

Sistemi Operativi ed in Tempo Reale

Assegnamento di Programmazione UNIX A.A. 2022

Istruzioni

1. Il presente assegnamento deve essere consegnato entro le ore 23.59 di giovedì 6 aprile 2022. Lo svolgimento dell'assegnamento, determinato dalla sua consegna, esclude dalla partecipazione della prova di programmazione in laboratorio.
2. L'assegnamento deve essere svolto individualmente dagli studenti che vogliono partecipare. Il testo dell'assegnamento riportato di seguito è in due versioni distinte denominate *a)* e *b)*. Gli studenti con numero di matricola pari devono fare riferimento alla versione *a)*, quelli con numero dispari alla versione *b)*.
3. L'elaborato deve essere contenuto in una directory denominata *sort-{numero di matricola}/*. Ad esempio, se il numero di matricola è 123456, la directory viene denominata *sort-123456/*. Una volta terminata la prova il candidato crei un file archivio avente il medesimo nome con estensione *.tar.gz*. Ad esempio usando l'istruzione:

```
tar czf sort-123456.tar.gz sort-123456
```
4. La consegna avviene tramite l'apposito form sul portale <https://elly2021.dia.unipr.it/> nella pagina di Sistemi Operativi ed in Tempo Reale. Solo in caso di difficoltà ben motivate lo si può consegnare inviando un'e-mail a dario.lodirizzini@unipr.it con oggetto "SORT: Assegnamento Unix di <Cognome Nome>", a cui è allegato il file archivio con l'elaborato.
5. Il candidato deve inserire in ogni file del codice sorgente (file *.h* e *.c*) della soluzione un commento contenente i propri nome, cognome e numero di matricola. Se lo ritiene opportuno, il candidato può inserire commenti all'interno del proprio codice o, qualora siano più generali, un file README.txt con ulteriori osservazioni.
6. Fatta salva la possibilità di chiedere precisazione al docente nel corso delle esercitazioni, lo svolgimento della prova è individuale ed è vietato lo scambio di codice tra i candidati.

Testo dell'Assegnamento

Si realizzi in ambiente Unix/C l'interazione tra processi descritta nel seguito. Il sistema consiste di quattro tipi di processi: un processo server generale S, un processo server hub H, processi sensori di temperatura T e processi attuatori A. Per la comunicazione tra il processo server e i processi cliente vengono utilizzate socket di tipo Stream. I processi menzionati gestiscono il riscaldamento di un edificio regolato dalle misure di temperatura. I processi sensori T raccolgono le misure e le inviano al processo hub H, che svolge l'intermediazione con il processo server. L'intermediazione è necessaria perché i sensori T sono dispositivi limitati e senza protezione dei dati. I processi hub H trasmettono i dati al processo server S, che a sua volta li distribuisce ai processi attuatori A che si sottoscrivono alla ricezione di messaggi.

I processi sensori T sono identificati da una stringa che si suppone univoca.

- a) Ad ogni invocazione vengono generati Nt processi figli (Nt sia un parametro specificato dall'utente), ciascuno dei quali si connette al processo hub H e, dopo avere atteso un intervallo temporale casuale compreso tra 0 e 6 secondi, gli invia una misura di temperatura (tipo floating point) generata casualmente tra -5°C e 35°C . Ogni processo figlio di T rimane in attesa di ricevere il numero di processi attuatori A, che hanno ricevuto e letto la misura inviata. Successivamente termina la propria esecuzione.
- b) Ad ogni invocazione di un processo T viene specificato il numero Nt di misure di temperatura che saranno generate e trasmesse. Iterativamente per Nt volte, il processo T si connette al processo hub H, gli invia una misura di temperatura (tipo floating point) generata casualmente tra -5°C e 35°C , attende di ricevere il numero di processi attuatori A, si disconnette ed attende un intervallo temporale casuale compreso tra 0 e 6 secondi. Dopo aver eseguito le precedenti operazioni Nt volte termina la sua esecuzione.

