Esercitazione 6

pipe e esercizio riassuntivo

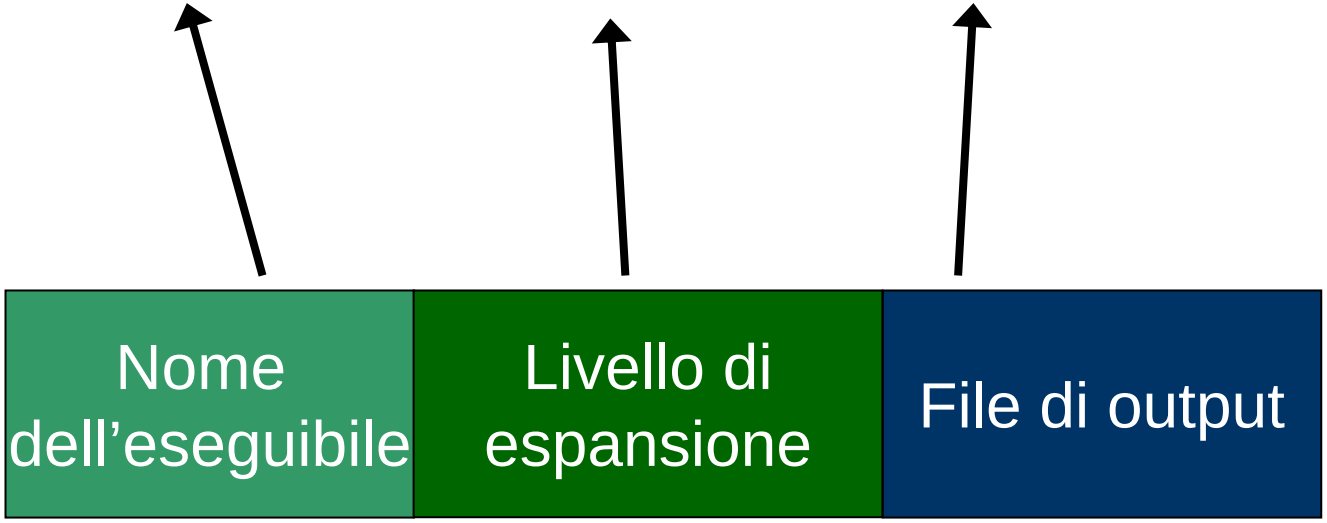
Parte 1

PIPE

Obiettivo 1/2

- Scrivere un comando che abbia la sintassi

```
tartaglia <level> <output>
```



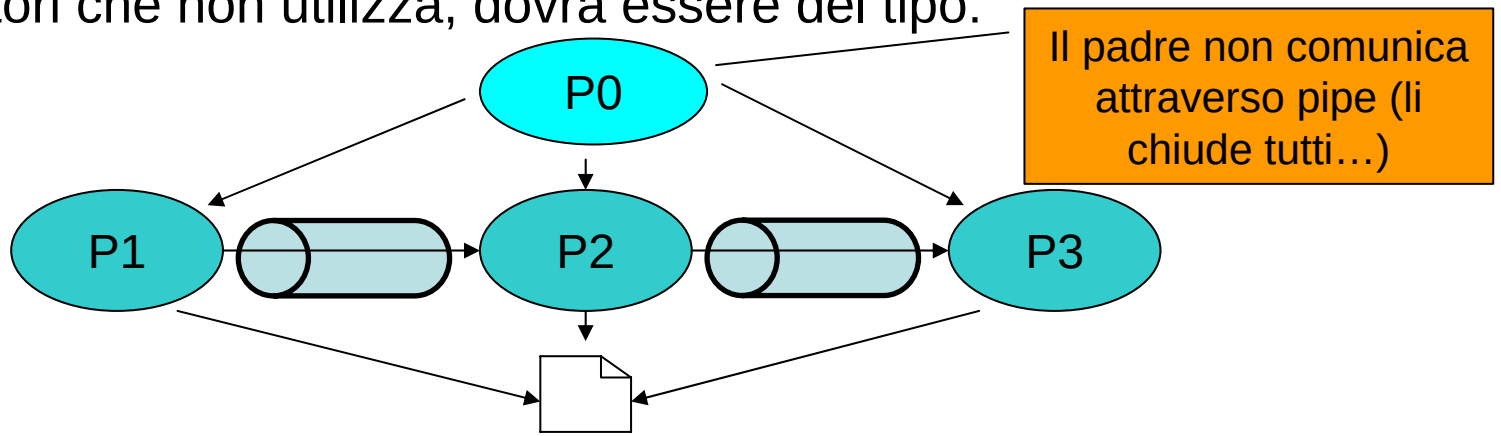
Il comando deve scrivere nel file di output il triangolo di Tartaglia fino al livello di espansione dato

Obiettivo 2/2

- In particolare:
  - il processo padre crea N figli, tanti quanti i livelli di espansione richiesti (si faccia l'ipotesi esemplificativa di al massimo MAX figli, con MAX costante)
  - Il figlio i-esimo è associato all'i-esimo livello; in particolare, ogni figlio legge l'output del figlio precedente (ovvero il contenuto del livello i-1) e produce il contenuto del livello i scrivendolo sul file di output e su una pipe con cui comunica con il figlio i+1
  - i figli comunicano "tra fratelli" con delle pipe!
  - ECCEZIONI:
    - Il primo figlio parte con 1 (primo livello del triangolo)
    - L'ultimo figlio scrive solamente sul file di output passato come argomento

