System and device programming

Iniziato	mercoledì, 1 giugno 2022, 00:10
Stato	Completato
Terminato	mercoledì, 1 giugno 2022, 00:10
Tempo impiegato	9 secondi
Valutazione	**0,00** su un massimo di 14,00 (**0**%)

Domanda 1

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 generating a random char and storing it into a vector.
2. Generate 1 thread evaluating how many chars in the vector are digits.
3. Generate 1 thread evaluating how many chars in the vector are punctuation marks.
4. Generate 1 thread evaluating how many chars in the vector are letters.

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 resemble what you remember together some "// comment" briefly summarizing what you were willing to use.

Domanda 2

Risposta non data

Punteggio max.: 3,00

Implement the POSIX system call

```
void pthread_barrier_wait (&barrier);
```

in the Windows system, using only **semaphores** and **mutexes**, in a situation in which **the barrier is inserted within a cycle which is repeated more than once**. Motivate and describe your solution with short comments.

Please, remind and use the following system calls.

```
HANDLE CreateMutex(LPSECURITY_ATTRIBUTES lpsa,BOOL fInitialOwner,LPCTSTR lpszMutexName);
BOOL ReleaseMutex (HANDLE hMutex);
HANDLE CreateSemaphore (LPSECURITY_ATTRIBUTES lpsa,LONG cSemInitial,LONG cSemMax,LPCTSTR lps
zSemName);
BOOL ReleaseSemaphore (HANDLE hSemaphore,LONG cReleaseCount,LPLONG lpPreviousCount);
DWORD WaitForSingleObject (HANDLE hObject,DWORD dwTimeOut);
```

Domanda 3

Risposta non data

Punteggio max.: 2,00

An ASCII file is organized as a sequence of lines. Each line specifies a shell command followed by all its parameters.
The following is a correct example of such a file:

```
ls -laR
wc --byte --words --lines
find . -name "*.c" -exec grep -H "foo" \{} \;
...
```

Write the function

```
void my_exec (char *name);
```

which receives the name of the file as a parameter, and it runs all commands specified in the file appropriately using the system calls **fork** and **exec**.

Please, remind the prototype of the following system calls:

```
pid_t fork (void);
int execv (char *path, car *argv[]);
int execvp (cahr *name, char +argvè]9;
int execve (char *path, char *argv[], char *envp[]);
```

Domanda 4

Risposta non data

Punteggio max.: 3,00

Write a small C++ program that computes the factorial of several numbers using asynchronous tasks and prints them altogether in the main program once all computations are completed. Each task will get its input from a queue container that is going to be filled by the main program as soon as the user enters the numbers.

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

In a Windows system, a **binary** file stores an undefined number of **fixed length records**. Each line includes the following fields:

- The identifier current_id of the current record, i.e., an integer value increasing at each line, and starting from 0 on the first line.
- Two strings, s1 and s2, each one of exactly 30 characters.
- A real (float) value f.
- The identifier next_id of the next record.

Overall, the structure of each record is the following one:

```
current_id    s1    s2    f    next_id
```

Essentially, each line of identified **current_id** points to (that is, refers or indicates) the line **logically** following it, i.e., the one with the identifier **next_id**. The following is the ASCII version of a correct binary file:

```
0  Harry  Potter  9.5  4
1  Hermione  Granger  9.0  7
2  Ron  Weasley  8.5  1
3  Albus  Silente  8.6  0
4  Severus  Piton  6.5  -2
5 ...
```

where, for example, the line of **current_id=0** (i.e., "Harry Potter 9.5") indicates the line with identifier **next_id=4** as next line (i.e., the line "Severus Piton 6.5"), which in turns does not have any line following it, as its **next_id** is equal to **-2.**

Write the procedure

```
void my_read (LPCTSTR file_name, DWORD current_id);
```

which opens the binary file of name **file_name**, and then it reads it, following the chain of identifier **current_id -> next_id**, with the following logic:

1. It reads and displays (on the standard output) the line with the identifier equal to **current_id**.
2. It moves on the record whose identifier is specified by the identifier **next_id** of the current line until such an identifier is a negative value or it identifies a record that does not exist in the file.
3. It repeats the entire process from step 1.

For example, with the previous file, starting from **current_id=3**, the function must display (on standard output) the following values:

```
Albus Silente 8.6
Harry Potter  9.5
Severus Piton 6.5
```

Note: if you do not remember the exact syntax of a specific system call, please write down a version that likely resembles what you remember together with some comments, briefly summarizing what you were willing to use.

Domanda 6

Risposta non data

Punteggio max.: 2,00

Two UNIX processes must transfer information between them.
Depending on

- The quantity of information the two processes need to transfer (a few bytes/sec or Gbytes/seconds)
- The type of information to transfer (formatted or byte-oriented)
- The relationship between the processes (related or unrelated)

describe the advantages and disadvantages of using pipes, FIFOs, message queues, and shared memory.