System and device programming
25 June 2015

Examination Time: 1h 45min. Evaluation. 18 marks.
Textbooks and/or course material allowed.
The final mark is the sum of the 1st and the 2nd parts

Write a concurrent C program `concurrent_file_processing. c` in the Unix environment which takes from the command line an argument `n ,` which must be an even integer, and generates `n A_threads` and `n B_threads.`
These threads perform the same task, but belong to two different types.
The synchronization among the threads follows these specifications:

- The main thread generates all the other threads, then it terminates.
- All the threads run concurrently, and are not cyclic.
- Then `A_threads` are created with an associated identifier `(0 to n-1).`
- Then `B_threads` are created with an associated identifier `(0 to n-1).`
- Each thread sleeps a random number of seconds (max 3), then it is supposed to process a file identified by the thread identifier, but in our case it does nothing.
- When a pair of threads of type A has processed their "files", one of them (the last) must concatenate the two files. In our case it simply prints for example:

  A4 cats: A4 A8
- When a pair of threads of type B has processed their "files", one of them (the last) must concatenate the two file, in our case it simply prints for example:

  B5 cats B5 BO
- When a pair of A_threads and a pair of B_threads have completed their concatenate operation, one of them (the last) must combine the four file. In our case it simply prints for example:

  A1 merges A1 A4 B3 B4

This is an example of output for the command `concurrent_file_processing 12`

```
A9  cats    A9  A4
>B3 cats    B3  B1
                    B3 merges    B3  B1  A4  A9
A2  cats    A2  A10
B7  cats    B7  B6
                    B7 merges    B7  B6  A2  A10
B9  cats    B9  B8
A5  cats    A5  A3
                    A5 merges    A5  A3  B8  B9
A7  cats    A7  A6
All . cats All A8
BO  cats    BO  Bll
                    BO merges    BO  Bll A6 A7
B4  cats    B4  B2
                    B4 merges    B4  B2  ` ` All
BIO  cats   i BIO B5
AO  cats    AO  A1
                    AO merges    AO  Al  B5  BIO
```

Hint: Use an array of counters with one counter per each `A_thread.`
Use an array of counters with one counter per each `B_thread.`
Manage these counters to get your solution.

System and device programming
25 June 2015

(Theory: no textbooks and/or course material allowed)

15 marks. The final mark is the sum of the 1st and the 2nd part.

1. (3.0 points) Write the sequence of instructions that allow the bash process to interpret and execute the command

   ```
   p1 | p2 > f1.txt
   ```

   where **`p1`** and **`p2`** are two executable files.

2. (3.0 points) List the steps to create a bootdisk with a minimal Linux kernel.

3. (4.0 points) Explain the behaviour of WaitForSingleObject and WaitForMultipleObjects in WIN32.
   Are calls to the two functions blocking ? What can we wait for, with the two functions ? How many, and which, different synchronization schemes are possible through WFMO ? Is it possible to use WFMO in order to wait for one among multiple events/objects ?
   What does constant WAIT_OBJECT_0 represent?

   Given the following loop, where the tHandles array is an array of handles of running threads, and processThreadResult works on the result produced by a thread, explain what the loop does.

   ```
   /* wait thread completion 1 */
   for (iThrd = 0; iThrd < N; iThrd++) {
      WaitForSingleObject (tHandles[iThrd], INFINITE);
      processThreadResult (tData[iThrd]);
   }
   ```

   Since the loop forces a given order in waiting for thread completion, write an alternative loop, based on WaitForMultipleObjects, where thread results are processed following an order given by thread completions.

4. (2.5 marks) Explain the main features of dynamic libraries in Win32. Motivate the main advantages of dynamic libraries vs. static ones. Explain the difference between implicit and explicit linking. What kind of modification is required by a program in order to become a dynamic library (answer for both implicit and explicit linking).

5. (2.5 marks) Which are the roles of files pointers and of the overlapped structures in direct file access on WIN32 systems. Briefly describe common aspects and differences. Provide a brief example of application for both of them.
   How can we increment by 100 bytes a file pointer in an overlapped structure? (provide an example)
   Does an overlapped structure include an event? Is it automatically created? When is it signaled ?