I processi attuatori A sono anch'essi identificati da una stringa che si suppone univoca. Essi possono essere invocati per svolgere due operazioni.

- *Iscrizione.* Ad ogni invocazione di A vengono specificati: la stringa identificativa dell'attuatore, la temperatura obiettivo $Tgoal$ (set point) e la lista dei sensori $T1, \dots, Tk$ (le stringhe che li identificano) alle cui misure si vuole sottoscrivere. Quindi, il processo A si connette al server ed invia il proprio nome e la lista $T1, \dots, Tk$ dai quali intende ricevere misure. Successivamente si mette in un ciclo di attesa della ricezione di una misura da uno dei sensori sottoscritti.
 - a) Ricevuta la misura il processo A calcola per ciascun sensore $T1, \dots, Tk$ la media delle misure di temperatura. Una media è valida se calcolata su almeno 3 misure. Se la maggioranza delle medie è inferiore a $Tgoal$, allora stampa a video il comando di accensione del riscaldamento. Se la maggioranza risulta inferiore, stampa invece il comando di spegnimento.
 - a) I dati ricevuti dall'attuatore sono valori medi della temperatura già calcolati dal server. Se la maggioranza delle misure di $T1, \dots, Tk$ è inferiore a $Tgoal$, allora stampa a video il comando di accensione del riscaldamento; altrimenti stampa a video un comando di spegnimento.

La ricezione termina quando il processo A riceve una misura associata ad una stringa sensore vuota.

- *Disiscrizione.* Il processo A si connette al server e comunica la stringa identificativa del processo attuatore che vuole disiscrivere dalla ricezione di misure. Successivamente termina la propria esecuzione.

Il processo hub H è un server concorrente che si occupa di inoltrare al server le misure ricevute dai processi sensori che ad esso si connettono. L'inoltro al server non avviene immediatamente dopo ogni connessione, ma il processo hub attende che ci siano J processi sensori T connessi, dove J è un parametro specificato al momento dell'invocazione di H.

Quando il numero di processi viene raggiunto, ciascun inoltro di una misura viene gestito tramite da un processo figlio di H. Ogni processo figlio si connette al server, invia la misura del processo sensore da servire, attende dal server la risposta e, quando l'ha ricevuta, la invia al processo sensore. Successivamente il processo figlio chiude la connessione con il server S e con il processo sensore T, che ha servito, e termina la sua esecuzione.

- a) L'ordine di invocazione dei processi figli per servire i diversi processi sensori T dipende dall'ordine di arrivo dei processi sensori, ossia secondo una politica FIFO (first in first out).

- b) L'ordine di invocazione dei processi figli per servire i diversi processi sensori T dipende dal numero di volte che sono stati serviti. Viene data priorità ai processi sensori T_i che sono stati serviti meno volte.

Il processo server S gestisce la ricezione e la distribuzione delle misure tra i processi clienti.

Ad ogni connessione di un processo attuatore A per l'iscrizione, il server riceve la lista dei sensori da cui l'attuatore desidera ricevere le misure. Completata l'iscrizione, la connessione con l'attuatore viene mantenuta attiva.

Ad ogni connessione di un processo attuatore A per la disiscrizione, il server riceve il nome (ossia stringa identificativa) del processo attuatore da disiscrivere. Successivamente termina la connessione con entrambi i processi (sia il processo attuatore da disiscrivere sia l'attuatore che ha comunicato la disiscrizione).

Ad ogni processo sensore T che si connette con l'intermediazione dell'hub, il server ne riceve il nome e la misura da esso inviata. Successivamente il server scorre la lista dei processi attuatori che si sono sottoscritti al sensore.

- a) Per ogni attuatore il processo server S crea un processo figlio che si occupa della trasmissione della misura all'attuatore in modo concorrente.
- b) Il processo server S aggiorna il valore medio delle misure fornite da quel preciso processo sensore T . Tale valore medio viene, quindi, trasmesso ai processi attuatori sottoscritti al server.

Infine, il processo server comunica al processo sensore il numero di processi attuatori che hanno ricevuto la notifica della misura.