Laboratorio di Sistemi Operativi 2017/2018 – Esercitazione Finale

DOCUMENTAZIONE PROGETTO

La simulazione della società di individui si articola nei seguenti eseguibili principali:

# GESTORE (gestore.c)

Avvia l'intero sistema leggendo il file di configurazione, facendo partire i processi e gestendo la distruzione dei essi al verificarsi dei seguenti casi:

* Due processi (A e B) si accoppiano e passano i loro dati al gestore che si occupa di eliminarli (*eliminaindividuo(pid)*)
* Passano *birth\_death* secondi e dunque viene eliminato l’individuo più vecchio(*uccidi\_individuo(individuo)*)
* Dopo *sim\_time* termina la simulazione e i processi attivi vengono eliminati

Il gestore crea processi solo all’inizio della simulazione e quando riceve le informazioni da parte di una coppia di processi (A e B) intenzionati ad “accoppiarsi”. In tal caso calcola genoma e assegna nome ai due figli creati in base alle indicazioni del testo.

Coda di messaggi e semafori vengono creati e inizializzati nel gestore.

L’eliminazione di individui viene fatta in mutua esclusione.

# PROCESSO A (processo\_A.c)

# Una volta creato un processo A, rimane in attesa di essere contattato da un processo B. Quando succede, questo processo A deve valutare che il genoma risultante dei figli generati con quel processo B sia più alto del suo stesso genoma; in tal caso accetta e comunica i dati al gestore nella coda di messaggi.

# Ogni volta che un processo A rifiuta la richiesta di un processo B, il suo target viene abbassato del 10% (*abbassa\_target()).*

# Solamente i processi di tipo A salvano le proprie informazioni nella memoria condivisa alla quale accedono i processi di tipo B per poter inviare le richieste.

# PROCESSO B (processo\_B.c)

# I processi B, una volta in esecuzione, accedono in mutua esclusione alla memoria condivisa dove possono leggere le informazioni dei processi A attivi e iniziare a inviare le richieste *(scegli\_A()*).

# Una volta selezionato il processo A da contattare, lancia una richiesta e se il processo A accetta (risp == 1) allora invia i propri dati al gestore altrimenti (risp == 2) cerca un altro processo da contattare (*contatta\_processo\_A(individuo \*arrA, int numA)*).

# 

I programmi precedenti si avvalgono del supporto dei seguenti file:

# sharedmem.c

# contiene le funzioni per la gestione della memoria condivisa.

# semaphore.c

# contiene le funzioni per la gestione dei semafori.

# library.h

# contiene le funzioni per la stampa dei dati di un individuo, il clalcolo del MCD e la struttura degli individui e dei messaggi.

Il programma potrà essere compilato in ogni sua parte mediante il comando **make** e avviato tramite il comando ./**gestore.**

Il file di testo configurazione **config.txt** è strutturato nel seguente modo:

