

Nome: MAGLIOLO Leonardo | Matricola: 844910

Corso: Sistemi operativi 2018/2019 4°Turno

Relazione di progetto

Struttura directory:

All'interno delle cartelle "datastructure" e "ipc" sono immessi gli headers in grado di implementare tutte le funzionalità riguardanti le strutture dati e istruzioni ipc da utilizzare all'interno dei programmi "gestore.c" e "studente.c".

All'interno della cartella "optconf_e_costanti" risiedono invece le costanti definite all'interno della consegna di progetto (comuni a entrambi i programmi c elencati precedentemente) assieme al file di configurazione "opt.conf".

Compilazione/esecuzione del progetto:

Per compilare programma "studente.c" è necessario eseguire il file "cmpstudente.sh" tramite bash da riga di comando. Se il compilatore non riuscisse a compilare il file studente e si eseguisse successivamente il programma gestore allora verrebbe eseguita l'ultima versione compilata di "studente.o".

Per compilare ed eseguire correttamente il programma "gestore.c" è sufficiente eseguire il file "cmpgestore.sh" come elencato precedentemente nel caso del programma studente. Ogni qualvolta si esegue lo script viene eliminata automaticamente la versione eseguibile precedente "gestore.o", in modo tale che l'esecuzione automatica non parta se la compilazione non ha avuto successo. Lo script in questione elimina automaticamente le istanze di memoria condivisa residue prima di eseguire il programma gestore. Se vengono cambiate le key ipc deve essere modificato anche "cmpgestore.sh".

Dopo una modifica di un qualsiasi parametro inserito all'interno dell'header "costanti.h" è necessario avviare dunque: "cmpstudente.sh", assicurarsi che non ci siano stati errori di compilazione e successivamente avviare "cmpgestore.sh".

Il progetto è stato compilato ed eseguito con successo su diverse macchine con compilatore: gcc (Debian 4.9.2-10) 4.9.1.

Liste e strutture dati:

listaStudenti.h

Lista di studenti: La lista di studenti è lunga POP_SIZE studenti.

Studente: La struttura contiene unsigned int rappresentanti:

pid, matricola, voto_AdE, nof_elems, grupOwner, voto.

listaGruppi.h

Lista dei gruppi: La lista dei gruppi contiene al più POP_SIZE istanze di gruppo.

Gruppo: La struttura gruppo contiene al più 4 pid e un flag intero unsigned il cui valore vale 0 se il gruppo è aperto oppure 1 se il gruppo è chiuso.

Il gruppo contiene un flag isUsed impostato a 1 se il gruppo esiste, a 0 se invece il gruppo all'index della lista non è stato ancora creato.

Al momento della sua creazione tranne casi specifici il gruppo deve risultare aperto.

conf.h

Conf: La struttura conf contiene quattro interi unsigned in grado di rappresentare le percentuali del tipo di gruppo.

La struttura contiene anche due interi unsigned in grado di rappresentare nof_invites e max_reject.

comunicazioni.h

Segnale: La struttura segnale contiene due interi unsigned per rappresentare rispettivamente: PID dell'interlocutore e segnale.

Comunicazioni: La struttura comunicazioni contenente: (i), (ii), e (iii) tramite liste di segnali.

(i)Lista di richieste aggregazione ricevute in pendenza: Struttura dati a nodi contenente la lista di richieste ricevute in pendenza a cui non si è risposto

(ii)Lista di richieste aggregazione inviate in pendenza: Definizione analoga alla precedente.

(iii)Lista di segnali da processare: Lista FIFO, una volta che i segnali vengono prelevati vengono eliminati dal buffer.

Strumenti/modalità di attuazione del progetto:

Fase uno - "Inizio simulazione": Viene adoperata la struttura dati condivisa "Lista di studenti" immettendo da parte del gestore i pid generati tramite fork all'interno dei campi "pid", viene anche inizializzata la struttura condivisa "conf" tramite lettura file della lista associata.

Per implementare l'attesa dei processi pronti viene utilizzato l'header "signals.h" per settare gli handler opportuni e immettere i segnali 'SIGREADY' e 'SIGSTART' all'interno della lista FIFO contenente i segnali da processare.

Fase due - "Formazione dei gruppi": Le comunicazioni in fase due tra gli studenti avvengono tramite scambio di segnali:

'SIGAGGREGAZIONE', 'SIGCONFERMA' e 'SIGRIFIUTO'.

La fase si interrompe qualora avvenga la ricezione del segnale 'SIGCONCLUSIONE' da parte del programma studente.

Fase tre – "Voto del progetto": Gli strumenti utilizzati in fase tre sono i medesimi utilizzati in fase uno (con l'aggiunta del segnale 'SIGVOTO').