System and device programming

Iniziato	mercoledì, 1 giugno 2022, 00:12
Stato	Completato
Terminato	mercoledì, 1 giugno 2022, 00:12
Tempo impiegato	9 secondi
Valutazione	**0,00** su un massimo di 14,00 (**0**%)

Domanda 1

Risposta non data

Punteggio max.: 2,00

A UNICODE file stores a sequence of records, in which each record includes 3 fields: an integer value, a string, and a real number. The 3 fields have all variable size and are separated by a variable number of spaces. The following is a possible correct format of such a file:

```
123 ThisIsAString 45.67
687989 ABC -12345.879
. . .
```

Show how it is possible to use the system call

```
BOOL ReadFile (
  HANDLE hFile, LPVOID lpBuffer, DWORD nNumberOfBytesToRead,
  LPDWORD lpNumberOfBytesRead, LPOVERLAPPED lpOverlapped
);
```

To store the indicated records in an array of structures of type **record_t**, defined as follow:

```
#define N 1000

struct record_s {
  DWORD i;
  TCHAR s[N];
  FLOAT f;
} record_t;
```

Domanda 2

Risposta non data

Punteggio max.: 2,00

Using the following data structure and system calls:

```
struct dirent {
  inot_t d_no;
  char d_name[NAM_MAX+1];
  . . .
};
DIR *opendir (const char *filename);
struct dirent *readdir (DIR *dp);
int closedir (DIR *dp);
```

write the function

```
void visit (char *pathname, char *prefix, int level);
```

which displays the content of all directories within the directory called **pathname** whose name start with the string **prefix** and have a level (within the directory tree) equal to **level**.
For example, if **pathname is "mydir"**, **prefix is "foo"** and **l=2**, the function must display the content of all subdirectories of directory **"mydir"** such as **"foo1"**, **"foobar", "fooABC"**, etc., at level 2 of the directory tree rooted at **"mydir"**.

Domanda 3

Risposta non data

Punteggio max.: 4,00

A program runs N threads of type A, 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, ..., N).
Coordinate the effort of all threads such that for each set of 3 threads running consecutively, there are only two possible sequences: A B C or C B A.
The following is a correct example of execution with N=10:

```
A3 B1 C6
A2 B4 C1
C8 B7 A8
etc.
```

Write the code using the C language but, or the sake fo simplicity, suppose semaphores can be managed by pseudo-code functions such as: i**nit(sem,k)**, **wait(sem)**, **signal (sem)**, and **destroy (sem)**.

Domanda 4

Risposta non data

Punteggio max.: 2,00

Verify the following C++ multi-thread code and propose a solution that avoids deadlocks without introducing or removing any mutex:

```
std::mutex read_m, write_m, verify_m;
void writer() {
    read_m.lock(); write_m.lock(); verify_m.lock();
    // body
    verify_m.lock(); write_m.lock(); read_m.lock();
 }

void reader() {
   write_m.lock(); read_m.lock();
    // body
    read_m.unlock(); write_m.unlock();
}

void verifier() {
    verify_m.lock(); write_m.lock();
    // body
    write_m.unlock(); verify_m.unlock();
}
```

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

Domanda **5**

Risposta non data

Punteggio max.: 2,00

Write two small segments of C code to explain how it is possible to use the two following system calls

```
DWORD WaitForSingleObject (HANDLE hObject, DWORD dwTimeOut);
DWORD WaitForMultipleObjects (DWORD nCount, LPHANDLE lpHandles, BOOL fWaitAll, DWORD dwTimeOut);
```

to wait for N threads (with N>64).

Why "N>64"?

If the epilogue of a critical section includes (only) the following system call

```
ReleaseSemaphore (hSem, 3, &previousCount);
```

How would you write the corresponding prologue? Motivate the strategy adopted.

Domanda 6

Risposta non data

Punteggio max.: 2,00

Write a small C++ example able to start N threads and let them all waiting for their input value to start the internal computation. The input value (for each thread) comes from the user input, and it must happen after the thread creation.

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