:~$
```

# LINUX – CLI – REDIRECCIONAMENTO DE I/O

Também é típico que quando se indica um ficheiro na linha de comando ele seja usado em vez do stdin (situação mais comum) ou do stdout:  
xpto ficheiro



**cat /etc/passwd**

o ficheiro /etc/passwd substitui o stdin!. Como resultado prático, resulta a visualização do seu conteúdo no stdout(monitor)

```
usbmux:x:120:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
quental:x:1000:1000:Quental,,,:/home/quental:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
```

Redireccionamento do stdout - É possível solicitar que, em vez de a saída ser enviada para o stdout (monitor), ela seja redireccionada para um ficheiro indicado (recorrendo ao operador '>').



**cat > exp0**  
**ls exp0**  
**cat exp0**

(exp0 é criado automaticamente, não precisa de existir; se existir, o conteúdo anterior perde-se).

```
quental@quental-VirtualBox:~$ cat > exp0
O que estou a digitar no teclado vai ser gravado no ficheiro de texto exp0 (stdout). Te
rmino com as teclas <ctrl> <d>quental@quental-VirtualBox:~$ ls exp0
exp0
quental@quental-VirtualBox:~$ cat exp0
O que estou a digitar no teclado vai ser gravado no ficheiro de texto exp0 (stdout). Te
rmino com as teclas <ctrl> <d>quental@quental-VirtualBox:~$
```

# LINUX – CLI – REDIRECCIONAMENTO DE I/O

Redireccionamento do stdout - Se não se desejar que o conteúdo se perca, pode-se usar '>>' para juntar (no fim)(append).



**cat >> exp0**

o que for digitado(stdin) é acrescentado no fim do conteúdo de exp0.

```
quental@quental-VirtualBox:~$ cat >> exp0
O que estou a escrever vai ser acrescentado ao ficheiro exp0
quental@quental-VirtualBox:~$ cat exp0
O que estou a digitar no teclado vai ser gravado no ficheiro de texto exp0 (stdout). Terminando com as teclas <ctrl> <d> o que estou a escrever vai ser acrescentado ao ficheiro exp0
quental@quental-VirtualBox:~$ █
```

Redireccionamento do stdout - Existe um ficheiro especial (/dev/null) para o qual pode ser redireccionada a saída que não se pretende de facto guardar/visualizar (quando a ideia é simplesmente "calar" um processo). Funciona como um buraco negro.



**cat exp0 > /dev/null**

```
quental@quental-VirtualBox:~$ cat exp0 > /dev/null
quental@quental-VirtualBox:~$
```

# LINUX – CLI – REDIRECCIONAMENTO DE I/O

**Redireccionamento do stdin** - É possível solicitar que, em vez de a entrada de dados ser efectuada a partir do stdin (teclado), ela se realize a partir de um ficheiro indicado(recorrendo ao operador '<').



**cat < exp0**

Pelo que já foi mencionado, naturalmente que cat <exp0 é equivalente a cat exp0.

**stderr** - Quando ocorre um erro(ex. ficheiro que não existe) é produzida uma mensagem de erro que é enviada para o stderr (por omissão o stderr coincide com o stdout(monitor)).



**cat exp0 ficheiroquenaosexiste**

```
to@estgv-so:~$ cat exp0 ficheiroquenaosexiste
It is a long established fact that a reader will be distracted by the readable
content of a page when looking at its layout.
cat: ficheiroquenaosexiste: No such file or directory
to@estgv-so:~$
```

conteúdo de exp0.



**cat exp0 ficheiroquenaosexiste >exp1**  
**ls exp?**  
**cat exp1**

mensagem de erro enviada para stderr

conteúdo de exp0 é enviado para o ficheiro exp1;  
a mensagem de erro é enviada para o stderr=monitor;  
stdout é diferente de stderr!!

