

Tecnologie di progettazione di sistemi informatici e di telecomunicazioni

VOTO:

La seguente verifica deve essere svolta entro la scadenza indicata dal docente.

L'intervallo dei voti è [1-8] e ogni domanda ha un punteggio massimo pari a 1. La sufficienza si raggiunge totalizzando 4,5 punti su 6 punti totali.

Le risposte dovranno essere scritte direttamente sul documento, sotto la rispettiva domanda.

Per l'assegnazione dei punti saranno considerati i seguenti criteri:

- aderenza alla traccia, correttezza logica e completezza della soluzione/risposta;
- presentazione precisa e ordinata dell'elaborato;
- adeguatezza ed efficacia comunicativa;
- uso appropriato dei software per la stesura e la consegna della verifica.

1. Di seguito viene riportata una porzione del codice di un programma che utilizza i thread. Fornisci una spiegazione esaustiva per ognuno dei seguenti punti.
 - a. Spiega qual è la funzione delle istruzioni nelle righe 23, 24, 36.
 - b. Spiega se la classe **GroupMultiplierTask** estende la classe **Thread** o implementa l'interfaccia **Runnable**.
 - c. Spiega a cosa serve in generale l'istruzione **Thread.currentThread().getName()**.
 - d. Spiega a cosa serve in genere il metodo **run()** implementato in una classe.

```
01 public class ParallelProcessorMultiplier extends MatrixGenerator {
02     public ParallelProcessorMultiplier (int rows, int columns) {
03         super(rows, columns);    }
04
05     @Override
06     public void multiply(MatrixGenerator sndMatrix) {
07         List<Thread> threads = new ArrayList<>();
08         int rowsFstMatrix = this.getRows();
09         double[][] result =
10             new double[this.getRows()][sndMatrix.getColumns()];
11
12         int numThreads = Runtime.getRuntime().availableProcessors();
13         System.out.println("Number of processors: " + numThreads);
14         int startIndex, endIndex, step;
15         step = rowsFstMatrix / numThreads;
16         startIndex = 0;
17         endIndex = step;
18
19         for (int i = 0; i < numThreads; i++) {
20             GroupMultiplierTask task =
21                 new GroupMultiplierTask(result, this, sndMatrix, startIndex,
22                                         endIndex);
23             Thread thread = new Thread(task);
24             thread.start();
25             threads.add(thread);
26             startIndex = endIndex;
27             endIndex = ((i == (numThreads - 2)) ? rowsFstMatrix :
28                       (endIndex + step));
29         }
```

```

30         waitForThreads(threads);
31     }
32
33     private static void waitForThreads (List<Thread> threads) {
34         for (Thread thread : threads) {
35             try {
36                 thread.join();
37             } catch (InterruptedException e) {
38                 e.printStackTrace();
39             }
40         }
41         threads.clear();
42     }
43 }

```

2. Considerando l'implementazione di un generico protocollo di comunicazione client-server, descrivi le differenze principali tra un server mono-thread e un server concorrente multi-thread.
3. Supponi di dover scrivere il codice di un server che usi a livello di trasporto i servizi offerti dal protocollo TCP, rispondi in modo esaustivo alle seguenti domande.
 - a. Scrivi l'istruzione necessaria affinché il server possa creare un socket utilizzando un determinato numero di porta.
 - b. Scrivi la corrispondente istruzione che dovrà essere presente nel client affinché venga creato un socket per connettersi successivamente a questo server, indicando i parametri che dovranno essere passati e spiegandone il significato.
 - c. Scrivi l'istruzione necessaria affinché il server possa accettare una richiesta di connessione da parte di un client.
 - d. Spiega cosa avviene a livello di trasporto in corrispondenza dell'esecuzione delle istruzioni indicate nel punto (c).
 - e. Spiega cosa viene creato a livello di trasporto in corrispondenza della seguente riga di codice nel server:

```

PrintWriter out = new PrintWriter(new OutputStreamWriter(
                                connection.getOutputStream()));

```
4. Spiega perché le seguenti association identificano comunicazioni diverse:
 - a. TCP, 200.100.50.20, 2780, 150.75.31.5, 50123
 - b. UDP, 200.100.50.20, 2780, 150.75.31.5, 50123
 - c. TCP, 200.100.50.20, 2780, 150.75.31.5, 50200
 - d. TCP, 200.100.50.20, 2780, 150.75.31.3, 50123
5. Supponi di dover organizzare il sito Web di una banca per l'home banking dei clienti, indica tra le seguenti opzioni quali sceglieresti e motiva accuratamente tutte le tue scelte. Per ogni punto (a, b, c, d) devi sicuramente scegliere almeno un'opzione, in alcuni punti potrebbero essere scelte anche più opzioni.
 - a. Tecnologia Web: client-side, server-side.
 - b. Tipologia di servizi offerti dallo strato di trasporto: trasferimento di dati affidabile, ampiezza di banda, temporizzazione, sicurezza.
 - c. Tipologia di comunicazione: unicast, multicast.
 - d. Architettura tier: 1-tier, 2-tier, 3-tier, n-tier.

6. Relativamente alla domanda 5 spiega perché è stata scelta un'architettura di rete client-server invece di una architettura peer-to-peer.