

Sistemi Operativi – a.a. 2019/2020

prova di laboratorio in aula

– 14 luglio 2020 –

Creare un programma **pari-dispari-generalizzato.c** in linguaggio C che accetti invocazioni sulla riga di comando del tipo:

pari-dispari-generalizzato <n=numero-giocatori> <m=numero-partite>

Il programma gestisce una serie di $m \geq 1$ partite tra $2 \leq n \leq 6$ giocatori virtuali (processi) che giocano una variante generalizzata del gioco "Pari & Dispari". Il programma padre **J** fungerà da giudice e, verificata la correttezza dei parametri in input, creerà inizialmente gli n processi giocatori P_i con $0 \leq i \leq n-1$ e una coda di messaggi che tutti utilizzeranno per comunicare. Per indirizzare un messaggio ad uno specifico processo si dovrà utilizzare i tipi numerici previsti dallo specifico strumento di IPC.

Gli $n+1$ processi opereranno secondo le seguenti regole:

- il giudice all'inizio di ogni partita chiede ad ogni singolo giocatore di fare la propria mossa usando un apposito messaggio;
- ogni giocatore P_i risponde inviando al giudice la propria mossa: un intero $0 \leq x_i \leq 9$ scelto a caso;
- ricevute le mosse da tutti i giocatori, il giudice stabilisce se si tratta di una partita patta o se c'è un vincitore: la partita è da ritenersi patta, e quindi da reiterare, se tra le mosse dei giocatori c'è qualche duplicato; diversamente il giocatore vincitore è P_j dove $j = (x_1 + \dots + x_n) \bmod n$;
- quando il giudice determina che sono state svolte tutte le partite (con vincitore) richieste, segnala a tutti i giocatori di auto-terminarsi usando un apposito messaggio; fatto ciò visualizza il numero di partite vinte da ogni giocatore e l'eventuale vincitore assoluto.

I processi dovranno tutti terminare spontaneamente alla fine del torneo, rilasciando correttamente le risorse di IPC persistenti.

Un possibile output del programma potrebbe essere il seguente:

```
$ pari-dispari-generalizzato 3 10

J: inizio partita n.1
P0: mossa 4
P2: mossa 0
P1: mossa 3
J: partita n.1 vinta da P1
J: inizio partita n.2
P0: mossa 9
P1: mossa 9
P2: mossa 3
J: partita n.2 patta e quindi da ripetere
J: inizio partita n.2
P0: mossa 3
P1: mossa 2
P2: mossa 9
J: partita n.2 vinta da P2

[...]

J: inizio partita n.10
P1: mossa 1
P0: mossa 0
P2: mossa 5
J: partita n.9 vinta da P0
J: classifica finale: P0=2 P1=3 P2=5
J: vincitore del torneo: P2
```

Tempo: 3 ore

Note importanti sulla valutazione:

- è **obbligatorio ricopiare in bella**: questa sarà l'unica copia da consegnare e valutata; contingentare il tempo di conseguenza;
- la copia in bella dovrà riportare la struttura completa di un programma in C funzionante;
- oltre alla semantica e alla sintassi del codice verrà valutata anche la leggibilità e la corretta indentazione dello stesso;
- per agevolare la leggibilità del codice è possibile scrivere i nomi delle variabili, funzioni e chiamate di libreria/sistema in completo maiuscolo.