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.
- Per il vecchio ordinamento: non è necessario usare un docker. Sia client che server in C.

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/pbdxJNvvKoiqD2us8
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: La partita di Tris

Lo studente dovrà realizzare un server multi-client per giocare a Tris (con due giocatori per ogni partita). Un giocatore può creare una o più partite, ma può giocare solo ad una partita alla volta. Il creatore di una partita può accettare o rifiutare la richiesta di partecipazione alla partita da un nuovo giocatore. Gli stati di gioco possono essere terminata, in corso, in attesa, nuova creazione. Gli stati di terminazione di partita possono essere vittoria, sconfitta, pareggio rispetto al giocatore. In base allo stato di gioco di ogni singola partita, tutti i giocatori collegati al server dovranno ricevere un messaggio diverso. Per esempio, "in attesa" tutti i giocatori vengono invitati a partecipare alla partita. Le partite devono essere identificate in maniera univoca. A fine partita (terminata), i giocatori di ogni partita possono scegliere se iniziare o meno un'altra partita.

- Opzionale: Il vincitore di una partita può decidere se fare un'altra partita - in questo caso, se non era il proprietario, diventa il proprietario della partita e attende un nuovo giocatore. Il perdente deve lasciare la partita. Se c'è pareggio, entrambi i giocatori possono decidere se farne un'altra.

## Progetto 2: Telefono senza fili

Lo studente dovrà realizzare la simulazione di un sistema che modella il gioco Telefono senza fili tra giocatori di diverse lingue. Gli utenti devono potersi registrare, specificando la propria lingua di appartenenza, e loggare al server. Gli utenti possono accedere a delle stanze di gioco nella quale vogliono giocare, e possono mandare messaggi con altri utenti. Le stanze devono avere un minimo numero di giocatori da attendere prima di iniziare la partita (minimo 4). Il creatore decide se l'ordine di gioco deve essere orario o anti-orario. Il creatore della stanza è il primo che invia il messaggio. Tale messaggio sarà tradotto nella lingua del secondo giocatore, il secondo giocatore aggiungerà un'altra parola a quella messaggiata, e la passerà al terzo giocatore. La frase verrà tradotta nella lingua del terzo giocatore, e così via finché non arriverà all'ultimo giocatore. Alla fine del turno, mostra a tutti i giocatori il percorso della frase.

Utenti che si vogliono inserire in una partita già iniziata, vengono messi in coda fin quando un altro utente non lascia la stanza. Gli utenti vengono inseriti nella stanza con una logica FIFO.

- 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
- Opzionale: le stanze di gioco possono essere create dagli utenti

## Progetto 3: La videoteca

Lo studente dovrà realizzare la simulazione di un sistema che modella una videoteca per un numero non precisato di utenti. Gli utenti devono potersi registrare e loggare al server. Il negoziante ha una lista di film disponibili nella videoteca, 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. I clienti possono connettersi al server per cercare, noleggiare o restituire film. Gli utenti possono prendere in prestito un numero K di film in prestito, dove K è deciso dal venditore.

Gli utenti mettono nel carrello tutti i film 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 negoziante dovrà contattare con un messaggio un cliente che non ha restituito il film dopo la scadenza prevista per la restituzione.

- Registrazione può essere sviluppata tramite database sql o file
- I film possono essere gestiti in un database, json, etc..
- Opzionale: La ricerca di un film può essere fatta per genere, nome oppure i più popolari

## Progetto 4: Dialogare con i robot

Lo studente dovrà sviluppare un sistema che permette di dialogare con i robot. Il client deve essere scritto in Python oppure Kotlin, il server deve essere scritto in C. Il server e i client dovranno comunicare attraverso le sockets. Il client avrà il compito di fare da intermediario tra il robot ed una persona; il client permetterà al robot di fare domande ad una persona (esempio, come ti chiami, cosa fai, etc..), e dovrà stabilire quale personalità ha l'interlocutore (livello di estroversione, amicalità, coscienziosità, stabilità emotiva, apertura a nuove esperienze).

In base alla personalità delle persone, il server deve decide cosa far dire al robot (client) e che tipo di comportamento adottare. Per esempio, se una persona è introversa, il robot mostrerà emozioni più riservate, tenderà a non guardare direttamente negli occhi la persona, parlerà in modo più calmo e usarà un linguaggio più cauto.

Come questionario della personalità può essere usato TIPI il https://gosling.psy.utexas.edu/scales-weve-developed/ten-item-personality-measure-tipi/ Esiste una versione in italiano e come calcolare le diverse personalità.

Il robot che si può usare è Furhat, la cui SDK è disponibile https://docs.furhat.io/getting_started/ Questo robot ha anche un emulatore.

- Opzionale: usare openai per la gestione dei dialoghi nel client.
- Opzionale: sviluppare diversi comportamenti per il robot. Per esempio, sviluppare anche comportamenti opposti a quelli della personalità percepita della persona. Esempio, se è estroverso, avere comportamenti introversi.