

Concurso3

Signals-Threads-Mutex

A partir de quinta-feira 17/04 poderá entregar o relatório (num único ficheiro formato pdf) do concurso 3 através de link reservado para o efeito.

A entrega ficará aberta até ao dia 21/04, às 23h59. Nenhuma extensão do prazo será possível.

Ex. 1.

- a) Escreva um programa que não termine até o quinto Ctrl-C.
- b) Escreva um programa que permita que 5 threads escrevam uma mensagem completa (por exemplo "Mensagem da Thread 2") sem ser perturbada por outras threads. A mensagem será exibida letra por letra com um atraso de 1 segundo entre cada exibição de letra.

Ex. 2 :

Escreva um programa que usa um thread que lê caracteres no teclado e outro thread que é responsável por exibi-los. O main thread é o responsável por criar as threads. A paragem de execução é programado quando o caractere "E" é introduzido.

Example:

Thread 1: ghjhglo gh 12

Thread 2: g

Thread 2: h

Thread 2: j

Thread 2: h

Thread 2: g

....

Thread 2: 2

Thread 1: E

Thread 2: E

Prog finished

Ex. 3.

// #####prog.c#####

| | |
|--|--|
| <pre>#include <pthread.h> #include <stdio.h> #include <stdlib.h> #include <string.h> #include <unistd.h> pthread_t tid[2]; int counter; void* trythis(void* arg) { unsigned long i = 0; counter += 1; printf("\n Job %d has started\n", counter); for (i = 0; i < (0xFFFFFFFF); i++) ; printf("\n Job %d has finished\n", counter); return NULL; }</pre> | <pre>int main(void) { int i = 0; int error; while (i < 2) { error = pthread_create(&(tid[i]), NULL, &trythis, NULL); if (error != 0) printf("\nThread can't be created :[%s]", strerror(error)); i++; } pthread_join(tid[0], NULL); pthread_join(tid[1], NULL); return 0; }</pre> |
|--|--|

- a) Compilar e executar o programa prog.c
- b) Explique o output.
- c) Propor um código alternativo para um correto output.