

## C - Comunicazione tra processi

### Esercizio 1

Si scriva un programma C in cui, dato un vettore di interi, il processo padre genera due processi figli; il primo calcolerà il valore massimo e lo stamperà su stdout, il secondo calcolerà il valore minimo e lo stamperà su stdout. Il processo padre invece calcolerà il valore medio stampandolo su stdout; prima di terminare attenderà che entrambi i processi figli abbiano concluso l'esecuzione.

### Esercizio 2

Si scriva un programma C che abbia due processi, parent e child. Il processo padre esegue per  $N$  volte il seguente ciclo:

- scrive una riga su stdout
- dorme un secondo
- comunica al processo figlio che può continuare
- si ferma fin quando il processo figlio non gli comunica di continuare

Il processo figlio esegue all'infinito il seguente ciclo:

- attende che il processo padre gli comunichi di partire
- scrive una riga su stdout
- dorme un secondo
- comunica al processo padre che può continuare

Quando il processo padre ha scritto la  $N$ -esima riga invia un segnale di terminazione al processo figlio, aspetta che termini e poi termina anch'esso.

Alla fine dell'esecuzione, con ad esempio  $N = 2$ , si avrà il seguente output:

```
[parent] data=01
[child ] data=01
[parent] data=02
[child ] data=02
```

### Esercizio 3

Si scriva un programma C, utilizzando una o più pipe, in cui il processo padre origina due processi figli A e B. Il processo A «produce» una stringa RA e il processo B «produce» una stringa RB. Il processo padre aspetta la conclusione dei due processi figli; quando entrambi hanno concluso l'esecuzione, concatena RA e RB e scrive il risultato su stdout.

**Esercizio 4**

Si scriva un programma C in cui il processo padre crea  $N$  processi figli; il processo figlio  $i$ -esimo calcola  $fibonacci(i)$  e lo inserisce all'interno di una memoria condivisa. Il processo padre aspetta che tutti i figli abbiano terminato per poi scrivere su stdout i valori di fibonacci restituiti dai processi figli.