ISTRUZIONI PRATICHE

Esame del modulo di laboratorio di “Sistemi Operativi”

Durata: 150’ (2:30 ore)

● Creare una cartella principale denominata con il proprio numero di matricola e dentro esclusivamente un file main.c ed eventuali altri header files.

● Questa cartella andrà consegnata “zippandola” (compressione formato “zip”) in modo da creare un file avente per nome il proprio numero di matricola più l’estensione “.zip”. Deve essere compressa l’intera cartella e non solo il suo contenuto.

Se il proprio numero di matricola fosse 123456 questo deve essere anche il nome della cartella e l’archivio compresso da consegnare deve chiamarsi 123456.zip . All’estrazione dovrà essere presente la cartella 123456.

● Consegna:

○ Dopo 60’ ed entro 70’ dall’inizio della prova si deve fare una prima consegna (lavoro parziale) con il lavoro compiuto complessivo fino a tale momento ANCHE SE NON FUNZIONANTE utilizzando il modulo nell’homepage del browser. Il file consegnato dovrà chiamarsi **<matricola>_parziale.zip**

○ Dopo 120’ ed entro 150’ dall’inizio della prova si deve fare una seconda consegna (lavoro finale) utilizzando il modulo nell’homepage del browser: questa consegna è l’unica considerata per la valutazione finale .

○ I punteggi [x] sono indicativi dato che la valutazione tiene conto anche di dettagli “trasversali” che non sono riferibili a singoli punti. Un punteggio complessivo maggiore o uguale a 31 porterà alla lode.

NOTA: parte delle verifiche può avvenire con procedure completamente o parzialmente automatizzate per cui le denominazioni e gli output devono essere rigorosamente aderenti alle indicazioni. Eventuali irregolarità comportano l’esclusione dalla prova oltre a possibili sanzioni disciplinari.

Il codice va opportunamento commentato.

Programma in C

Deve essere creato un programma in C che gestisca una caccia alla balena (digitale).

Arruolamento

1. [2] Il processo principale, "*nave*", gestirà vari marinai e vari eventi. Il programma deve essere lanciato con due parametri: il primo parametro `<timoniere PID>`, che corrisponde ad un PID di un processo esistente, ed il secondo parametro un intero `<n>`. Un numero diverso di parametri deve far terminare il programma con **codice 90**.

2. [3] La nave deve generare un processo *vedetta*, il quale sarà responsabile dell'avvistamento delle varie balene. Una volta creata, la vedetta deve, prima di tutto, inviare un segnale `SIGUSR1` al processo esterno timoniere `<timoniere PID>`.

3. [4] La nave deve generare `<n>` processi arpionieri, ognuno dei quali dovrà arruolarsi **creando un file** nella cartella **`/tmp/registrazione/`** denominato con il PID del creatore e permessi di lettura e scrittura. Una volta creato, gli arpionieri dovranno rimanere attivi per svolgere le restanti attività !

   Es: l'arpioniere con PID 40 dovrà creare un file vuoto ***`/tmp/registrazione/40`*** *mentre l'arpioniere con PID 41 dovrà creare il file* ***`/tmp/registrazione/41`****. La cartella* ***`/tmp/registrazione/`*** *sarà presente in fase di valutazione.*

4. [2] Tutti gli arpionieri fanno parte della stessa squadra ed esiste sempre un capo arpioniere. Uno degli arpionieri deve diventare process group leader e tutti gli altri arpionieri devono unirsi al suo gruppo.

Inizia la caccia

5. [3] Il ruolo della vedetta è avvistare le balene, le quali amano sfiatare con `SIGUSR2`! Quando la vedetta riceverà dei segnali `SIGUSR2`, allora avrà avvistato una balena e dovrà quindi gridare su stdout "`Balena!\n`".

6. [4] La vedetta dovrà capire il tipo della balena e la sua distanza dalla nave. Per far ciò, dovrà leggere i messaggi sulla coda identificata dal file `/tmp/mare` e `<timoniere PID>`. Ogni messaggio conterrà il tipo e la

distanza di ogni balena: la distanza rappresentata dal tipo del messaggio, la tipologia della balena scritta invece in un payload di 50 caratteri.

Es: all'avvistamento di una balena, la vedetta potrà leggere sulla coda un messaggio contenente come payload "`beluga`" di tipo `10`.

NB: sia la coda che il file `/tmp/mare` saranno presenti in fase di valutazione.

7. [6] Una volta avvistata una balena, la vedetta deve comunicare a tutti gli arpionieri la sua tipologia e la sua distanza. Come segnalare ciò, sta a voi! Ricevute queste informazioni, ogni arpioniere dovrà creare una fifo **`/tmp/<pid>-<distanza>`** con **`<pid>`** uguale al suo PID e **`<distanza>`** uguale alla distanza della balena. Una volta creata la propria fifo (con i permessi adeguati), ogni arpioniere dovrà scrivere su di essa la tipologia della balena.

   *Es: quando viene avvistato un beluga a distanza 10, la vedetta lo comunicherà ad ogni arpioniere. L'arpioniere con PID 23 dovrà quindi creare una fifo **`/tmp/23-10`** e scriverci dentro "`beluga`".* L'arpioniere con pid 24 dovrà creare una fifo **`/tmp/24-10`** e scriverci dentro *"`beluga`". Quando viene avvistata una nuova balena, dovranno fare lo stesso con una nuova fifo.*
   NB: le distanze saranno sempre diverse ed in fase di valutazione verranno aperte le fifo in lettura per leggere il messaggio. Attenzione a rendere disponibile il messaggio fino alla sua lettura!

Moby Dick!

8. [3] Tra tutte le balene, la più temibile è certamente la cerea Moby Dick! Solo una persona non la teme: il capitano Ahab. Quando viene avvistata Moby Dick, si deve chiedere aiuto al Capitano Ahab, lanciando un binario "`./ahab.out`" e passandogli come parametro il `<timonierePID>` (**NB**: i parametri sono sempre e solo stringhe!). Come lanciare il binario è a vostra discrezione, purché non si interrompano le altre funzionalità del programma.
   NB: il binario `ahab.out` sarà collocato nella cartella del vostro eseguibile in fase di valutazione.

9. [4] Moby Dick attaccherà alcuni arpionieri uccidendoli (in fase di valutazione verranno fatti terminare i relativi processi). Una volta che moby dick ha terminato l'attacco, invierà un segnale `SIGRTMAX` al processo nave. Ricevuto questo segnale, la nave dovrà quindi capire quanti arpionieri sono stati uccisi, ed inviare al processo `<timonierePID>` un segnale `SIGRTMIN` con un payload intero contenente il numero totale di arpionieri uccisi.