GX1992

OLLSCOIL NA hÉIREANN, GAILLIMH
NATIONAL UNIVERSITY OF IRELAND, GALWAY

SEMESTER I EXAMINATIONS 2003/2004

SECOND YEAR COMPUTER SCIENCE [CS201]
OPERATING SYSTEMS and OBJECT ORIENTED PROGRAMMING in C++ [CS208]

Dr Dave Johnson
Prof. T. Hurley
Dr Niall Madden

Time allowed: **Two** hours.
Attempt **FOUR** questions.

Q 1. (a) Explain what a *process* is, in particular distinguish between a process and a program.
What are the *First-Come-First-Served* (FCFS), *Shortest-Job-First* (SJF), and *Round-Robin* (RR) schemes for process scheduling? Give an advantage and disadvantage of each.

(b) Suppose the following Process were submitted in the following order, all at time $t = 0$.

Process	Burst Time
P_1	20
P_2	10
P_3	8
P_4	2

For each of FCFS, SJF, and RR with a time quantum of $q = 5$ sketch the appropriate Gantt chart and calculate the average wait time for the processes.
(Assume that no CPU time is used when terminating a process or performing a context switch.)

(c) Using RR scheduling and the processes in (c), show that a shorter averaged wait time is obtained by taking $q = 2$.
Give a reason why, in a realistic setting, taking the smallest possible value for q may not result in the shortest possible wait times.

Q 2. (a) Suppose that a system has n processes, and a total of m resources that it can allocate. Resources can only be requested or released one at a time.
Show that the system is *Deadlock Free* if the following conditions hold:

(i) each process requires at least 1 resource and at most m resources.

(ii) the sum of all their requirements is less than $m + n$.

(b) Explain the "*Dining philosophers problem*" as a model for process synchronisation. Suggest a solution to the problem that involves

(i) Deadlock prevention or avoidance

(ii) Deadlock recovery

P.T.O

Q 3. (a) With regard to *Virtual Memory*, explain the term *Page Fault*. What action is take by the OS when a page fault occurs?

(b) Describe the FIFO, Optimal, and LRU algorithms for page replacement. What are the main disadvantages of the FIFO and Optimal algorithms?

Suppose a system has $F = 4$ frames, all empty to begin with. Show how each of FIFO, Optimal, and LRU would deal with the following reference sting:

$$\{1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5\}.$$

(c) What is *Belady's anomaly*? Using this page replacement string, but a different number of frames, show that FIFO suffers from it.

Q 4. You are a C++ programmer on a software team and have been asked to develop a class for storing characters in a *stack* (FILO queue). There should be methods for pushing characters onto the stack, popping them off, and checking the number of elements on the stack. The maximum number of items on the stack is fixed at 20.

(a) Describe how you would design the class. State clearly which members would be `private`, which would be `public`. Give the return values of the member functions and their argument lists. Describe what would be achieved by the constructor.

(b) Write down some C++ code that implements the class you designed in (a).

(c) Before your class can be integrated with the rest of the team's project, it must be tested. Give a short piece of C++ code that invites the user to enter a string, and using a stack as temporary storage, prints that string backwards.

Q 5. (a) Give a short example of a C or C++ program that demonstrates *interprocess communication*. The program should spawn a subprocess and then use a pipe to communicate the parent's Process Identification Number (PID) to the child.

(b) The software company you work for is building a multiuser database system. However, the first version fails miserably; the system doesn't prevent two users from trying to modify the same record at the same time, causing chaos (and a *race condition*).

Explain briefly (to your team manager) how a *binary semaphore* could be used to solve the problem.

Write down an implementation of a C++ class for the semaphore.

(c) Your team manager was impressed with your solution to the database problem. However, the testers have discovered a new problem: there are n printers attached to the system. A user is allowed to print to more than one printer at a time. However, if the total number of printers requested exceeds n, then the printer subsystem crashes.

The team manager turns to you to generalise your solution to (a) so that access to the n resources can be controlled.

Explain how you can use *function overloading* to extend the semaphore class from (a) so that it deals with the new situation of:

(i) having more than one instance of each resource,

(ii) being able to request more than one resource at a time,

but without having to recode any of the database software.

Show how the extended class can be coded in C++.