```
to@estgv-so:~$ cat exp0 ficheiroquenaosexiste > exp1
cat: ficheiroquenaosexiste: No such file or directory
to@estgv-so:~$ ls exp?
exp0  exp1
to@estgv-so:~$ cat exp1
It is a long established fact that a reader will be distracted by the readable
content of a page when looking at its layout.
to@estgv-so:~$
```

# LINUX – CLI – REDIRECCIONAMENTO DE I/O

**Redireccionamento do stderr** - De forma análoga ao que acontece com o stdout, é possível redireccionar o stderr. A sintaxe desse redireccionamento depende da shell. Para a bash, este redireccionamento é realizável com o operador '2>'.  
O conteúdo de exp0 enviado para exp1.  
mensagem de erro enviada para erros



```
cat exp0 ficheiroquenaosexiste >exp1 2>erros
cat exp1
cat erros
```

```
ffrancisco@ffrancisco-VirtualBox:~$ cat exp0 ficheiroquenaosexiste >exp1 2>erros
ffrancisco@ffrancisco-VirtualBox:~$ cat exp1
O que estou agora a digitar no teclado(stdin) vai ser gravado no ficheiro de
texto exp0(stdout).Termino com as teclas <ctrl> <d>

O que estou a digitar vai ser acrescentado no fim de exp0.
ffrancisco@ffrancisco-VirtualBox:~$ cat erros
cat: ficheiroquenaosexiste: No such file or directory
ffrancisco@ffrancisco-VirtualBox:~$
```



```
cat exp0 ficheiroquenaosexiste >exp1 2>>erros
```

o que fará o '2>>'?  
Teste e tire conclusões.



# LINUX – CLI – REDIRECCIONAMENTO DE I/O

Redireccionamento simultâneo do stdout e do stderr para o mesmo destino – Realizável com o operador '>exp1'.



```
cat exp0 ficheiroquenaooexiste &>exp1  
cat exp1
```

Tanto as saídas destinadas ao stdout quanto a destinada ao stderr são concatenadas no ficheiro indicado.

```
to@estgv-so:~$ cat exp0 ficheiroquenaooexiste &>exp1  
to@estgv-so:~$ cat exp1  
It is a long established fact that a reader will be distracted by the readable  
content of a page when looking at its layout.  
cat: ficheiroquenaooexiste: No such file or directory  
to@estgv-so:~$
```

Nota: Se na mesma linha houver a necessidade de fazer o redireccionamento do stdout e do stderr para destinos distintos, primeiro deve-se realizar o redireccionamento do stdout e só depois realizar o redireccionamento do stderr.

```
to@estgv-so:~$ cat exp0 ficheiroquenaooexiste > osqueexistem 2> osquenaooexistem  
to@estgv-so:~$ cat exp0  
It is a long established fact that a reader will be distracted by the readable content of a page when look  
ing at its layout.  
to@estgv-so:~$ cat osqueexistem  
It is a long established fact that a reader will be distracted by the readable content of a page when look  
ing at its layout.  
to@estgv-so:~$ cat osquenaooexistem  
cat: ficheiroquenaooexiste: No such file or directory  
to@estgv-so:~$
```

# LINUX – CLI – REDIRECCIONAMENTO DE I/O

pipe – operador de redireccionamento '|' que redirecciona o stdout de um comando para o stdin de outros comando.



```
ls -l /usr/bin | less
```

```
ls -rlS | grep -v '^total' | head -5
```

o output do comando `ls -l /usr/bin` é redireccionado para o input do comando `less`

Resultado prático: lista os cinco ficheiros mais pequenos!!

```
ffrancisco@ffrancisco-VirtualBox:~/testes/so$ ls -rlS
total 1396
-rw-rw-r-- 1 ffrancisco ffrancisco  42 Fev 23 17:19 texto.txt
-rw-r--r-- 1 ffrancisco ffrancisco  43 Fev 23 18:00 xpto.pdf
-rw-r--r-- 1 ffrancisco ffrancisco  43 Fev 23 17:59 textol
-rw-rw-r-- 1 ffrancisco ffrancisco 2197 Mar  2 15:15 bigfile
```

```
ffrancisco@ffrancisco-VirtualBox:~/testes/so$ ls -rlS | grep -v '^total'
-rw-rw-r-- 1 ffrancisco ffrancisco  42 Fev 23 17:19 texto.txt
-rw-r--r-- 1 ffrancisco ffrancisco  43 Fev 23 18:00 xpto.pdf
-rw-r--r-- 1 ffrancisco ffrancisco  43 Fev 23 17:59 textol
-rw-rw-r-- 1 ffrancisco ffrancisco 2197 Mar  2 15:15 bigfile
-rw-r--r-- 1 ffrancisco ffrancisco 27199 Fev 23 18:04 Untitled2
-rw-rw-r-- 1 ffrancisco ffrancisco 55690 Fev 23 18:36 Lampadas
-rw-r--r-- 1 ffrancisco ffrancisco 303992 Fev 23 18:04 Untitled1
-rw-r--r-- 1 ffrancisco ffrancisco 1017278 Fev 23 18:05 Untitled3
```

