
Redireccions d'entrada i sortida

Índex

1. Descriptors de fitxer	1
2. Redirecció de la sortida estàndard (stdout)	2
> --- Redirigir a un fitxer (sobreescriu)	2
>> --- Afegir a un fitxer (no sobreescriu)	2
3. Redirecció de la sortida d'errors (stderr)	2
2> --- Redirigir errors a un fitxer	2
2>> --- Afegir errors a un fitxer	2
4. Redirecció de stdout i stderr simultàniament	3
> fitxer 2>&1 --- Tots dos canals al mateix fitxer	3
&> --- Sintaxi abreujada de Bash (Bash 4+)	3
Separar stdout i stderr en fitxers diferents	3
5. Redirecció de l'entrada estàndard (stdin)	3
< --- Llegir des d'un fitxer	3
<< --- <i>Here Document</i> (document incrustat)	3
<<< --- <i>Here String</i> (cadena incrustada)	4
6. La canonada () --- Pipe	4
Pipeline amb stderr	4
7. El dispositiu /dev/null	4
8. tee --- Escriure i mostrar alhora	5
9. Resum dels operadors de redirecció	5
10. Exemples pràctics combinats	6
Guardar la sortida d'una ordre i veure-la a la vegada	6
Comptar línies que contenen errors en un log	6
Ordenar i eliminar duplicats d'una llista	6
Executar script guardant tota la sortida (stdout + stderr)	6
Cerca silenciosa (sense mostrar errors de permisos)	6
Enviar el contingut d'un fitxer per correu	6
Generar un fitxer de configuració amb here document	6
Pipeline complex: els 5 processos que més CPU consumeixen	7
11. Errors comuns	7
Sobreescriure un fitxer sense voler	7
Ordre incorrecte en 2>&1	7
Confondre > amb >>	7
12. Protecció contra sobreescritura (noclobber)	7

A GNU/Linux, cada procés treballa amb tres **canals estàndard** de comunicació, coneguts com a *streams* estàndard:

Canal	Nom	Descriptor	Per defecte
Entrada estàndard	stdin	0	Teclat
Sortida estàndard	stdout	1	Pantalla (terminal)
Sortida d'errors	stderr	2	Pantalla (terminal)

La **redirecció** permet canviar l'origen o la destinació d'aquests canals: llegir des d'un fitxer en comptes del teclat, escriure la sortida en un fitxer en comptes de la pantalla, o enviar la sortida d'un programa com a entrada d'un altre.

1. Descriptors de fitxer

El nucli Linux identifica cada canal obert amb un **descriptor de fitxer** (un nombre enter):

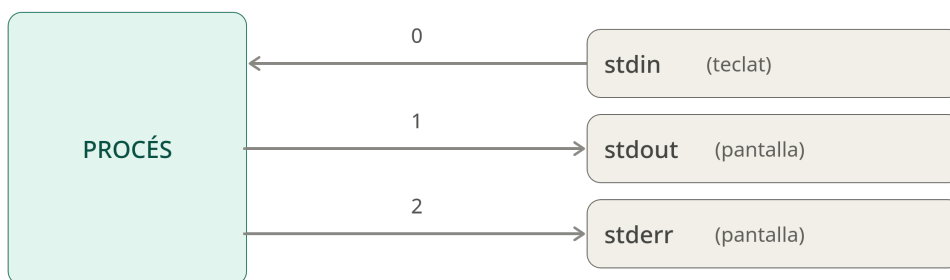


Figura 1: Descriptors

Quan es redirecciona, es canvia a quin fitxer o dispositiu apunta cada descriptor.

2. Redirecció de la sortida estàndard (stdout)

> --- Redirigir a un fitxer (sobreescriu)

Envia la sortida d'una ordre a un fitxer. Si el fitxer ja existeix, el **sobreescriu**.

```
ls -l > llistat.txt
echo "Hola món" > salutació.txt
date > data_actual.txt
```

Si el fitxer existeix, el seu contingut anterior es **perd**.