Note e suggerimenti 1/2

- Sarà il padre, ovviamente, a predisporre le pipe necessarie, ma ricordatevi che ogni processo, COME PRIMA COSA, dovrà chiudere i descrittori che non gli competono (tutti meno il descrittore di una pipe in lettura e il descrittore di una pipe in scrittura)
- Con N figli, dovranno essere predisposte N-1 pipe, ma ogni figlio utilizzerà solo 2 lati di due pipe (fatta eccezione del primo e ultimo figlio)
- Inoltre, dovrà essere disponibile per tutti i figli il descrittore del file di output
- La situazione con 3 figli, una volta che ogni processo chiude i descrittori che non utilizza, dovrà essere del tipo:



Note e suggerimenti 2/2

- Stavolta quindi, il codice dei figli è uguale, fatta eccezione per
  - Il livello di competenza
  - Quali descrittori vanno chiusi (dipende dalla posizione del figlio)
- Conviene tenere e manipolare una struttura dati di descrittori del tipo

Pipe1[0]	…	PipeN-1[0]	X
Pipe1[1]	…	PipeN-1[1]	Output_file

Descrittori di lettura

Descrittori di scrittura

~~Pipe1[0]~~	~~…~~	~~PipeN-1[0]~~	X
Pipe1[1]	~~…~~	~~PipeN-1[1]~~	Output_file

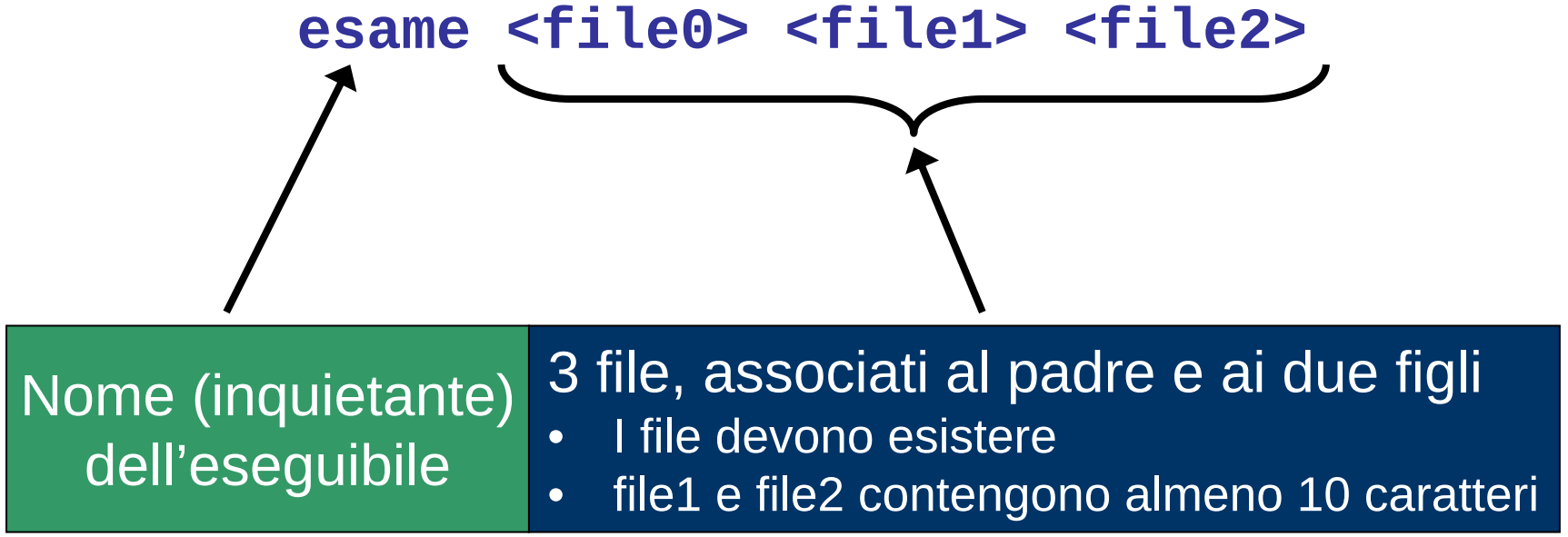
Esempio: primo figlio

Parte 2

ESERCIZIO RIASSUNTIVO

Obiettivo

- Scrivere un comando che abbia la sintassi



Funzionamento

- L’esecuzione si basa su 3 processi: il padre (associato a file_0) e 2 figli (associati a file_1 e file_2)
- I figli devono leggere i primi 10 caratteri del file a cui sono associati e salvarli in un buffer
- Una volta terminata l’operazione precedente, il padre comincia a leggere i caratteri contenuti in file_0 e a COMUNICARLI uno per uno a entrambi i figli
- Non appena un figlio riceve un carattere presente nel proprio buffer, invia un segnale al padre e termina
- Il padre risponde
  - stampando il pid del processo che gli ha inviato il segnale
  - terminando l’altro figlio
  - terminando a sua volta
- I figli devono comunque terminare una volta che il padre ha finito di leggere il proprio file; il padre ne deve rilevare lo stato di terminazione

Note e suggerimenti

- I figli devono avvertire il padre che hanno finito di leggere il proprio file, il padre continua solo dopo che ENTRAMBI l'hanno fatto
- Come comunica il padre coi figli? Se uso le pipe, basta una pipe sola?
- Come faccio a capire qual è il figlio che eventualmente mi ha mandato il segnale? Tenete presente che stavolta uso per forza lo stesso segnale per entrambi i figli, l'altro infatti l'ho già impiegato (di nuovo, uguale per entrambi i figli) nel realizzare il meccanismo dell'avvertimento iniziale…