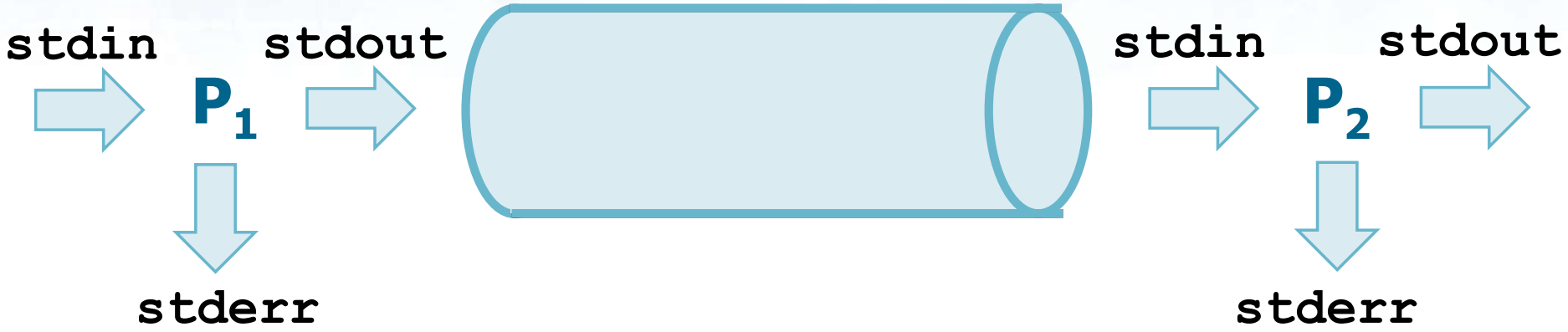
Processi

Pipe e ridirezione in UNIX/Linux

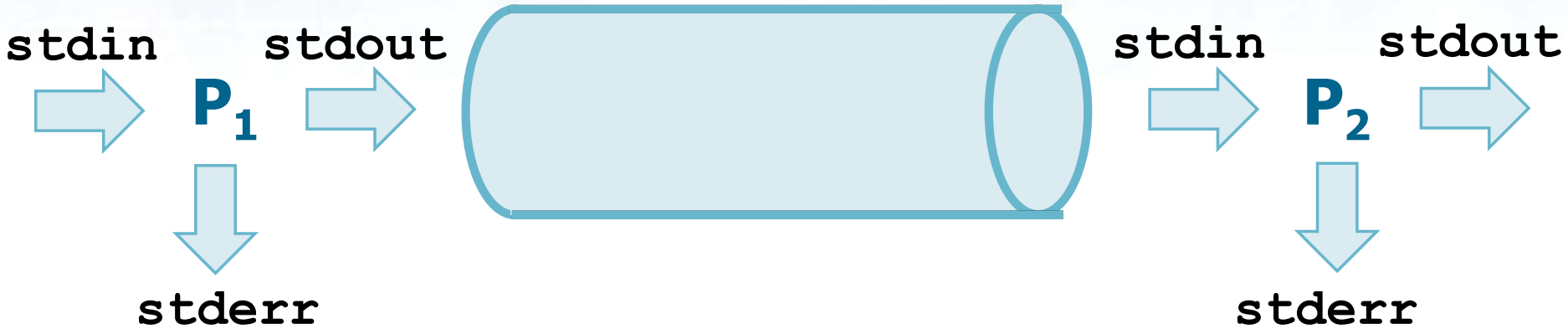
Stefano Quer
Dipartimento di Automatica e Informatica
Politecnico di Torino

Pipe



- La comunicazione tra processi attraverso pipe può essere realizzata anche quando i processi sono eseguiti mediante comandi di shell
- Le pipe a livello di comandi di shell si rappresentano con il carattere "|"

Pipe

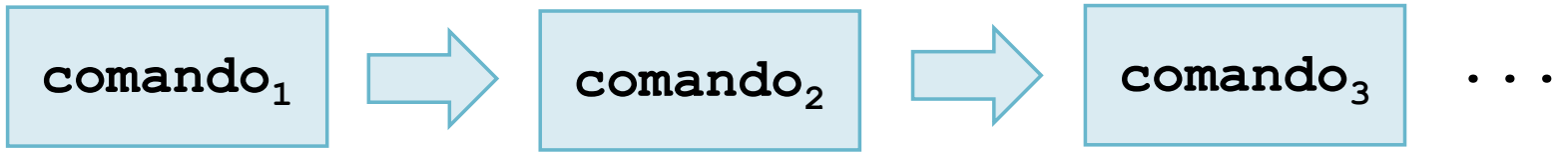


- Una pipe
  - Crea un collegamento tra lo standard output del commando precedente e lo standard input di quello successivo
  - Un pipe risiede completamente in memoria
    - La comunicazione è più veloce che quella che avviene attraverso disco e R/W da/su file

Pipe

comando$_1$ | comando$_2$

comando$_1$ | comando$_2$ | comando$_3$ ...



