

## **Descrizione Sintetica**

Realizzare un sistema client-server che consenta a più utenti di prendere e portare oggetti da una locazione di partenza ad una destinazione indicata. Scopo del gioco è consegnare più oggetti alla destinazione.

Si utilizzi il linguaggio C su piattaforma UNIX. I processi dovranno comunicare tramite socket TCP. Corredare l'implementazione di adeguata documentazione.

## **Descrizione Dettagliata**

Il server manterrà una rappresentazione dell'ambiente in cui verranno posizionati degli oggetti, delle locazioni e degli ostacoli. L'ambiente sarà rappresentato da una matrice in cui gli utenti si potranno spostare di un passo alla volta nelle quattro direzioni: S, N, E, O oppure prendere o depositare oggetti (es. con azioni P, D). Il server posizionerà nella matrice locazioni, oggetti ed ostacoli in posizioni random. Ogni oggetto avrà associata una locazione in cui portarlo (indicata da un opportuno nome simbolico, es. L1, L2, etc.). Ogni utente, una volta connesso al server, verrà posizionato in una posizione random della matrice. All'inizio del gioco gli ostacoli sulla mappa saranno nascosta per l'utente, saranno invece visibili le posizioni degli altri utenti, degli oggetti e delle possibili locazioni. Il gioco durerà un tempo fissato a partire dal primo utente che inizierà a giocare. Gli utenti potranno inserirsi nel gioco anche a gioco già iniziato. Dopo ogni passo l'utente riceverà l'informazione sull'effetto proprio movimento: se lo spostamento porterà ad una collisione con un ostacolo oppure con un altro utente, il movimento avrà effetto nullo. In corrispondenza di un oggetto l'utente potrà prendere tale oggetto e leggere la locazione di destinazione. Dovrà quindi portarlo in tale locazione muovendosi sulla mappa per poi posarlo. Quando uno degli utenti avrà consegnato un numero massimo di pacchi, o alla scadenza di un limite di tempo fissato, il server notificherà agli utenti la fine della sessione e ne genererà una nuova.

Per accedere al servizio ogni utente dovrà prima registrarsi al sito indicando password e nickname.

Non c'è un limite a priori al numero di utenti che si possono collegare con il server.

Il client consentirà all'utente di collegarsi ad un server di comunicazione, indicando tramite riga di comando il nome o l'indirizzo IP di tale server e la porta da utilizzare. Una volta collegato ad un server l'utente potrà: registrarsi come nuovo utente o accedere al servizio come utente registrato. Il servizio permetterà all'utente di: spostarsi di una posizione, disconnettersi, vedere la lista degli utenti collegati, vedere il tempo mancante, vedere gli ostacoli incontrati e la posizione degli altri utenti.

Il server dovrà supportare tutte le funzionalità descritte nella sezione relativa al client. All'avvio del server, sarà possibile specificare tramite riga di comando la porta TCP sulla quale mettersi in ascolto.

Il server sarà di tipo concorrente, ovvero è in grado di servire più client simultaneamente.

Durante il suo regolare funzionamento, il server effettuerà logging delle attività principali in un file apposito. Ad esempio, memorizzando data e ora di connessione dei client, il loro nome simbolico (se disponibile, altrimenti l'indirizzo IP), data e ora della consegna dei pacchi insieme al nome dell'utente che ha consegnato.

## **Regole generali**

Il server ed il client vanno realizzati in linguaggio C su piattaforma UNIX/Linux. Le comunicazioni tra client e server si svolgono tramite socket TCP. Oltre alle system call UNIX, i programmi possono utilizzare solo la libreria standard del C. E' sconsigliato l'uso di primitive non coperte dal corso (ad es., code di messaggi) al posto di quelle studiate.

## **Relazione**

Il progetto va accompagnato da una relazione che contenga almeno le seguenti sezioni:

1. Una guida d'uso per il server e per il client, che illustri le modalità di compilazione e d'uso dei due programmi.
2. Una sezione che descriva le comunicazioni tra client e server nelle varie fasi del gioco.
3. Una sezione che descriva i dettagli implementativi giudicati più interessanti (con particolare riferimento alle system call oggetto del corso), eventualmente corredati dai corrispondenti frammenti di codice.
4. In appendice, la relazione deve riportare il codice sorgente integrale del progetto. Orientativamente, la relazione dovrebbe constare di almeno 10 pagine, esclusa l'appendice. Indicare sulla copertina della relazione i componenti del gruppo.

## **Consegna del progetto**

Entro la data prescelta per lo scritto finale (con eccezione per il primo appello), vanno consegnati al docente il progetto e la relazione. Il progetto e la relazione vanno inviati all'indirizzo [alberto.finzi@unina.it](mailto:alberto.finzi@unina.it) in un archivio compresso in formato zip o rar. La relazione va consegnata al docente anche in formato cartaceo al momento della discussione. Durante l'esame orale, il client ed il server verranno testati, eseguendoli su due o più macchine diverse.