

Nel seguito sono riportate le istruzioni per la creazione del progetto di Laboratorio di Sistemi Operativi per il corso di Laurea in Informatica.

Gli studenti devono seguire tutte le indicazioni riportate sotto. Per tutto quanto non riportato, gli studenti possono decidere in autonomia come procedere.

Il progetto consiste nello sviluppo di un client-server. Il server deve essere sviluppato in linguaggio C, mentre siete liberi di usare un linguaggio diverso per il client (es. C, Java, Kotlin, etc..).

A prescindere dal linguaggio di implementazione del client, Client e Server devono comunicare attraverso Socket. Client e Server possono essere locali, non è richiesto averli in remoto, ma è consentito.

Differenze per ordinamento:

- Per il nuovo ordinamento: bisogna sviluppare il progetto usando Docker-compose o ADE.
- Per il vecchio ordinamento: non è necessario usare un docker.

Il progetto deve essere accompagnato da:

- un README file con le istruzioni per eseguire il progetto
- una documentazione di massimo 10 pagine con la descrizione delle scelte implementative

Il progetto verrà poi presentato e il suo funzionamento dimostrato previo appuntamento, dopo la consegna del progetto. La presentazione del progetto deve essere effettuata da tutti i componenti del progetto nello stesso giorno.

Il progetto deve essere consegnato al docente tramite email o su Teams come file zip contenente client, server, README e breve documentazione. Se per lo sviluppo del progetto è stato usato GIT, si può, in alternativa, anche dare accesso al repository che però dovrà contenere tutto il materiale richiesto.

La discussione del progetto viene stabilita dopo la consegna del progetto.

Il progetto si ritiene superato con un minimo di 18/30.

I gruppi possono essere formati da un minimo di 2 componenti e un massimo di 3 componenti.

Qualsiasi altro tipo di formazione per eventuali esigenze straordinarie deve essere valutata e approvata dal docente. In queste situazioni, i progetti non subiranno variazioni di complessità.

La creazione dei gruppi deve essere fatta tramite form di google accessibile al seguente link ->

<https://forms.gle/xK6yfWfC3QjgCGbn6>

da un solo componente del gruppo. Nel form si dovranno specificare le seguenti informazioni:

- Nome del gruppo
- Ordinamento: vecchio o nuovo
- Progetto scelto
- Per ogni componente (almeno 2 e massimo 3):
 - Nome
 - Cognome
 - Matricola
 - Email

Progetto 1: Il supermercato

Lo studente dovrà realizzare la simulazione di un sistema che modella un supermercato con K casse e frequentato da un certo numero di clienti. Il numero dei clienti nel supermercato è contingentato: non ci possono essere più di C clienti che fanno acquisti (o che sono in coda alle casse) in ogni istante. All'inizio, tutti i clienti entrano contemporaneamente nel supermercato, successivamente, non appena il numero dei clienti scende a $C-E$ ($0 < E < C$), ne vengono fatti entrare altri E . Ogni cliente spende un tempo variabile T all'interno del supermercato per fare acquisti, quindi si mette in fila in una delle casse che sono in quel momento aperte ed aspetta il suo turno per "pagare" la merce acquistata. Dopo aver pagato, il cliente esce dal supermercato. Ogni cassa attiva ha un cassiere che serve i clienti in ordine FIFO con un certo tempo di servizio. Il tempo di servizio del cassiere ha una parte costante (diversa per ogni cassiere) più una parte variabile che dipende linearmente dal numero di prodotti acquistati dal cliente che sta servendo. I clienti che non hanno acquistato prodotti ($P=0$), non si mettono in coda alle casse.

