

System call Project

Emma Maria Sole Tosato

24 January 2021

1 Informations

- Matricola: VR446676
- Nome e cognome: Emma Tosato
- Data di realizzazione: 24/01/2021
- Titolo esercizi : F4Server.c / F4Client.c / informazioni.h

2 Server

Il programma per il gioco forza4 Ã¨ costituito da un server e un client che viene eseguito due volte.

Il server viene eseguito da linea di comando, con lâaggiunta di righe, colonne e simboli.

Il main si occupa di eseguire un controllo sulle righe e colonne (che non devono essere minori di 5) e poi provvede a stampare i dati del gioco inseriti da riga di comando. Crea la prima memoria condivisa, inizializzando un puntatore che punta a questâultima, a cui attacca unca struttura dichiarata preventivamente.

La struttura, come alcuni altri dati, sono dichiarati nel file âinformazioni.hâ.

Crea anche una memoria condivisa a cui si attacca la matrice bidimensionale. Poi crea un vettore di semafori (sono 3): il semaforo 0 si occupa di bloccare/sbloccare il processo server, il semaforo 1 Ã¨ per il processo-giocatore1 e il 2 per il processo-giocatore2. Le prime semop servono per bloccare il server, in modo che venga sbloccato prima dal primo client e poi dal secondo. Entrambe le volte i dati portati dal client vengono inizializzati nella memoria condivisa. Infine il main entra in un while che si occupa di alternare i giocatori per le loro mosse, controllando ogni volta se ha vinto qualcuno. Il while continua ad essere eseguito fin tanto che la variabile che indica la fine della partita rimane vera per la condizione del ciclo. Dopo il ciclo, il main termina con lâeliminazione delle IPC.

Per quanto riguarda le funzioni, âvoid arbitraggioâ si occupa del controllo orizzontale e verticale. Le funzioni per la gestione del control-C invece sono tre.

Se termina il server, alla prima pressione il server stamper  un avviso, e, sei nei successivi 7 secondi lâutente non digita nuovamente control-c, il programma prosegue e la variabile contatore verr  azzerata, âdimenticandosiâ delle pressioni precedenti. Quando il server termina, manda un segnale ai client, elimina le IPC ed esce. Se   un client a terminare, il server ricever  uno specifico segnale da uno di questi due, comunicher  all altro client della conclusione della partita, ripristiner  la funzione del segnale ed uscir  .

3 Client

Il main del client, dopo aver definito la memoria condivisa, crea un contatore per vedere quante volte   stato eseguito il client. Poi memorizza il nome del giocatore inserito da linea di comando e infine controlla se un giocatore deve essere il computer. Salva anche i pid dei processi per le azioni successive.

Con la definizione dei semafori procede con lo sblocco del server, bloccando poi i due client con i rispettivi semafori.

Definisce la memoria condivisa per la matrice e poi inizia un while che terminer  , come nel server, con la conclusione della partita. Nel while vengono assegnati i simboli ai due client, poi viene stampata la matrice tramite una funzione e poi il client chiede il numero della colonna, facendo i dovuti di controlli. (Se il giocatore   il computer la colonna viene generata automaticamente). Poi viene stampata nuovamente la matrice, viene il sbloccato il server e bloccato il client che sta giocando. Concluso il while, il client esegue un controllo per la comunicazione dell esito della partita ad ogni giocatore.

Le funzioni per la gestione del control-c sono connesse al server: se termina un client, la funzione **segnale_c** controlla quale client sta terminando e manda un messaggio al server (che lo riceve come spiegato prima). Poi esce. La funzione **fine_client** stampa un messaggio per il client che ha vinto a tavolino (  il server che manda un segnale a questo client). La funzione **fine_server** invece comunica la terminazione di quest ultimo ad entrambi i client, dealloca ed elimina IPC varie e poi esce. La funzione stampa invece stampa le matrici.