```
ffrancisco@ffrancisco-VirtualBox:~/testes/so$ ls -rlS | grep -v '^total' | head -5
-rw-rw-r-- 1 ffrancisco ffrancisco  42 Fev 23 17:19 texto.txt
-rw-r--r-- 1 ffrancisco ffrancisco  43 Fev 23 18:00 xpto.pdf
-rw-r--r-- 1 ffrancisco ffrancisco  43 Fev 23 17:59 textol
-rw-rw-r-- 1 ffrancisco ffrancisco 2197 Mar  2 15:15 bigfile
-rw-r--r-- 1 ffrancisco ffrancisco 27199 Fev 23 18:04 Untitled2
```



# LINUX – CLI – REDIRECCIONAMENTO DE I/O

pipe - operador de redireccionamento '|' que redirecciona o stdout de um comando para o stdin de outros comando.



`file * | grep image`

Resultado prático: lista apenas os ficheiros de imagens

```
ffrancisco@ffrancisco-VirtualBox:~/testes/so$ file *
bigfile: UTF-8 Unicode text, with overstriking
Lampadas: Microsoft Excel 2007+
textol: ASCII text
texto.txt: UTF-8 Unicode text
Untitled1: PNG image data, 636 x 358, 8-bit/color RGBA, non-interlaced
Untitled2: JPEG image data, JFIF standard 1.01
Untitled3: TIFF image data, big-endian
xpto.pdf: ASCII text
```

```
ffrancisco@ffrancisco-VirtualBox:~/testes/so$ file * | grep image
Untitled1: PNG image data, 636 x 358, 8-bit/color RGBA, non-interlaced
Untitled2: JPEG image data, JFIF standard 1.01
Untitled3: TIFF image data, big-endian
```

# LINUX – CLI – REDIRECCIONAMENTO DE I/O

## Comandos para tratamento de ficheiros de texto(muito utilizado com pipes)

- find– searchawk – good for working with files that contain information in columns.
- cat– concatenate files and print on the standard output
- csplit – split a file into sections determined by context lines
- cut – print selected parts of lines from
- ch for files in a directory hierarchy
- fmt– reformat each paragraph of a file
- fold– wrap each input line to fit in specified width
- grep– prints the lines of a file that match a search string (string can be a regular expression)
- head– output the first part of files
- join– join lines of two files on a common field (files should be sorted by common field)
- nl– number lines of files
- paste– merge lines of files, separated by tabs. The columns of the input files are placed side-by-side with each other.
- perl– can do anything sed and awk can do, but not always as easily as shown in the examples above.
- sdiff– print differences between files
- sed– sed is a stream editor. A stream editor is used to perform basic text transformations on an input stream (a file or input from a pipeline).
- sort– sort lines of text files
- split– split a file into pieces (default is 1000 lines each)
- tail– output the last part of files
- uniq– displays unique lines from a sorted file
- wc– print the number of bytes, words, and lines in files



**file \* | grep image**

pipe- operador de redireccionamento 'l' que redirecciona o stdout de um comando para o stdin de outros comando.

# LINUX – CLI – REDIRECCIONAMENTO DE I/O

Tratamento de ficheiros de texto-exemplos



```
cat texto.txt xpto.pdf
```

```
f Francisco@f Francisco-VirtualBox:~/testes/so$ cat texto.txt xpto.pdf
Este é o meu primeiro ficheiro de texto.
fkdfghkdjhfgkjsdhfg
dfgds
fgdsfg

dfg
dfg
f Francisco@f Francisco-VirtualBox:~/testes/so$
```



```
file * | grep -v image | sort -r
```

Resultado prático: lista, por ordem alfabética inversa, apenas os ficheiros que não sejam de imagem.

