

## **ESERCIZIO JS 2:**

### **Parte A**

Si realizzi un'infrastruttura client-server in cui

- I due processi comunicano usando i servizi del protocollo **TCP**
- Il client legge da tastiera un operatore tra '+', '-', '\*', '/' (operazione)
- Il client legge da tastiera due numeri interi (operandi)
- Il client invia al server un messaggio contenente nell'ordine <operazione, 1° numero, 2° numero>
- Il server interpreta il messaggio, esegue l'operazione richiesta e invia il risultato (che può essere *double...*) al client, quindi termina
- Il client stampa a video il risultato e termina.

### **Parte B**

Si realizzi un'infrastruttura client-server in cui

- I due processi comunicano usando i servizi del protocollo **UDP**
- Il client legge da tastiera un operatore tra '+', '-', '\*', '/' (operazione), oppure '.' (terminazione)
- Se l'input precedente è diverso da '.', il client legge da tastiera due numeri interi (operandi)
- Il client invia al server un messaggio contenente nell'ordine <operazione, 1° numero, 2° numero>, oppure il singolo '.'
  - Se il client ha letto e inviato '.', poi termina
- Il server interpreta il messaggio
  - Se esso contiene una richiesta di operazione, la esegue e invia il risultato al client
  - Se esso contiene '.', allora il server termina
- Se il client ha inviato un'operazione, allora stampa a video il risultato ricevuto dal server e poi torna a leggere da tastiera una nuova operazione.

### **Parte C**

Si realizzi un'infrastruttura client-server in cui

- I due processi comunicano usando i servizi del protocollo **TCP**
- Il server gestisce più client implementando un'architettura di tipo **iterativo**
- Il client legge da tastiera un operatore tra '+', '-', '\*', '/' (operazione), oppure '.' (terminazione)
- Se l'input precedente è diverso da '.', il client legge da tastiera due numeri interi (operandi)
- Il client invia al server un messaggio contenente nell'ordine <operazione, 1° numero, 2° numero>, oppure il singolo '.'
  - Se il client ha letto e inviato '.', poi termina
- Il server interpreta il messaggio, stampa l'indirizzo del client da cui lo ha ricevuto, e:
  - Se esso contiene una richiesta di operazione, la esegue e invia il risultato al client
  - Se esso contiene '.', allora il server conclude la comunicazione con il client attuale e si mette a disposizione per servire un nuovo client
- Se il client ha inviato un'operazione, allora stampa a video il risultato ricevuto dal server e poi torna a leggere da tastiera una nuova operazione.

### **Parte D**

Si realizzi un'infrastruttura client-server in cui

- I due processi comunicano usando i servizi del protocollo **TCP**
- Il server gestisce più client implementando un'architettura di tipo **concorrente multi-thread**
- Il client legge da tastiera un operatore tra '+', '-', '\*', '/' (operazione), oppure '.' (terminazione)
- Se l'input precedente è diverso da '.', il client legge da tastiera due numeri interi (operandi)
- Il client invia al server un messaggio contenente nell'ordine <operazione, 1° numero, 2° numero>, oppure il singolo '.'

- Se il client ha letto e inviato '.', poi termina
- Il server interpreta il messaggio, stampa l'indirizzo del client da cui lo ha ricevuto, e:
  - Se esso contiene una richiesta di operazione, la esegue e invia il risultato al client
  - Se esso contiene '.', allora il server conclude la comunicazione con il client attuale (e continua con il suo comportamento abituale nei confronti di altri client)
- Se il client ha inviato un'operazione, allora stampa a video il risultato ricevuto dal server e poi torna a leggere da tastiera una nuova operazione.

### Parte E

Si realizzi un'infrastruttura client-server in cui

- I due processi comunicano usando i servizi del protocollo **UDP**
- Il server gestisce più clienti e non termina mai
- Il client legge da tastiera un operatore tra '+', '-', '\*', '/' (operazione), oppure '.' (terminazione)
- Se l'input precedente è diverso da '.', il client legge da tastiera due numeri interi (operandi)
- Il client invia al server un messaggio contenente nell'ordine <operazione, 1° numero, 2° numero>, oppure il singolo '.'
  - Se il client ha letto e inviato '.', poi termina
- Il server interpreta il messaggio, stampa l'indirizzo del client da cui lo ha ricevuto, e:
  - Se esso contiene una richiesta di operazione, la esegue e invia il risultato al client
  - Se esso contiene '.', allora il server *non fa nulla*
- Se il client ha inviato un'operazione, allora stampa a video il risultato ricevuto dal server e poi torna a leggere da tastiera una nuova operazione.

**PROVE DI ESECUZIONE:** nei casi C, D, E far partire 2 client; ogni volta che un client termina, farne partire uno nuovo. Osservare il comportamento delle tre infrastrutture in termini di (i) tempi di attesa osservati dai client per operare; (ii) possibile interleaving tra client molteplici.