- Esempi

```
ls -la | more
ps | grep "main"
cat file1.txt file2.txt file3.txt | sort
ls -laR *.c | wc
```

ls -laR *.c | wc
versus
wc *.c

Redirezione dell'I/O

- Il termine redirezione indica l'associazione dei dispositivi standard a unità logiche diverse
  - Le shell prevedono all'avvio che i terminali di "default" vengano associati a tre descrittori di file
    - Standard input: stdin, 0
    - Standard output: stdout, 1
    - Standard error: stderr, 2
  - La redirezione permette a un dato commando di leggere e/o scrivere i propri dati su terminali diversi da quelli predefiniti

Redirezione dell'I/O

- Si osservi che il dispositivo

```
/dev/null
```

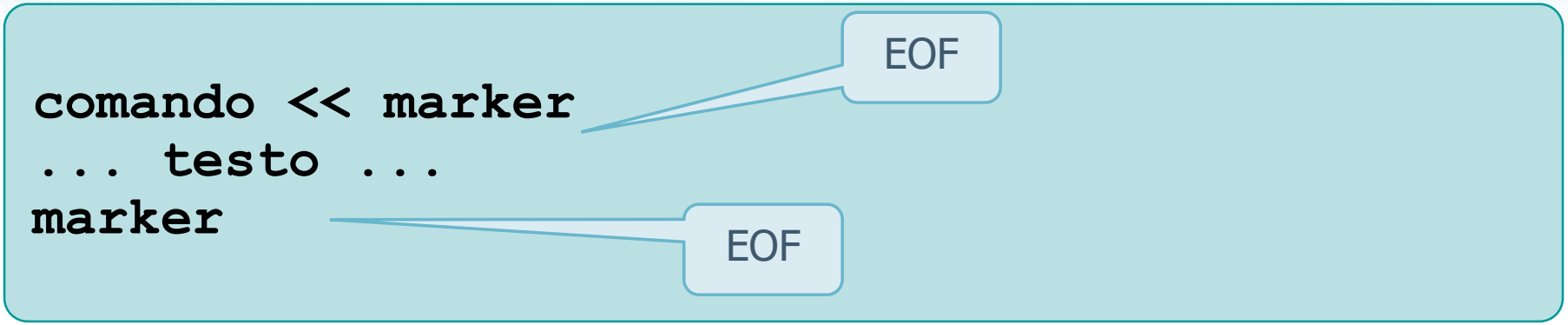
  - È un file speciale su cui si può
    - Scrivere senza accumulare byte su disco o visualizzarli a video
    - Leggere una sequenza di valori 0

Standard input

```
comando < file
```

- La redirezione dello standard input da un file si effettua con il carattere '<'

Standard input



- Redirezione dello standard input da tastiera
  - "Documento sul posto" ("here document")
  - Marker è una stringa arbitraria
    - Normalmente EOF

Standard output

```
comando  >  file
comando  1>  file
```

Default: 1> → >

- Redirezione dello standard output su un file
  - Se il file esiste viene cancellato
  - Il descrittore 1 (stdout) è quello di default
    - Quindi normalmente viene omesso
  - Esempi

```
ls -laR > file_out.
wc pgrm.c > file_pout.txt
```

Standard output

```
comando  >>  file
comando  1>>  file
```

- Accodamento dello standard output su un file
  - Se il file esiste **non** viene cancellato
  - Esempi

```
ls -laR >> file_out.
wc pgrm.c >> file_pout.txt
```

Standard error

```
comando  2>  file
comando  2>>  file
```

- Redirezione (ovvero accodamento) dello standard error sul file
  - Analogo allo stdout ma su stderr

Redirezione multipla

- Redirezione contemporanea
  - Dello standard output
  - Dello standard error
- Sullo stesso file

```
comando   &>   file
comando   &>>   file
```

- Su file diversi

```
comando   1>   fileOut   2>   fileErr
comando   >   fileOut   2>   fileErr
```

Esempio

Redirezione di stdin, stdout, stderr

```
int main () {
  char c;
  setbuf(stdout,0);
  setbuf(stderr,0);
  while (scanf ("%c", &c) == 1) {
    fprintf (stdout, "stdout:%c\n", c);
    fprintf (stderr, "stderr:%c\n", c);
  }
  return (0);
}
```

- Redirezione

```
comando | pgrm
pgrm < file
pgrm > file
pgrm 1> fileOut 2> fileErr
pgrm < fileIn 1> fileOut 2> fileErr
```