```
f Francisco@f Francisco-VirtualBox:~/testes/so$ file * | grep -v image | sort -r
xpto.pdf: ASCII text
texto.txt: UTF-8 Unicode text
textol: ASCII text
Lampadas: Microsoft Excel 2007+
bigfile: UTF-8 Unicode text, with overstriking
f Francisco@f Francisco-VirtualBox:~/testes/so$
```

# LINUX – CLI – REDIRECCIONAMIENTO DE I/O

Tratamento de ficheiros de texto-exemplos



**cat pautaAPC**

```
f Francisco@VM so$ cat pautaAPC
Carla Mendes,1546,PL1,19,14
Paulo Fernandes,1345,PL2,19,18
Pedro Duarte,111098,PL2,14,19
Carolina Pais,12101,PL1,17,16
Manuel Esteves,10897,PL2,12,11
f Francisco@VM so$
```

Ficheiro de texto csv( comma-separated values) :  
Nome, Número, Turno, Class. Prática, Class. Teórica



**cat pautaAPC | cut -d "," -f1-2,4-5**

```
f Francisco@VM so$ cat pautaAPC | cut -d "," -f1-2,4-5
Carla Mendes,1546,19,14
Paulo Fernandes,1345,19,18
Pedro Duarte,111098,14,19
Carolina Pais,12101,17,16
Manuel Esteves,10897,12,11
f Francisco@VM so$
```

Resultado práctico : omite  
a columna Turno

# LINUX – CLI – REDIRECCIONAMENTO DE I/O

## Tratamento de ficheiros de texto-exemplos



```
cat pautaAPC | awk -F "," '{printf"%10s %-25s %s\n",$2,$1,$3}' | sort -g
```

```
f Francisco@VM so$ cat pautaAPC | awk -F "," '{printf"%10s %-25s %s\n",$2,$1,$3}' | sort -g
      1345 Paulo Fernandes          PL2
      1546 Carla Mendes            PL1
     10897 Manuel Esteves          PL2
     12101 Carolina Pais           PL1
     111098 Pedro Duarte           PL2
f Francisco@VM so$
```

Resultado prático : lista  
formatada, por ordem  
crescente do Número



```
cat pautaAPC | awk -F "," '{printf"%-25s %10s %s\n",$1,$2,$3}' | sort | grep PL2
```


```
f Francisco@VM so$ cat pautaAPC | awk -F "," '{printf"%-25s %10s %s\n",$1,$2,$3}' | sort | grep PL2
Manuel Esteves          10897 PL2
Paulo Fernandes         1345 PL2
Pedro Duarte            111098 PL2
f Francisco@VM so$
```

Resultado prático : lista  
formatada, por ordem alfabética  
do Nome, do turno PL2



# LINUX – CLI – REDIRECCIONAMIENTO DE I/O

## Tratamento de ficheiros de texto-exemplos



```
file * | grep image | awk -F ":" '{printf " mv %s imgs\n", $1}' | bash
```

ffrancisco@VM so\$ file \*

```
bigfile: UTF-8 Unicode text, with overstriking
Lampadas: Microsoft Excel 2007+
pautaAPC: ASCII text
textol: ASCII text
texto.txt: UTF-8 Unicode text
Untitled1: PNG image data, 636 x 358, 8-bit/color RGBA, non-interlaced
Untitled2: JPEG image data, JFIF standard 1.01
Untitled3: TIFF image data, big-endian
xpto.pdf: ASCII text
ffrancisco@VM so$
```

ffrancisco@VM so\$ file \* | grep image

```
Untitled1: PNG image data, 636 x 358, 8-bit/color RGBA, non-interlaced
Untitled2: JPEG image data, JFIF standard 1.01
Untitled3: TIFF image data, big-endian
ffrancisco@VM so$
```

ffrancisco@VM so\$ file \* | grep image | awk -F ":" '{printf " mv %s imgs\n", \$1}'

```
mv Untitled1 imgs
mv Untitled2 imgs
mv Untitled3 imgs
ffrancisco@VM so$
```

ffrancisco@VM so\$ mkdir imgs

ffrancisco@VM so\$ file \* | grep image | awk -F ":" '{printf " mv %s imgs\n", \$1}' | bash

ffrancisco@VM so\$ ls -la imgs

```
total 1332
drwxrwxr-x 2 ffrancisco ffrancisco 4096 Mar  8 12:32 .
drwxrwxr-x 3 ffrancisco ffrancisco 4096 Mar  8 12:32 ..
-rw-r--r-- 1 ffrancisco ffrancisco 303992 Feb 23 18:04 Untitled1
-rw-r--r-- 1 ffrancisco ffrancisco 27199 Feb 23 18:04 Untitled2
-rw-r--r-- 1 ffrancisco ffrancisco 1017278 Feb 23 18:05 Untitled3
ffrancisco@VM so$
```

Resultado práctico move  
para a subdirectoria imgs  
todos os ficheiros de imagem.