

UNIVERSITA' di VERONA
DIPARTIMENTO di INFORMATICA
CORSO DI LAUREA in BIOINFORMATICA

14 Maggio 2017

Elaborato: 2 – System Call C

CONSEGNA:

Tramite E-LEARNIG entro non oltre il giorno 26 Giugno alle ore 02:00

COSA CONSEGNARE:

Archivio contenente il programma e tutti i sorgenti necessari debitamente commentati e indentati, una piccola relazione in formato PDF che **ILLUSTRI** il funzionamento delle parti salienti del programma. I sorgenti consegnati devono potersi compilare senza errori, pena la nullità del progetto, il progetto verrà **VALUTATO** anche se incompleto di alcune funzionalità richieste.

Verrà esposto apposito calendari per la difesa dei PROPRI elaborati.

Utilizzo delle system call Unix in C.

Il sistema dovrà gestire delle “Mailbox” tramite le code di messaggi.

La dimensione massima del messaggio è fissata in 256 caratteri. Si **assuma** che tale messaggio non contenga caratteri speciali.

Le singole mailbox sono individuate da un numero intero **MAGGIORE O UGUALE a 2**.

L'elaborato si compone di **due o più file .c**, per implementare una funzionalità Server e Client.

[server]

- Si deve occupare della creazione della coda di messaggi.
- Deve creare un figlio che si occupi di gestire i “messaggi” indirizzati al server ovvero che hanno **mtype=1**
- Se e solo se riceve il segnale **SIGUSR1** dovrà in modo autoritario, terminare il figlio, rimuovere la coda di messaggi e terminare a prescindere dal n° di messaggi ancora presenti in coda al sistema.
- Se e solo se riceve il segnale **SIGUSR2** dovrà verificare che NON vi siano messaggi in coda (suggerimento: si esegua una `msgctl` con comando `IPC_STAT` e si verifichi il campo `msg_qnum`) e in questo caso dovrà terminare il figlio e rimuovere la coda di messaggi.
- Deve intercettare il **CTRL+C** per evitare che venga terminato in modo inappropriato. Si può assumere una volta premuto **CTRL+C** venga inviato il segnale **SIGUSR1**.
- Il figlio dovrà leggere i messaggi `mtype=1` (a lui indirizzati) e interpretare i seguenti comandi:
 - **niceclose**
 - Se il testo del messaggio è `niceclose`, dovrà inviare al padre il segnale **SIGUSR2**.
 - **freeall**
 - Se se il testo del messaggio è `freeall`, dovrà rimuovere TUTTI i messaggi presenti nella coda a prescindere dal campo **mtype**.
 - **closetime <n>**
 - Richiede che il sistema di sistema si spenga dopo `n` secondi (a prescindere dal numero di messaggi presenti. Una volta passato il tempo, dovrà inviare al padre il segnale **SIGUSR1**.
- La sintassi di avvio del server sarà: `$: ./server`

[client]

- Saranno implementati 2 client, clientSend e clientRecive, rispettivamente per inviare un messaggio o riceverlo.

[clientSend]

- Se e solo se la struttura di coda di messaggi è operativa (in caso contrario dovrà segnalare il problema) prenderà come argomenti dalla riga di comando un numero rappresentante **mtype** (la mailbox) e il testo da inviare alla mailbox
- Si **assuma** di ricevere in ingresso un messaggio che non contenga caratteri speciali.

Es:

```
User$ ./clientSent 2 Ciao Mondo, stammi bene.
```

Manderà nella mailbox 2 come unico messaggio: Ciao Mondo

- **clientSend** è anche utilizzato (indicando come mailbox 1) per mandare i comandi al server:
 - **niceclose**
 - **freeall**
 - **closetime**

Es:

```
User$ ./clientSent 1 niceclose
```

```
User$ ./clientSent 1 closetime 100
```

L'ultimo comando farà in modo che il sistema si disattivi dopo 100 secondi a prescindere da quanti messaggi restano nel sistema.

[clientReceive]

- Se e solo se la struttura di coda di messaggi è operativa (in caso contrario dovrà segnalare il problema) prenderà come unico argomento dalla riga di comando un numero rappresentante **mtype** (la mailbox che dovrà essere maggiore o uguale a 2) e qualora sia presente un messaggio nella mailbox stamperà il messaggio a video.
- Qualora non siano presenti messaggi nella mailbox selezionata il programma resta in attesa dell'inserimento del prossimo messaggio.

Es:

```
User$ ./clientRecive 2
```

```
➔ Ciao Mondo, stammi bene.
```

Collegandosi alla mailbox 2 estrarrà un messaggio che verrà stampato a video: Ciao Mondo

COSA CI SI ASPETTA:

- Il programma deve compilarsi e girare senza errori...
- È titolo preferenziale per la valutazione (ma non obbligatorio) il controllo dei casi particolari, come ad esempio digitare **CTRL-Z** durante l'esecuzione del server o del client.
- **L'elaborato è personale, la presentazione di elaborati identici comporta la NULLITA' degli elaborati UGUALI.**
- In calce al file come commento deve essere riportata:

```
/*  
  Matricola  
  Nome e cognome  
  Data di realizzazione  
  Titolo esercizio  
*/
```