System and device programming

Iniziato	mercoledì, 1 giugno 2022, 00:06
Stato	Completato
Terminato	mercoledì, 1 giugno 2022, 00:06
Tempo impiegato	8 secondi
Punteggio	0,00/14,00
Valutazione	**0,00** su un massimo di 15,50 (**0**%)

Domanda 1

Risposta non data

Punteggio max.: 3,00

Explain what do we mean for asynchronous I/O, and describe the Windows's strategies to perform it.
Report a piece of C code to **motivate** its use over synchronous I/O.

If you do not remember the exact syntax of Windows API system calls, write down a mock version of these system
calls with some C comments briefly summarizing their meaning together with what you were trying to do.

Domanda 2

Risposta non data

Punteggio max.: 2,00

Write a small C++ class to support the dynamic memory allocation in a multi-thread protected way.

In particular, suggest two C++ methods:

1. void * allocate(const size_t & num_bytes, const char & init_value); // allocate num_bytes and return the po inter to the first byte.
2. void free(void * ptr); // free the memory pointed by ptr and make it ready to be allocated again.

The class is initialized with a certain amount of memory (in the constructor, size in bytes) and the placement of the allocation is due to the class code (hence no new, delete, malloc and free are necessary).

The class can include any data necessary to support both the basic requirements and the concurrency. Make the second parameter of the allocation method optional. **All issues must be thrown as exceptions.**

Note: if you do not remember the exact syntax of a C++ class, write down a version that likely resembles what you remember together some C++ comment, briefly summarizing what you were willing to use.

Domanda 3

Risposta non data

Punteggio max.: 3,00

Write a small piece of code showing how to generate asynchronous tasks in a chain mode, so that they are going to be executed as soon as the output of the previous task is ready.

Make the first task not to wait any input.

Note: if you do not remember the exact syntax of a C++ class, write down a version that likely resembles what you remember together some C++ comment, briefly summarizing what you were willing to use.

Domanda 4

Risposta non data

Punteggio max.: 2,00

Write a UNIX program able to simulate the behavior available in the Windows system through a thread pools.

More specifically:

- The main thread creates a queue **Q** and **N** working threads. Then, it inserts a task into the queue **Q** at random intervals. Suppose that the main thread can call the function **generate_task** to generate a task before inserting it into the queue.
- The queue **Q** is able to contain at most **M** tasks of type **task_t.**
- Each working thread reads a new task from the queue at random time interval and it executes suck a task. Suppose that each working thread can call the function **consume_task** to execute a task extracted from the queue.

Ensure the proper synchronization among the unique main thread and the N working threads and explicitly declare which synchronization scheme you adopt. The primitives required to manage the queue do not have to be implemented (they belong to an external but usable library).
Suppose the main thread will insert **N** tasks of type "stop" to end all working threads. Use function **is_task_stop** to detect these stopping (and fake) tasks.

```
task_t *generate_task ();   // It returns a task to insert into the queue
void consume_task (task_t);   // it consumes a task extracted from the queue
int is_task_stop (task_t);   // It returns 1 if the task is a fake/sopping one, 0 otherwise
```

Domanda 5

Risposta non data

Punteggio max.: 2,00

A program runs 3*N threads of type A, 2*N threads of type B, and N threads of type C.
N is a positive integer passed on the command line (e.g., 10) and all threads are identified by their category (i.e., A, B or C) and a creation index (i.e., 1, 2, ...).
Coordinate the effort of all threads such that for each set of 6 threads running consecutively, 3 are of type A, 2 of type B, and 1 of type C in any order but such that the thread of type C is always the last one for each group.
Write the code using C in the Windows environment.

The following is a correct example of execution with N=10:

```
A3 B1 A2 A4 B7 C6
A7 A8 B3 B5 A1 C8
B4 A9 A6 B2 A5 C1
etc.
```

If you do not remember the exact syntax of Windows API system calls, write down a mock version of these system calls with some C comments briefly summarizing their meaning together with what you were trying to do.

Domanda 6

Risposta non data

Punteggio max.: 2,00

Two processes use a pipe to synchronize and a section of memory to share and exchange some data between them. More specifically:

- Process P1 reads from standard input a sequence of strings and it writes these strings onto the shared memory. It stops this process when the string "STOP" is encountered. Before stopping, it wakes-up process P2 using a pipe.
- When process P2 starts running, it reads the strings from the shared memory and it writes them on standard output transforming each capital letters into a small letter and each small letter into a capital letter. When it has done it wakes-up process P1 using a pipe.

Organize the entire program in which P1 and P2 are supposed to run forever.
Please remind the following system calls.

```
key_t ftok (const char *path, int id);
int shmget (key_t key, size_t size, int flag);
void *shmat (int shmid, const void *addr, int flag);
int shmdt (const void *addr);
```