

# Prova Pratica 064 - online

La prova consiste di due esercizi, il primo è un esercizio di programmazione concorrente, il secondo è un esercizio bash.

Avrete a disposizione circa 50 minuti per risolvere i due esercizi e consegnarle la soluzione richiesta.

Potete usare il materiale contenuto in <http://www.cs.unibo.it/~ghini/didattica/TREE4OS2021.tgz> che dovete già avere scaricato prima di questo esame

Non potete navigare in internet cercando informazioni all'esterno.  
Ovviamente, non potete comunicare con nessuno, in nessun modo.

Salvate i files spesso, per premunirsi in caso di crash.

# Esercizio Esame Pratica Online 173- keppall2 (1/2)

Gli alieni della specie Keppall2 conducono una vita monotona, ciascuno di loro nasce, vive per un periodo di tempo casuale tra 1 e 4 secondi e poi muore. Però i keppall possono morire solo uno alla volta e secondo la seguente procedura: ogni volta che un keppall muore, subito DOPO (E SOTTOLINEO **DOPO**) l'avvenuta morte del keppall, il main crea un numero casuale di altri keppall, tra 0 e 2, i quali cominciano così la loro esaltante esistenza. Solo dopo che questi thread sono stati creati, allora un altro keppall potrà morire.

All'inizio ci sono **5 Keppall**. Implementare il sistema descritto, utilizzando dei thread POSIX per ciascuna figura (**keppal e main**).

Utilizzare come base dell'esercizio i seguenti files, aggiungendo il codice necessario:

**keppal2.c**    DBGpthread.c    DBGpthread.c    prnterror.h

Scrivere il Makefile per compilare e linkare i sorgenti. Inserire il necessario controllo di errore. In caso di errore grave, terminare il programma producendo un avviso a video.

Non è obbligatorio utilizzare, nel codice che aggiungerete voi, le funzioni DBG\* contenute nei files DBGpthread.\* ma è fortemente consigliato. Per implementare le attese di un tempo variabile, usare la funzione **attendi(int min, int max)** , implementata nel file keppal2.c, che genera un numero casuale X tra min e max ed attende X secondi prima di restituire il controllo al chiamante.

Suggerimento: quando un keppall vuole morire deve farlo sapere al main affinché questo poi crei altri keppall. Per avvisare il main, il thread morituro, **quando diventa possibile**, deve: mettere l'identificatore del proprio pthread in una variabile globale **th\_morto**, settare a 1 la variabile **morto**, avvisare il main che sta per morire e poi morire. Il main trova l'identificatore del thread nella variabile th\_morto e attende e verifica che il thread sia morto veramente e poi crea altri thread.

# Esercizio Esame Pratica - 174 - generaric.sh

Scrivere un file di testo generaric.sh che prende 3 argomenti:

il primo argomento è un carattere, ad esempio il carattere `*`

il secondo argomento è un carattere, ad esempio il carattere `;`;

il terzo è un numero intero N maggiore o uguale a 0, ad esempio 3.

Lo script deve utilizzare **un approccio ricorsivo** per mettere in output una stringa formata dalla concatenazione di  $2*N$  caratteri come il primo argomento, seguito da N caratteri come il secondo argomento. L'esempio riportato deve mettere in output la stringa `*****;;;`

Attenzione che, come nell'esempio indicato, i caratteri passati come argomento potrebbero essere dei caratteri speciali.