



Politecnico
di Torino

System and device programming

Iniziato	mercoledì, 1 giugno 2022, 00:09
Stato	Completato
Terminato	mercoledì, 1 giugno 2022, 00:09
Tempo impiegato	8 secondi
Valutazione	0,00 su un massimo di 14,00 (0%)

Domanda 1

Risposta non data

Punteggio max.:

3,00

A program must manipulate an extremely large file (e.g., several GBytes), thus it decides to do that using several threads each one manipulating a section of the file over and over again until the file has been completely manipulated.

Following this scheme, write a program able to run N threads (where N is an integer value, e.g., 8, 12, 16) to manipulate the file. Each one of the threads will manipulate 1GBytes of the file, **mapping** the correct section of it into its local memory. For the sake of simplicity, file manipulation consists in reading the file byte-by-byte and rewriting each byte by increasing its value of a constant value C (e.g., $C=5$) module 256.

For example, if the file is large 10 GBytes and there are 4 threads, thread number 3 may manipulate the first GByte of the file, thread number 2 the second GByte, thread number 4 the third GByte, etc., until the entire file has been read and written and all threads terminate.

Please remind the following system calls.

```
LPVOID MapViewOfFile (HANDLE hMapObject,DWORD dwAccess,DWORD dwOffsetHigh,DWORD dwOffsetLow,SIZE_T dwNumberOfByteToMap);  
BOOL UnmapViewOfFile (LPVOID lpBaseAddress);
```

If you do not remember the exact syntax of other Windows API system calls, write down a version that likely resembles what you remember together some a C comment, briefly summarizing what you were willing to use.

Domanda 2

Risposta non data

Punteggio max.:

3,00

Write a small C++ program to accumulate a matrix of integers by rows (so that the final results will be a vector where, at each index, is stored the sum of all elements on the columns for the indexed row).

The main program must print the vector only when ready (you can omit the print but not any synchronization primitive).

Make the task able to start the computation only when its row is fulfilled by the user (after their creation).

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.:

2,00

An ASCII file, stored in the UNIX system, includes an undefined number of records. Each record includes 4 fields: an integer value, two strings, and a real number. All 4 fields have variable size and are separated by an unknown number of spaces. The two strings have a maximum size equal to 100 characters. The following is a possible correct format of such a file:

```
123 string1 STRING2 45.67
68799 ABC VWXYZ -12345.879
...
```

Show how it is possible to use the UNIX system call **read** to store the entire file in an array of structures of type **record_t**. The number of records stored in the file is indicated on the first line of the file. The array of structures must be dynamically allocated by the function.

The structure **record_t** is defined as follow:

```
#define N 101

struct record_s {
    int i;
    char s1[N], s2[N];
    float f;
} record_t;
```

Domanda 4

Risposta non data

Punteggio max.:

2,00

A Windows program initializes a semaphore named **sem** to N, and then it allow **at most N** thread entering the critical section **CS** using in all threads the following prologue and epilogue:

```
WaitForSingleObject (sem, INFINITE);  
CS  
ReleaseSemaphore (sem, 1, &pc);
```

Re-implement the same prologue and epilogue using only mutexes as synchronization strategies.

Please, remind and use only the following synchronization system calls.

```
HANDLE CreateMutex(LPSECURITY_ATTRIBUTES lpsa, BOOL fInitialOwner, LPCTSTR lpszMutexName);  
BOOL ReleaseMutex (HANDLE hMutex);
```

Domanda 5

Risposta non data

Punteggio max.:

2,00

Write a C++ multi-thread piece of code that is able to:

1. Run N threads, each one creating a random number and storing it into a stack, until one of them generate the value 0. When the value 0 is generated all threads stop.
2. Run 2 threads, one for even and one for odd numbers. They all pick one element from the stack and, if the value is of their assigned type (even or odd), they save it into a shared vector.
3. Run 1 thread that, when the stack is empty and the vector full, prints the entire vector.

Let the synchronization be coherent with the task each thread has to carry.

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 6

Risposta non data

Punteggio max.:

2,00

First, illustrate the use of condition variables in the UNIX system.

Then, clarify how they can be used in a Producer and Consumer scheme (with P producers and C consumers, where P and C are generally larger than 1).

Please, remind the following system calls.

```
int pthread_cond_init (pthread_cond_t *restrict cond, const pthread_condattr_t *restrict attr);  
int pthread_cond_wait (pthread_cond_t *restrict cond, pthread_mutex_t *restrict mutex);  
int pthread_cond_signal (pthread_cond_t *cond); int pthread_cond_broadcast (pthread_cond_t *cond);  
int pthread_cond_destroy (pthread_cond_t *cond);
```