- Opzionale: i clienti senza acquisti non si mettono in coda, ma prima di uscire dal supermercato devono attendere il permesso di uscire. Il permesso può essere dato da un direttore del supermercato.
- Opzionale: la scelta degli acquisti di un cliente può essere fatta o in ordine casuale oppure scelta dal cliente su una semplice interfaccia grafica/testuale

Progetto 2: Il traduttore

Lo studente dovrà realizzare la simulazione di un sistema che modella un traduttore per un numero non precisato di utenti. Il vocabolario deve avere la seguente struttura: <parola> <traduzione>. Gli utenti devono potersi registrare, specificando la propria lingua di appartenenza, e loggare al server. Gli utenti possono accedere a delle stanze in base alla lingua nella quale vogliono parlare, e possono messaggiarsi con altri utenti. I loro messaggi devono essere tradotti nella lingua impostata per la stanza. Per esempio, se la stanza A è per la lingua inglese, i messaggi di un utente Luca (italiano) verranno automaticamente tradotti dal server in inglese, mentre i messaggi di un utente John (inglese) verranno direttamente inviati. Ogni stanza può avere al massimo un numero K di utenti per volta. Utenti che si vogliono inserire nella stanza, vengono messi in coda fin quando un altro utente non lascia la stanza. Gli utenti vengono inseriti nella stanza con una logica FIFO. Utenti inattivi per E secondi, vengono cacciati dalla stanza.

- Il dizionario si può realizzare come meglio si crede (es., file di testo, json, database sql o a grafo, etc..)
- Registrazione può essere sviluppata tramite database sql o file
- La traduzione avverrà dall'italiano all'inglese
 - Opzionale: aggiunta di un'ulteriore lingua (es., francese o spagnolo)
- Opzionale: le stanze possono essere create dagli utenti

Progetto 3: La libreria

Lo studente dovrà realizzare la simulazione di un sistema che modella una libreria per un numero non precisato di utenti. Gli utenti devono potersi registrare e loggare al server. Il libraio ha una lista di libri disponibili nella libreria, con il numero di copie disponibili, numero di copie prese in prestito, e per ognuno delle copie prese in prestito un nome o identificativo del cliente che le ha prese in prestito, un timestamp del giorno in cui è stato preso in prestito, e una data per la restituzione del libro. Gli utenti possono cercare un libro e possono prenderlo in prestito se ci sono ancora copie disponibili. Gli utenti possono prendere in prestito un numero K di libri in prestito, dove K è deciso dal libraio. Gli utenti mettono nel carrello tutti i libri presi in prestito e poi possono effettuare il check-out. Se un altro utente nel frattempo ha preso l'ultima copia disponibile di quel libro, l'altro utente riceverà un messaggio. Il libraio dovrà contattare con un messaggio un cliente che non ha restituito il libro dopo la scadenza prevista per la restituzione.

- Registrazione può essere sviluppata tramite database sql o file
- I libri possono essere gestiti in un database, json, etc..
- Opzionale: La ricerca di un libro può essere fatta per genere, nome oppure da lista di quelli disponibili

Progetto 4: La partita

Lo studente dovrà realizzare la simulazione di un sistema che modella una partita di calcetto. Per ogni partita ci sono due squadre, A e B, e un arbitro. Ogni squadra ha 5 giocatori. Tutti i giocatori in loop cercano di accedere ad un pallone, chi riesce ad impossessarsi del pallone, inizia a giocare la partita. La simulazione della partita può avvenire secondo i seguenti eventi:

- infortunio -> il giocatore si infortuna e quindi lascia la partita per un numero casuale di minuti
- tiro -> il giocatore tira in porta. In maniera casuale farà goal. L'arbitro comunicherà il goal
- dribbling -> come per il tiro, questo avviene in maniera casuale. Se il dribbling riesce, allora può tentare nuovamente un altro tiro, altrimenti il pallone viene liberato e può essere preso da un altro giocatore

L'arbitro crea un file di log con gli eventi della partita, inclusi numero di goal, infortunio (associato al giocatore e squadra), numero tiri falliti, numeri di dribbling. La partita terminerà dopo N tempo. Le squadre vengono create da un capitano (client), il quale accetta nella sua squadra altri 4 client. Sia il capitano che gli altri giocatori partecipano alla partita.

- Opzionale: l'infortunio coinvolge due giocatori: il giocatore infortunato, e un giocatore della squadra opposta scelto in maniera random. Il giocatore che ha fatto il fallo, avrà una penalizzazione di P tempo.