>> --- Afegir a un fitxer (no sobreescriu)

Afegeix la sortida al **final** del fitxer. Si el fitxer no existeix, el crea.

```
echo "Primera línia" > notes.txt
echo "Segona línia" >> notes.txt
echo "Tercera línia" >> notes.txt
```

Resultat de notes.txt:

```
Primera línia
Segona línia
Tercera línia
```

3. Redirecció de la sortida d'errors (stderr)

2> --- Redirigir errors a un fitxer

El descriptor 2 correspon a stderr. S'utilitza per capturar els **missatges d'error** separatament.

```
ls /directori_inexistent 2> errors.txt
find / -name "fitxer.txt" 2> /dev/null
```

2>> --- Afegir errors a un fitxer

```
./script.sh 2>> registre_errors.log
```

4. Redirecció de stdout i stderr simultàniament

> fitxer 2>&1 --- Tots dos canals al mateix fitxer

2>&1 significa "redirigeix el descriptor 2 cap on apunta el descriptor 1".

```
ls -l /etc > sortida.txt 2>&1
./script.sh > tot.log 2>&1
```

L'ordre importa: primer s'ha de redirigir stdout i després 2>&1. A l'inrevés no funciona correctament.

&> --- Sintaxi abreujada de Bash (Bash 4+)

```
ls -l /etc &> sortida_i_errors.txt
./script.sh &>> registre.log # afegint
```

Separar stdout i stderr en fitxers diferents

```
./script.sh > sortida.txt 2> errors.txt
```

5. Redirecció de l'entrada estàndard (stdin)

< --- Llegir des d'un fitxer

Substitueix el teclat per un fitxer com a font d'entrada.

```
sort < llista_desordenada.txt
wc -l < document.txt
mail -s "Assumpte" usuari@exemple.com < missatge.txt
```

<< --- **Here Document (document incrustat)**

Permet introduir múltiples línies d'entrada directament a l'interpret, fins a trobar una paraula delimitadora.

```
cat << FI
Això és un here document.
Pot tenir múltiples línies.
S'acaba quan apareix la paraula delimitadora.
FI
```

Útil en scripts:

```
mysql -u root -p base_de_dades << SQL
SELECT * FROM usuaris;
DROP TABLE temporal;
SQL
```

<<< --- Here String (cadena incrustada)

Passa una cadena de text directament com a stdin.

```
grep "hola" <<< "hola món"
bc <<< "5 * 8 + 3"
```

6. La canonada (|) --- Pipe

L'operador | connecta la **sortida estàndard** d'una ordre amb l'**entrada estàndard** de la següent. Permet encadenar ordres per crear pipelines potents.

```
ordre1 | ordre2 | ordre3
```

```
ls -l | grep ".txt"
cat fitxer.txt | sort | uniq
ps aux | grep apache | wc -l
cat /etc/passwd | cut -d: -f1 | sort
```

Pipeline amb stderr

Per defecte, la canonada només transmet stdout. Per incloure stderr:

```
./script.sh 2>&1 | grep "error"
./script.sh |& grep "error" # sintaxi abreujada Bash
```

7. El dispositiu /dev/null

/dev/null és un dispositiu especial que **descarta tot el que hi escrius** i retorna EOF quan el lleixes. S'utilitza per silenciar sortides no desitjades.

```
# Silenciar sortida estàndard
ls /etc > /dev/null

# Silenciar errors
find / -name "passwd" 2> /dev/null

# Silenciar tot
```

```
./script.sh > /dev/null 2>&1
# Buidar un fitxer sense esborrar-lo
> fitxer.txt
cat /dev/null > fitxer.txt
```

8. tee --- Escriure i mostrar alhora

L'ordre tee llegeix de stdin i escriu simultàniament a stdout i a un fitxer. És com una "T" en una canonada.

```
ls -l | tee llistat.txt
./script.sh | tee -a registre.log # -a per afegir
```

