



UNIVERSITÀ DEGLI STUDI DI UDINE
CORSO DI LAUREA TRIENNALE IN INFORMATICA

Secondo progetto
per il corso di
Laboratorio di Sistemi Operativi

prof. Ivan Scagnetto – Nicola Gigante

A.A. 2016-2017 – Primo Semestre

Consegna

Il progetto qui assegnato consiste nel redigere un programma in linguaggio C che operi nel modo descritto nella prossima sezione.

- Il progetto, *facoltativo*, svolto *individualmente*, è da consegnare entro dieci giorni dalla data dell'appello di esame al quale lo studente si vuole presentare, all'indirizzo email `gigante.nicola@spes.uniud.it`.
- Il presente progetto viene valutato in aggiunta al primo, già descritto precedentemente.
- Ai progetti, se consegnati in tempo e di valutazione sufficiente, vengono assegnati un punteggio da un minimo di 1 ad un massimo di 4 punti ciascuno, di cui poi verrà calcolata la media aritmetica.
- Il risultato incrementerà il punteggio dell'esame scritto del *corso di laboratorio*. Con un solo progetto, l'incremento massimo è di 2 punti.

Testo del Progetto

Il progetto consiste nell'implementazione di una versione client/server del gioco chiamato Nim¹. Il gioco, che si svolge tra due giocatori, prevede *due* pile di pedine di lunghezza arbitraria. A turno ogni giocatore sceglie una delle due pile e un numero arbitrario di pedine da *rimuovere* dalla pila scelta. Alla fine, vince il giocatore che si trova a rimuovere l'ultima pedina.

Il progetto sarà diviso in un programma **nimserver** che si occupa di gestire la partita, e un programma **nimclient** che fornisce l'interfaccia di gioco:

- **nimserver** resta in ascolto di connessioni da parte dei client su un socket locale di dominio UNIX.
- Ogni singolo giocatore si collega al server utilizzando **nimclient**.
- Alla connessione del primo client, **nimserver** aspetta che se ne connetta un secondo. Dopodiché, fa cominciare la partita gestendo la connessione con i due client su un *thread* separato, tornando nel thread principale ad aspettare altre connessioni per, eventualmente, gestire altre partite.
- All'inizio della partita, **nimserver** sceglie *casualmente*² il numero di pedine presenti nelle due pile, e comunica i due valori ai due client, assieme all'informazione di quale dei due giocatori deve muovere per primo (scelto in modo arbitrario, ad es. il primo ad essersi connesso).
- Ad ogni turno, il server chiede di muovere al client a cui spetta il turno, informandolo sul conteggio attuale di pedine rimaste nelle due pile.
- Quando la mossa di uno dei due giocatori fa terminare le pedine in entrambe le pile, il server informa entrambi i client del fatto che la partita è terminata, informando di chi ha vinto. I client riportano questa informazione all'utente.

Entrambi i programmi devono essere il più *robusti* possibile: eventuali errori di comunicazione (ad esempio se al server non si collega il client giusto) non devono risultare in crash del programma o in comportamenti bizzarri. Bensì, **nimserver** deve gentilmente chiudere la comunicazione con un client che comunichi in modo non valido, continuando a gestire correttamente eventuali altre partite in corso. In ogni caso, se un client si disconnette per questo motivo o di sua spontanea volontà, il server non deve subire interruzioni e il client dell'avversario deve accorgersi della chiusura della partita informando l'utente dell'accaduto. Non deve essere possibile per un client assicurarsi la vittoria in modo fraudolento.

¹<http://it.wikipedia.org/wiki/Nim>

²Utilizzare le funzioni standard **srandom()** e **random()**, rispettivamente per fornire un seme all'inizio dell'esecuzione di **nimserver**, e per estrarre un numero casuale quando necessario.

Suggerimenti

Si suggerisce di pensare ad un protocollo di comunicazione *binario*, e definire una o più **struct**, in un file che sia in comune tra entrambi i programmi, che rappresentino il contenuto di un singolo messaggio scambiato tra client e server. Il contenuto di questa struct può poi essere letto e scritto in modalità binaria nei modi visti a lezione.

È opportuno cercare di condividere il più possibile il codice, soprattutto quello riguardante la comunicazione via socket, tra i programmi **nimclient** e **nimserver**, in modo che sia più facile mantenere coerenza tra cosa si aspettano i due programmi l'uno dall'altro.

Requisiti

Il codice deve essere consegnato in un archivio **.zip** o **.tar.gz** contenente:

- Il codice sorgente completo dei due programmi, compilante senza errori
- Un **Makefile** per la compilazione.
- Una relazione in formato **.pdf** che descriva brevemente il funzionamento del codice ed eventuali scelte che valga la pena giustificare. Si indichi su che sistema operativo e con quale compilatore è stato testato il programma.