

Progetto Difesa Aerea

Descrizione sintetica:

Realizzare un'applicazione client/server che simuli un sistema di difesa di uno spazio aereo.

Il sistema consiste di n postazioni antiaeree. Ogni postazione è caratterizzata da un identificativo, una posizione, espressa da una coppia di coordinate (x,y) su un piano, un raggio di azione r ed un numero iniziale di missili. Una stazione è attiva se possiede almeno un missile.

Un bersaglio B è caratterizzato da un identificativo e da una posizione (x,y) nel piano. Si assuma che sia il bersaglio che le postazioni siano fisse.

Quando un bersaglio compare nel campo di gioco, viene identificato da tutte le postazioni. Il bersaglio può essere neutralizzato dalla rete di difesa se rientra nel raggio di azione di almeno una stazione attiva. Quando una stazione decide di lanciare un missile, il bersaglio viene sempre neutralizzato e la stazione decrementa di uno il numero di missili in suo possesso.

Quando un bersaglio rientra nel raggio di azione di più stazioni attive, la stazione che neutralizza il bersaglio è quella che possiede il massimo numero di missili. Tutte le stazioni (attive o meno) visualizzano il destino del bersaglio ("bersaglio $\langle x \rangle$ neutralizzato dalla stazione $\langle y \rangle$ " o "bersaglio $\langle x \rangle$ non neutralizzabile da nessuna stazione attiva" oppure "bersaglio $\langle x \rangle$ non raggiungibile da nessuna stazione").

L'applicazione deve essere scritta utilizzando il linguaggio C. I processi comunicano attraverso socket TCP. L'elaborato deve essere corredato da una opportuna documentazione.

Descrizione dettagliata:

Processo server di rendezvous:

Il server di rendezvous processo fornisce il servizio di connessione tra le diverse stazioni del sistema. Questo processo riceve su riga di comando un numero di porta su cui crea un socket in ascolto.

Quando riceve una connessione da una stazione di controllo, invia:

- Alla stazione appena connessa l'indirizzo IP ed il numero di porta di tutte le stazioni presenti
- A tutte le stazioni presenti, l'IP ed il numero di porta della nuova stazione

Processo server:

Ogni stazione antiaerea è implementata da un processo server.

Il processo server riceve su riga di comando i seguenti parametri:

- L'identificativo della stazione
- IP e numero di porta del server di rendezvous.
- Posizione
- Numero iniziale di missili
- Raggio d'azione
- Numero di porta

Quando un server parte, riceve dal server di rendezvous l'elenco degli indirizzi (IP,num_porta) delle stazioni presenti.

La connessione di un bersaglio/client corrisponde all'identificazione del bersaglio da parte della stazione di controllo. Quando un server identifica un bersaglio/client informa tutti gli altri server della presenza del bersaglio, inviando l'identificativo del bersaglio e la sua posizione. I server a questo punto eseguono algoritmo per decidere se è possibile neutralizzare il bersaglio e, in caso

affermativo, quale stazione deve eseguire la neutralizzazione. Al termine dell'algoritmo ogni stazione visualizza il destino del bersaglio ed il proprio stato (numero di missili disponibili). Il server è implementato tramite un'applicazione multi-thread che deve supportare la connessione contemporanea di più client.

Processo client:

Il processo client riceve sulla linea di comando i seguenti parametri:

- L'identificativo del bersaglio
- l'indirizzo IP del server da contattare
- il numero di porta del server
- la posizione del bersaglio.

Il client si collega al server ed invia la propria posizione e riceve in risposta il suo 'destino', i.e., ("neutralizzato dalla stazione <y>" o "non neutralizzabile da nessuna stazione attiva" oppure "non raggiungibile da nessuna stazione").

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. Sarà valutato negativamente l'uso di primitive non coperte dal corso (ad es., code di messaggi) al posto di quelle studiate.

Relazione

Il progetto va corredato 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 illustri il protocollo al livello di applicazione utilizzato nelle comunicazioni tra client e server (non il protocollo TCP/IP!).
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 il numero assegnato al gruppo e i suoi componenti.

Consegna del progetto

Entro due giorni dalla data prescelta per lo scritto finale, vanno consegnati al docente il progetto e la relazione. Il progetto va inviato all'indirizzo clemente.galdi@unina.it in un archivio compresso in formato zip. La relazione va consegnata al docente anche in formato cartaceo. Durante la discussione del progetto, il client ed il server verranno testati, eseguendoli su due o più macchine diverse.