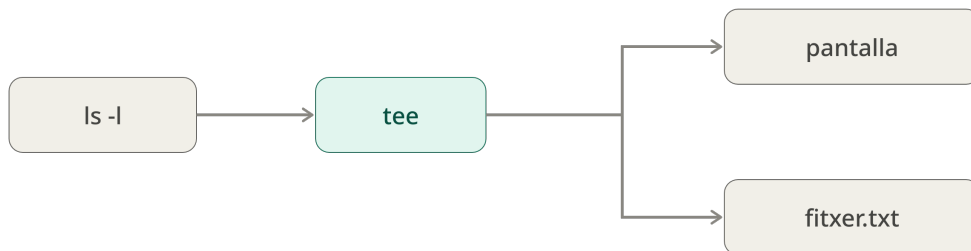


Figura 2: Escriure i mostrar alhora

9. Resum dels operadors de redirecció

Operador	Funció
>	Redirigir stdout a fitxer (sobreescriu)
>>	Afegir stdout a fitxer
<	Llegir stdin des d'un fitxer
2>	Redirigir stderr a fitxer (sobreescriu)
2>>	Afegir stderr a fitxer
2>&1	Redirigir stderr cap on va stdout
&>	Redirigir stdout i stderr a fitxer
&>>	Afegir stdout i stderr a fitxer
	Canonada: stdout d'una ordre → stdin de la següent
&	Canonada incloent stderr
<<DELIM	Here document
<<<	Here string

10. Exemples pràctics combinats

Guardar la sortida d'una ordre i veure-la a la vegada

```
df -h | tee espai_disc.txt
```

Comptar línies que contenen errors en un log

```
grep "ERROR" /var/log/syslog | wc -l
```

Ordenar i eliminar duplicats d'una llista

```
cat llista.txt | sort | uniq > llista_neta.txt
```

Executar script guardant tota la sortida (stdout + stderr)

```
./deploy.sh > deploy.log 2>&1
```

Cerca silenciosa (sense mostrar errors de permisos)

```
find /home -name "*.conf" 2> /dev/null
```

Enviar el contingut d'un fitxer per correu

```
mail -s "Informe diari" admin@empresa.cat < informe.txt
```

Generar un fitxer de configuració amb here document

```
cat > /etc/app/config.conf << EOF
servidor=192.168.1.10
port=8080
debug=false
EOF
```

Pipeline complex: els 5 processos que més CPU consumeixen

```
ps aux | sort -k3 -rn | head -6 | tail -5
```

11. Errors comuns

Sobre escriure un fitxer sense voler

```
# Perill: sobre escriu fitxer.txt!  
sort fitxer.txt > fitxer.txt      # el fitxer queda buit  
# Correcte: usar fitxer temporal  
sort fitxer.txt > fitxer_ordenat.txt && mv fitxer_ordenat.txt  
↪ fitxer.txt
```

Ordre incorrecte en 2>&1

```
# Incorrecte: stderr no va al fitxer  
./script.sh 2>&1 > sortida.txt  
  
# Correcte: primer redirigir stdout, després 2>&1  
./script.sh > sortida.txt 2>&1
```

Confondre > amb >>

```
# Cada cop que s'executa, esborra el contingut anterior  
echo "entrada" > registre.log  
  
# Acumula les entrades  
echo "entrada" >> registre.log
```

12. Protecció contra sobre escriptura (noclobber)

Bash permet activar l'opció noclobber per evitar sobre escriure fitxers existents accidentalment:

```
set -o noclobber      # activar  
set +o noclobber     # desactivar  
  
# Amb noclobber actiu, per forçar la sobre escriptura:  
echo "text" >| fitxer.txt
```

Versions d'aquest document

- HTML - [redireccions.html](#)
- PDF - [redireccions.pdf](#)
- ODT - [redireccions.odt](#)
- MD - [redireccions.md](#)

[Domini Públic \(CC0\)](#)