

Relazione progetto di Sistemi Operativi

Studente: Marina Gazzarrini

Matricola: 531162

Introduzione

Il progetto prevede la realizzazione di un sistema client–server per l'*out-of-band signaling*. La directory principale contiene tutti i file sorgenti, il makefile, lo script bash *misura e test*. Per costruire tutti gli eseguibili è sufficiente eseguire da terminale il comando *make*, per il test invece è necessario eseguire *make test*. Infine con *make clean* è possibile ripulire la directory di lavoro dai file generati.

Client

Il file client.c contiene l'implementazione del client. ID è stato dichiarato di tipo *long long int* così che sia almeno 64 bit (in alternativa poteva essere usato *int64_t*), mentre il *secret* è un semplice intero. Entrambi vengono generati casualmente combinando il valore delle funzioni *time()* e *getpid()* al fine di garantirne il più possibile l'univocità.

I *p* server con cui comunicare vengono scelti casualmente all'interno di un ciclo, successivamente il client si collega ai server selezionati e invia *w* messaggi contenenti il proprio ID in *network byte order*. Tra un messaggio e il successivo si simula un'attesa di *secret* millisecondi usando la system call *nanosleep*.

Infine si chiudono le connessioni.

Supervisor

Il supervisor per prima cosa deve lanciare *k* server come processi distinti e, per fare ciò, vengono utilizzate *fork()* ed *execl()* (Quest'ultima poteva essere evitata gestendo il processo figlio direttamente nel ramo *if*, ma è stato scelto di usarla per una maggiore chiarezza del codice). Inoltre mantiene con ciascun server una connessione attraverso pipe anonime e per gestire tali connessioni viene fatto uso di un selettore all'interno di un ciclo *while*. Il ciclo termina nel momento in cui vengono ricevuti due segnali *SIGINT* a distanza di meno di un secondo l'uno dall'altro. I segnali che vengono opportunamente

gestiti sono SIGINT e SIGALRM. Fondamentale è il flag `SA_RESTART` che forza la riesecuzione di `syscall` se interrotte da segnali (In alternativa si poteva gestire esplicitamente riattivando la `syscall` interrotta all'interno di un ciclo).

I dati ricevuti dai server vengono salvati, dopo opportuni controlli, all'interno di un albero binario di ricerca. Al momento della "chiusura" del supervisor viene inviato un segnale SIGTERM ai server.

Server

Il server (Processo figlio generato dal supervisor) riceve due argomenti: il *file descriptor* per la comunicazione con il supervisor e il proprio ID. All'interno di un ciclo, lancia, ad ogni nuova richiesta di connessione, un thread che si occuperà di gestirla.

La gestione delle connessioni è stata fatta senza utilizzare un selector poiché, in questo caso, il codice risulta più semplice e chiaro ed il selettore non è indispensabile (invece lo era nel caso del supervisor per non effettuare attesa attiva). Ogni thread riceve i messaggi di uno specifico client e infine invia la stima effettuata al supervisor attraverso la pipe anonima. Non è necessario l'utilizzo delle lock, anche se più thread devono inviare i dati sulla stessa pipe anonima, poiché la write è atomica se la dimensione del messaggio è minore della dimensione della pipe stessa; inoltre non c'è interesse nel dare precedenza ad un thread piuttosto che ad un altro. Infine, all'arrivo di un segnale SIGTERM si esce dal ciclo principale e vengono chiuse le connessioni.

Calcolo del secret

Per quanto riguarda il secret è stato scelto di prendere il minimo tra i valori stimati. Il metodo probabilmente più idoneo sarebbe stato quello di calcolare il MCD (massimo comun divisore) tra le varie stime, ma, dopo numerose prove e confronti, la scelta del minimo si è rivelata la più efficace. Infatti le condizioni richieste sui vari parametri (k , p , w) rendono alquanto improbabile, anche se non impossibile, che ad uno stesso server non arrivino due messaggi consecutivi da parte dello stesso client.

lib

Libreria che contiene funzioni utilizzate più volte all'interno del progetto, sia dal supervisor che dal server. Contiene la *define* per il controllo sistematico degli errori, la struttura dati *message* per i messaggi scambiati da server e supervisor e le funzioni per il calcolo del

minimo e del massimo comun divisore (Utilizzata solo in una iniziale implementazione, per poi decidere che fosse più opportuno usare in ogni caso il minimo).

abr

Rappresenta un albero binario di ricerca utilizzato per tenere traccia delle varie stime dei secret. Ogni volta che il supervisor riceve il secret stimato di un particolare client lo inserisce all'interno dell'albero, dopo opportuni controlli. Vengono salvati ID del client (utilizzato come chiave), la stima calcolata fino a quel momento e il numero di server che hanno contribuito alla stima. E' stato scelto di usare un albero binario di ricerca perché è sufficientemente efficiente (In alternativa poteva essere usata una tabella hash, ma sarebbe risultata più complessa a livello implementativo e non avrebbe apportato grandi vantaggi).

Makefile

Da la possibilità di generare gli eseguibili (*make*), eseguire i test (*make test*), ripulire la directory (*make clean*).

test.sh

Script in bash che esegue un ciclo completo di test. Per prima cosa lancia il supervisor con 8 server, dopo 2 secondi lancia 10 coppie di client ad 1 secondo di distanza l'una dall'altra. Infine invia un doppio SIGINT e lancia lo script *misura*. Si utilizza la redirectione dello *stdout* su file che verranno opportunamente passati allo script *misura* e poi eliminati.

misura.sh

Script in bash per effettuare le misurazioni sui dati raccolti. Viene fatto uso di due array associativi, contenenti uno i valori reali dei secret e l'altro quelli stimati.

Si effettua la lettura riga per riga del file che contiene l'output del supervisor e di quello che contiene l'output del client e i dati relativi ai secret vengono salvati negli array. Infine si effettua il confronto tra i valori contenuti nei due array e si stampano le statistiche associate.

Sistemi operativi su cui è stato testato il progetto:

Ubuntu 16.04 LTS

Ubuntu 18.04.2 LTS