

## Sistemi di Calcolo 2 (A.A. 2019-2020 e seguenti)

### Prova congiunta Sistemi di calcolo 12 CFU - Seconda parte (A.A. 2017-2018 e precedenti)

*Appello straordinario - 12 Ottobre 2021*

Tempo a disposizione: 1h 45m.

Attenzione: assicurarsi di compilare il file **studente.txt** e che il codice prodotto non contenga **errori di compilazione**, pena una non correzione dell'elaborato.

#### Regole Esame

- Domande ammesse  
Le domande possono riguardare solo la specifica dell'esame e la struttura di alto livello del codice, nessuna domanda può riguardare singole istruzioni.
- Oggetti vietati  
I seguenti oggetti non devono essere presenti sulla scrivania, né tantomeno usati: smartphone, smartwatch, telefonini, tablet, portatili, dispositivi di archiviazione USB, copie cartacee della dispensa, astucci e qualsiasi forma di libri ed appunti. **Chi verrà sorpreso ad usare uno di questi oggetti verrà automaticamente espulso dall'esame.**
- Modalità di risposta  
Le risposte alle domande di teoria vanno fornite nei file txt indicati. Il professore fornisce fogli per appunti e dove fornire illustrazioni o codice utile ad integrazione delle risposte di teoria. Qualsiasi altro foglio portato dallo studente non può essere usato.
- Azioni vietate  
È assolutamente vietato comunicare in qualsiasi modo con gli altri studenti. **Chi verrà sorpreso a comunicare con gli altri studenti per la prima volta verrà richiamato, la seconda volta verrà invece automaticamente espulso dall'esame.**

### Teoria 1 - Comunicazione di rete (rispondere nel file teoria1.txt)

Descrivere ad alto livello il funzionamento di un network adaptor (scheda di rete) spiegando come si interfaccia con il processore

### Teoria 2 - Algoritmi di concorrenza (rispondere nel file teoria2.txt)

Considerate il seguente algoritmo di Dijkstra "modificato":

---

```
Initially
    /* global info */
    boolean interested[N] = {false, ..., false};
    boolean passed[N] = {false, ..., false};
    int k = <any> ;          //  $k \in \{0, 1, \dots, N-1\}$ 
    /* local info */
    int i = <entity ID>;    //  $i \in \{0, 1, \dots, N-1\}$ 
repeat
    1  <Non Critical Section>;
    2  interested[i] = true;
    3  while (k != i) {
    4      passed[i] = false;
    5      if (!interested[k]) then k = i;
    6  }
    7  passed[i] = true;
    8  for j in 0 ... N-1 except i do
    9      if (passed[j]) then goto 3;
    10 interested[i] = false; passed[i] = false;
    11 <critical section>;
forever
```

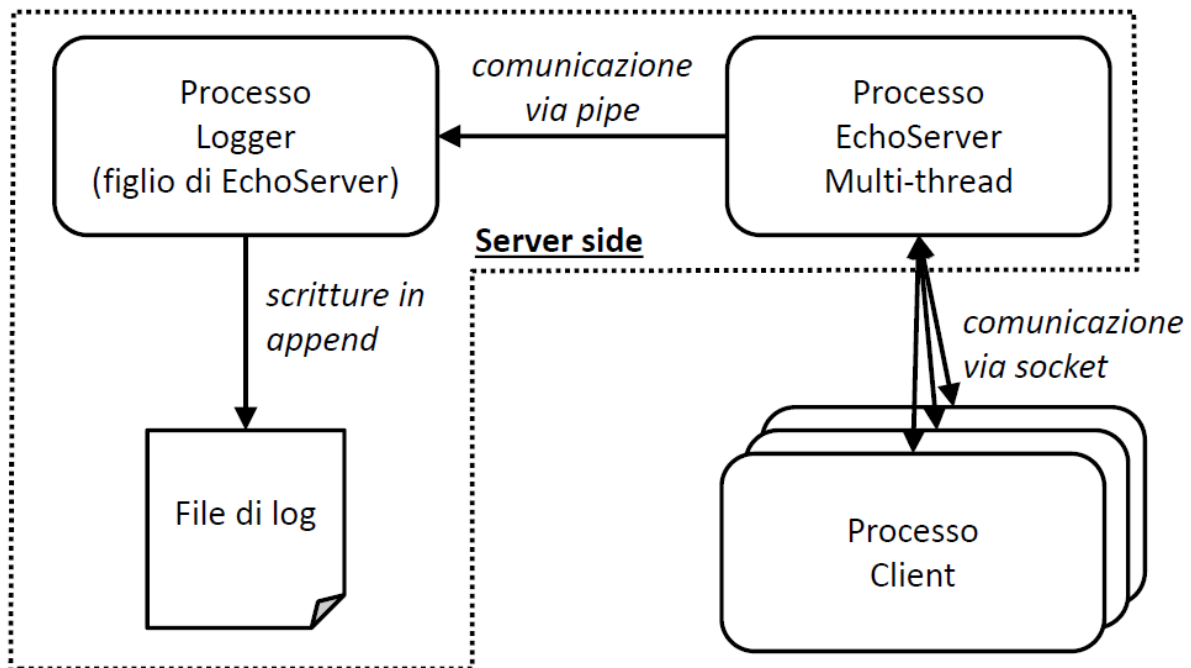
---

L'algoritmo di Dijkstra è stato modificato invertendo la riga 10 con la riga 11

- A. Descrivere cosa comporta tale modifica nell'esecuzione del programma
- B. Quale o quali sono le problematiche che si creano legate alle proprietà di Mutua Esclusione, No-Deadlock e No-Starvation? Se e quali vengono violate? Motivare la risposta.

### Programmazione - Realizzazione di un server multi-thread a concorrenza controllata

Un echo server multi thread si avvale di un processo logger per memorizzare tutte le operazioni eseguite all'interno di un file. Il logger è un componente software che scrive su un file tutta una serie di informazioni rilevanti per l'applicazione e viene usato in genere per il monitoring delle applicazioni server e fornisce utili informazioni su eventuali problemi occorsi a runtime. Il server istanzia un processo figlio dedicato alle attività di logging e gli comunica ogni informazione da memorizzare tramite una pipe, su cui reindirizza il proprio canale stderr. In tal modo ogni messaggio che viene inviato a stderr, viene in realtà "trasferito" alla pipe e scritto sul file log, secondo lo schema in figura:



Completare le parti identificate all'interno del file server.c

#### Obiettivi dell'esercizio

1. apertura e gestione di una pipe
2. scambio di messaggi tra processi mediante la pipe
3. creazione di thread

**Attenzione:** per verificare la corretta esecuzione è necessario aprire il file di log creato nella cartella, poichè il server non mostra nessuna stampa a schermo.

#### **Altro**

- i commenti nel codice contengono molte informazioni utili per lo svolgimento della prova, si consiglia quindi di tenerli in debita considerazione
  - il file `dispensa.pdf` contiene una copia della dispensa *Primitive C per UNIX System Programming* preparata dai tutor di questo corso
  - il file `raccomandazioni.pdf` contiene una serie di considerazioni sugli errori riscontrati più di frequente
-