

Progetto di Laboratorio di Sistemi Operativi

Sessione autunnale a.a. 2013-14

Introduzione

Lo studente dovrà realizzare un sistema per inviare **segnali Morse**, in cui *segnali* è inteso come segnali POSIX. Il sistema consta di due applicazioni, un mittente e un destinatario; il mittente riceve un testo e lo invia tramite codice Morse al destinatario, che poi lo riconverte in testo. La trasmissione vera e propria è effettuata tramite l'invio di segnali POSIX. Scopo del progetto è determinare qual'è la massima velocità a cui si possono trasmettere dati con questo sistema su una particolare macchina; a questo scopo, una terza applicazione effettuerà test ripetuti di trasmissione con vari parametri, scoprendo a quale velocità cominciano a verificarsi troppi errori di trasmissione.

La codifica

La versione del codice Morse da utilizzare è quella raffigurata in tabella:

Lettere	Codice	Lettere	Codice	Numeri	Codice	Punteg.	Codice
A	• —	N	— •	0	— — — — —	•	• — • — • —
B	— • • •	O	— — —	1	• — — — —	,	— — • • — —
C	— • — •	P	• — — •	2	• • — — —	:	— — — • • •
D	— • •	Q	— — • —	3	• • • — —	?	• • — — • •
E	•	R	• — •	4	• • • • —	=	— • • • —
F	• • — •	S	• • •	5	• • • • •	-	— • • • • —
G	— — •	T	—	6	— • • • •	(— • — — •
H	• • • •	U	• • —	7	— — • • •)	— • — — • —
I	• •	V	• • • —	8	— — — • •	"	• — • • — •
J	• — — —	W	• — —	9	— — — — •	'	• — — — — •
K	— • —	X	— • • —			/	— • • — •
L	• — • •	Y	— • — —			Sottolineato	• • — — • —
M	— —	Z	— — • •			@	• — — • — •
						!	— • — • — —

Fonte: Wikipedia

Ogni punto corrisponde all'invio di un segnale SIGUSR1, mentre ogni linea corrisponde a un SIGUSR2. Fra un segnale e l'altro della stessa lettera deve trascorrere un tempo t ; fra l'ultimo segnale di una lettera e il primo della lettera successiva un tempo $3t$, fra l'ultimo segnale di una parola e il primo della parola successiva un tempo $6t$. È garantito che non verrà richiesto di trasmettere caratteri diversi da quelli elencati in tabella; non si fa differenza fra lettere minuscole e maiuscole.

Il mittente

Il mittente (nome eseguibile: **sendm**) riceve come argomento sulla riga di comando il PID del destinatario (p) e l'intervallo (t), espresso in millisecondi, che regola la velocità di invio dei segnali. Una volta avviato, dovrà leggere dal suo *stdin* un testo, sull'alfabeto di cui sopra, e inviare segnali al processo p con intervalli regolati da t come visto sopra. Il mittente termina quando il suo *stdin* viene chiuso, oppure in caso di errori non recuperabili nell'invio dei segnali.

Il destinatario

Il destinatario (nome eseguibile: **recvm**) riceve come argomento sulla riga di comando il solo parametro t , e si mette in attesa di segnali SIGUSR1 e SIGUSR2, li decodifica secondo le modalità viste, e invia l'output sul suo stdout. Il processo terminerà alla ricezione di un SIGINT.

Il controller

Il controller (nome eseguibile: **ctlm**) si occupa di testare il sistema e misurare il tasso di errori di trasmissione, al variare della lunghezza del messaggio e del tempo t . Riceve come argomento sulla riga di comando il valore iniziale di t , che indicheremo con t_0 , e il numero di test k da effettuare per ogni t . A questo scopo, dovrà coordinare una serie di esecuzioni di mittente e server, con valori di t che inizialmente saranno pari a t_0 , per poi effettuare una ricerca binaria del valore ottimale (ovvero, il minimo valore di t per cui non si verificano errori di trasmissione). Per ogni valore di t , il controller genererà casualmente k blocchi di testo di dimensione fra 16 e 8192 caratteri, con la condizione che il tempo stimato di trasmissione non superi 1 minuto a messaggio (ovvero: i blocchi di dimensione maggiore possono essere generati solo per valori sufficientemente piccoli di t – è sufficiente una stima approssimativa basata sulla durata media di un simbolo, non occorre una misurazione precisa basata sugli effettivi contenuti del blocco). Il controller dovrà quindi inviare il blocco di testo al mittente, e leggere il testo ricevuto dal destinatario, confrontandolo con il testo originale. Un test si intende superato se i due testi coincidono, fallito in caso contrario. Per ogni trasmissione, il controller dovrà stampare sul suo *stdout* una riga nel formato:

$$t \ dim \ r$$

In cui t è il valore dell'intervallo di trasmissione, dim la dimensione del blocco inviata, e r è OK se il test ha avuto successo, o KO se è fallito.

Misurazione

Si dovrà realizzare uno script bash (nome eseguibile: **statm**) che legga da *stdin* una serie di righe nel formato indicato sopra come output del controller, e stampi su stdout la percentuale di successo (ovvero: il rapporto fra il numero di test superati e il numero di test effettuati) per ogni valore di t , ordinato per t decrescente.

Makefile

Il progetto dovrà includere un makefile avente, fra gli altri, i target *all* (per generare tutti gli eseguibili), *clean* (per ripulire la directory di lavoro dai file generati), e *test*. Quest'ultimo deve eseguire un ciclo completo di test, lanciando il controller con parametri 500 e 10, in maniera tale che il suo output sia visibile a schermo, ed eseguendo anche lo script statm, di nuovo con output visibile a schermo.

Note finali

Nel realizzare il sistema oggetto del progetto, si tenga presente che una implementazione accurata sarà in grado di raggiungere velocità di trasmissione maggiori, e che il progetto si presta a un qualche riuso di codice fra vari componenti che dovrebbe essere ben realizzato.

Ci si attende che tutto il codice sviluppato sia, per quanto possibile, conforme POSIX. Eventuali eccezioni vanno documentate nella relazione di accompagnamento.

La consegna dovrà avvenire, entro il termine pubblicato, attraverso l'upload sul sito Moodle del corso di un archivio con nome *nome_cognome.tar.gz* contenente tutto il necessario. Scompattando l'archivio in una directory vuota, dovrà essere possibile eseguire i comandi *make* (con target di default) per costruire tutti gli eseguibili, e *make test* per eseguire il test e vederne i risultati. L'archivio dovrà anche contenere una breve relazione (3-5 pagine) in formato PDF in cui illustrate il vostro progetto.