

Relazione Progetto SOL

Danilo Crugliano
Matricola: 559346

Gennaio 2020

Indice

1	Supervisor	2
1.1	supervisor.c	2
1.1.1	Segnali	2
1.1.2	Pipe	2
1.1.3	Avvio server	2
1.1.4	Comunicazione	2
1.2	server.c	2
1.3	dispatcher.c	2
1.4	worker.c	3
2	Client	3
2.1	client.c	3
2.1.1	wait_	3
2.1.2	generate_id	3
3	Makefile	3
3.1	make all	3
3.2	make clean	3
3.3	make test	3
4	Test.sh	4
5	Misura.sh	4
6	"file".h	4
7	info.h	4

1 Supervisor

Di seguito vi sono riportati tutti i file contenenti il codice di implementazione del supervisor.

1.1 supervisor.c

1.1.1 Segnali

Per la gestione dei segnali sono stati creati due handler per i segnali -SIGINT e -SIGALRM. L'handler del SIGINT è costruito in modo che prenda in considerazione una variabile volatile sigatomic_c, se essa viene incrementata (arriva un cntrl-c) viene chiamata la funzione allarm(1), se passa un secondo la funzione lancia un -SIGALRM, nell'handler di -SIGALRM la variabile volatile sigatomic_c viene decrementata e viene stampato su stderr la tabella delle stime. Se nell'arco di quel secondo arriva un'altro SIGINT viene stampata su stdout la tabella delle stime e viene eseguita una kill su tutti i server.

1.1.2 Pipe

Per le pipe viene creata una matrice $2 \times k$, dove appunto ogni colonna è una pipe per la comunicazione tra il server e il supervisor. Le pipe sono anche state settate in maniera non bloccante per la read.

1.1.3 Avvio server

Viene inizializzato un array di dimensione k e per ogni elemento viene poi chiamata la funzione server (presente nell'omonimo file ".c") per la creazione dello stesso.

1.1.4 Comunicazione

La comunicazione è costituita da un ciclo infinito, si attende fino a quando non viene letto un messaggio con una stima da parte dei server e viene aggiornata la tabella delle stime calcolate fino a quel momento.

1.2 server.c

In questo file viene eseguita la fork in modo che il server sia un processo distinto, poi viene creata la socket e viene lanciato un thread per l'accettazione delle connessioni. Al thread come worker viene passata la funzione dispatche presente nell'omonimo file ".c".

1.3 dispatcher.c

In questo file vi è un attesa per accettare le connessioni dei client, in seguito viene avviato un thread che ha come parametri la funzione worker (presente

nell'omonimo file ".c") e come argomento di questa funzione viene passato un array contenente il file descriptor creato dall'accept e il numero del server.

1.4 worker.c

Questo file contiene l'implementazione della comunicazione tra server e client, ogni volta che il server legge un messaggio mandato dal client e (se sono arrivati almeno 2 messaggi) calcola la stima, come il minimo della differenza dei tempi di arrivo dei messaggi. Se alla fine del ciclo di comunicazione l'id del client è 0 e la stima è consistente viene inviato un messaggio al supervisor contenente l'id del client e la stima calcolata, tramite l'apposita pipe. Per ovviare a un possibile errore (perchè la write non è atomica), viene eseguita una lock prima della write e ovviamente una unlock dopo la write.

2 Client

2.1 client.c

2.1.1 wait_

Procedura usata per attendere secret millisecondi tra l'invio di un messaggio e un'altro.

2.1.2 generate_id

Funzione avente come valore di ritorno un id di 64 bit generato pseudo-casualmente.

3 Makefile

3.1 make all

Viene ripulita la directory da tutti i file creati in una ipotetica precedente compilazione, Viene compilato il supervisor con i relativi file necessari creando l'eseguibile "main". Viene compilato il client creando l'eseguibile "mainclient".

3.2 make clean

Può essere usata per ripulire la directory.

3.3 make test

Viene lanciato "make all" e in seguito ./test.sh per eseguire il ciclo di test.

4 Test.sh

Esegue il test lanciando il supervisor con 8 client, dopo 2 secondi lancia 20 client a coppie con attesa di 1 secondo e con parametri 5 8 20, in seguito attende 60 secondi e ogni 10 secondi lancia un segnale di -SIGINT, al termina lancia un doppio SIGINT per far terminare il supervisor e esegue lo script misura.sh con i risultati raccolti (i quali sono stati reiderizzati appositamente su determinati file ".log").

5 Misura.sh

Script che prende come parametri i file ".log" contenenti gli output di Supervisor e Client, li analizza e stampa le statistiche su quanti secret sono stati correttamente stimati e qual è l'errore medio di stima.

6 "file".h

File header in cui sono definite le varie struct le signature delle funzioni.

7 info.h

File in cui sono definite variabili a cui devono aver accesso più file, sono presenti anche delle Macro per la gestione di eventuali errori.