

Relazione progetto Laboratorio di
Sistemi Operativi (2)
a.a. 2018-19
Out-Of-Band Signaling System

Tommaso Colella

09/10/2019

Indice

1	Scelte progettuali	1
1.1	Strutture dati d'appoggio	1
1.1.1	<code>linkedlist</code>	1
1.2	Moduli	1
2	Funzionamento generale	1
2.1	Interazioni tra moduli	1
2.1.1	Gestione della terminazione di supervisor e server . . .	2
2.2	Gestione dei segnali	2
2.3	Gestione della memoria	2
	Appendice A Istruzioni di esecuzione	3
	Appendice B Sistemi operativi usati per il testing	3
	Appendice C Test	3

1 Scelte progettuali

Di seguito vengono presentate le varie scelte progettuali effettuate durante la realizzazione del progetto.

1.1 Strutture dati d'appoggio

L'unica struttura dati utilizzata all'intero del progetto è `linkedlist`, una linked list implementata all'interno di `linkedlist.c`.

L'implementazione di `linkedlist` viene linkata all'eseguibile del supervisor come file oggetto.

1.1.1 `linkedlist`

La linked list è stata scelta principalmente per la semplicità e velocità di implementazione.

Nel corso dei test la componente è risultata stabile e sufficientemente affidabile in un contesto non critico.

1.2 Moduli

Il progetto è organizzato in moduli, tutti necessari al funzionamento dell'OOB-Signaling-System:

- `supervisor.c`
Deputato alla creazione dei server e alla preparazione di opportuni meccanismi di IPC per comunicare con essi (pipe anonime)
- `server.c`
Attende i messaggi dei client su una socket, inviando successivamente le stime al supervisor per mezzo delle pipe.
- `client.c`
Si occupa di generare il secret e di comunicare il proprio ID ai server.
- `linkedlist.c`
Implementazione della linked list di supporto.

2 Funzionamento generale

2.1 Interazioni tra moduli

Il supervisor, dopo aver creato i processi server, si mette in attesa delle stime ricevute da essi. Ogni stima ricevuta viene inserita nella linked list, in modo

da poter essere opportunamente recuperata al momento della ricezione di **SIGINT**.

I server ricevono messaggi (eventualmente ripetuti) contenenti ID esadecimali da parte dei client e si occupano di comunicare le loro stime sul secret al supervisor. I server terminano dopo aver ricevuto **SIGTERM** dal supervisor

I client comunicano coi server tramite socket inviandogli i propri ID. Il client termina dopo aver esaurito i messaggi.

2.1.1 Gestione della terminazione di supervisor e server

Il processo di gestione della terminazione ha inizio immediatamente dopo l'arrivo al supervisor di un doppio segnale **SIGINT**; in seguito alla ricezione di tale segnale il supervisor stampa le statistiche per poi inviare **SIGTERM** ai processi server.

2.2 Gestione dei segnali

La gestione di **SIGINT** avviene interamente all'interno del supervisor, il quale, dopo aver creato i processi server, registra un signal handler per **SIGINT**.

Il signal handler è dotato del flag **SA_RESTART**, in questo modo si evitano problemi che potrebbero nascere dalla interruzione della SC Select nel ciclo del supervisor.

2.3 Gestione della memoria

Per quanto concerne la gestione della memoria si è cercato di usare il più possibile lo stack e di usare le allocazioni sullo heap solo per dati di natura dinamica o di grandi dimensioni (in modo da scongiurare situazioni di stack overflow).

La funzione `snprintf()` è stata preferita a `sprintf()`, in tal modo si evita l'overflow dei buffer.

Appendice A Istruzioni di esecuzione

Scompattare il tar, eseguire `make && make test` e attendere la stampa delle statistiche finali.

Appendice B Sistemi operativi usati per il testing

- macOS 10.14 Mojave (piattaforma di sviluppo)
- Xubuntu 14.10 (macchina virtuale)
- Arch Linux
- Ubuntu 16.04 LTS

Appendice C Test

`make test` lancia il test standard della consegna, contenuto in `test.sh`. Al termine viene eseguito lo script di misura `misura.sh`, utilizzato per estrarre le statistiche sulle stime.

Il test aggiuntivo `test_n.sh` riceve come argomento un numero `n` e dopo aver eseguito gli `n` client lancia separatamente lo script `misura`. (N.B. `test_n.sh` richiede che siano già in esecuzione le istanze del server)