Operating Systems

29/06/2016

1. Write the pseudo-code of a concurrent program that includes **4 cyclic processes** (**P1**, **P2**, **P3**, P4). The initial statement of every process is sleep(r), where r is a random int. After executing **P1**, two instances of **P2** are executed, then after the **P2** processes, three instances of **P3** must be executed. At every cycle, process **P4** can be executed in parallel with processes **P1** or **P3**, but not in concurrency with processes **P2**. After all **P3** have terminated, the cycle starts again with **P1**. Use the minimum number of semaphores for synchronization.

Hint: write a solution ignoring **P4**, then update this solution adding **P4**, and the appropriate synchronizations.

1. Write the solution of the Producer & Consumer problem generalized as follows. You have **one** producer and an **infinite length** communication queue **Q**, and **N** consumers.
2. In a concurrent system, the OS resource manager controls 6 resources (R1,R2,R3,R4,R5,R6). At a given time 4 processes are running:
   * Process P1owns resources R1 and R4, and needs resources R2 and R5.
   * Process P2 owns resource R2, and needs resource R5
   * Process P3 owns resources R3 and R6
   * Process P4 owns resource R5, and needs resources R3 and R6

Process P1 will need in the future also resource R3.

Draw the resource allocation graph, and the wait-for graph. Explain how the system can detect whether it is in deadlock or not in general and in this particular case.

1. Write a **bash** script that takes a filename argument, the file contains a number of file pathnames, which may correspond to regular, directories, links, or other special files.

The script must output the number of regular files that fulfill these conditions: the file is owned by the user executing the script, and have dimension greater than 1Kbyte.

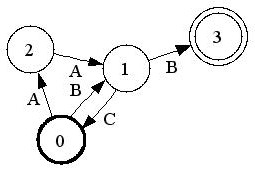
If the pathname refers to a directory, the script must also print the directory pathname and take into account the regular files on that directory that fulfill the same conditions. No further processing is necessary for nested sub-directories.

Finally, the script must print the sum of the lines of all these files.

**Exam questions 5 and 6 on the back side**

1. Implement an **awk** script that takes a filename argument, the file represents a graph similar to the one given as example in figure, by means of a sequence of transitions

**<initial state> <destination state> and <label>**:



**0 1 B**

**2 1 A**

**0 2 A**

**1 0 C**

**1 3 B**

**3**

andoutputs the graph using a representation

**<state> <list of adjacent states and corresponding label>**

The lists must be sorted according to the initial state, as shown in this example:

**0 1 B 2 A**

**1 3 B 0 C**

**2 1 A**

**3**

1. Write a **bash** script that takes as arguments a **username**, a group identifier (**gid**), and a **string** that represents the user first and last name. The script must append to file **/etc/passwd** a line that allows the system to recognize this user. In particular, the assigned user identifier (**uid**) will be computed by adding **1** to the **uid** of the user listed in the last row of file **/etc/passwd**, and the same interpreter.

This is an example of **/etc/passwd** line:

**laface:x:1001:1001:Pietro Laface,,,:/home/laface:/bin/bash**