THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2014
Campus: City

COMPUTER SCIENCE

TEST

COMPUTER SCIENCE & SOFTWARE ENGINEERING

(Time Allowed: 45 minutes)

Note:

- Read these instructions carefully.
- Compare the exam version number on the Teleform sheet supplied with the version number printed in the top left corner of this page. If they do not match, ask the examination supervisor for a new sheet.
- No books, calculators or other electronic aids are allowed.
- Enter your name and student ID on the Teleform sheet. Your name should be entered left aligned. If you have a middle initial, enter it under M. If your name is longer than the number of boxes provided, truncate it.
- Use a dark pencil to mark your answers in the multiple choice answer boxes on the Teleform sheet. Check that the question number on the sheet corresponds to the question number in this question book. If you spoil your sheet, ask the supervisor for a replacement. Writing on the sheet will NOT be marked.
- Each question should have exactly one correct answer and carries the number of marks indicated.
- If you believe that a question erroneously may have no correct answer, choose the one you believe comes closest to a correct answer. If you believe that a question erroneously has more than one correct answer, choose whichever you believe may have been intended as the correct answer. In either case, please notify the course supervisor immediately after the exam.
- Take your question book home with you and keep it in a safe place. Writing on the question book will not be marked.
- This term test is marked out of 30 marks and is worth 10% of your final mark for this course**.**

Choose the BEST answer from the lectures and recommended reading.

Question 1
[1 mark] Which of the following is an operating system theme or model as defined in lectures?

(a) The onion model
(b) The distribution model
(c) The donkey model
(d) The processor model

Question 2
[1 mark] What is a resident monitor?

(a) It is a program which monitors the processes which are currently resident. It moves them between real and virtual memory as their memory requirements change.
(b) It is a program which monitors all of the processes in the system, ensuring that no process performs privileged instructions or accesses illegal memory.
(c) It is a program which remains in memory and passes control to the next program when the current program has finished. It can interpret simple job control commands.
(d) None of the above.

Question 3
[1 mark] Which of the statements about developing an operating system as a series of layers is TRUE?

(a) It makes debugging the operating system easier.
(b) It simplifies verifying that the operating system is correct.
(c) It is difficult to get the design of the layers right, especially as new requirements arise.
(d) All of the above.

Question 4
[1 mark] Unix is commonly thought of as comprising two separable parts, what are those parts?

(a) The kernel and the standard system programs.
(b) The standard system programs and the file system.
(c) The file system and the CPU scheduler.
(d) The kernel and the CPU scheduler.

Question 5
[1 mark] Early Unix had what type of operating system kernel?

(a) A client/server module kernel.
(b) A hybrid layered client/server kernel.
(c) A client/server microkernel.
(d) An all in one monolithic kernel.

Question 6
[1 mark] The first PDA and smartphone operating systems were most similar to which of the following types of historical operating systems?

(a) Time-sharing systems
(b) Hard real-time operating systems
(c) Resident monitors
(d) Batch systems

Question 7
[1 mark] Which of the following statements about spooling is FALSE?

(a) Spooling usually relies on interrupts generated by the IO devices when they are ready to supply or receive data.
(b) SPOOL stands for Simultaneous Peripheral Operation Off-Line.
(c) Spooling is commonly used to share one printer between different processes so that the output of each process does not interfere with the output of the other processes.
(d) Spooling removed the need for small computers to do IO processing because the main computer could deal with slow IO devices without slowing the performance of the rest of the system.

Question 8
[1 mark] Which of the following statements about symmetric multiprocessing (SMP) is FALSE?

(a) Most modern desktop operating systems are SMP systems because of multiple cores.
(b) SMP systems are tightly coupled.
(c) All processors run the same operating system in an SMP system.
(d) SMP allocates one processor to be the master processor and the other processors are slave processors.

Question 9
[1 mark] Which of the following statements about virtual machines is FALSE?

(a) The use of virtual memory makes virtual machines more complicated.
(b) Type 1 hypervisors run as applications on the host operating system.
(c) Type 1 hypervisors are better than type 2 for data centre implementations of virtual machines.
(d) Trap and emulate virtualization was not possible on early x86 architectures.

Question 10
[1 mark] Which of the following statements is NOT a reason that C is used to implement many operating systems?

(a) C provides low-level access to memory locations.
(b) The runtime requirements of C are low.
(c) C has well designed collection libraries.
(d) C maps easily to machine instructions.

Use this process burst time information in the next three questions. All processes are ready at the same time.

Process	Burst Time (ms)
X	12
Y	3
Z	6

Question 11
[1 mark] What is the average waiting time of these processes using first come first served, in the order X, Y then Z?

(a) 7
(b) 4
(c) 9
(d) 8

Question 12
[2 marks] If the processes are scheduled round-robin in the order X, Y then Z, which of the following time slice lengths would give the best average waiting time?

(a) 2
(b) 3
(c) 12
(d) 6

Question 13
[1 mark] What is the minimum average waiting time if the optimal schedule is chosen?

(a) 3
(b) 2
(c) 6
(d) 4

Question 14
[1 mark] Which of the following statements best explains what the fork system call does?

(a) It splits the process into two and both versions carry out the same code. There is no way to distinguish between the parent and the child process.
(b) It makes a complete copy of the calling process and starts running the child at the current point of execution.
(c) It makes a complete copy of the calling process and starts running the child at the beginning of the program.
(d) It makes a complete copy of the code of the calling process and runs with the same data in both the parent and child processes.

Question 15
[1 mark] Which of the following statements about shell commands is FALSE?

(a) Built-in commands usually access state which the shell maintains.
(b) Built-in commands can be used in a pipeline.
(c) Built-in commands always run in the shell process the user is interacting with.
(d) External commands always run in processes separate from the shell.

Here is some code from a Python shell program, refer to this code for the following 3 questions. The `commands` list would look something like `[["ls", ""-l"], ["grep", "robert"]]` when the code is called. This represents the line "`ls -l | grep robert`" typed at the shell prompt.

```
current_pid = os.fork()
if current_pid == 0:                    # line A
    while len(commands) > 1:
        command_list = commands.pop(0) # front of the pipeline
        command = command_list[0]      # command name
        rd, wr = os.pipe()
        if os.fork() == 0:
            os.dup2(wr, sys.stdout.fileno())
            os.close(rd)
            os.execvp(command, command_list)
        else:
            os.dup2(rd, sys.stdin.fileno())  # line B
            os.close(wr)
    command_list = commands.pop(0)
    os.execvp(command, command_list)
```

Question 16
[1 mark] What is the best description for what this code does?

(a) It starts all of the commands in a pipeline and runs them one after another in the shell process. As each command finishes it passes its output on as input to the next command to run. The commands run in series.
(b) It starts all of the commands in a pipeline in separate processes, redirecting the output from each command to be the input to the next command in the pipeline. The commands run in parallel.
(c) It starts all of the commands in a pipeline in separate processes, redirecting the output from each command to be the input to the next command in the pipeline. The commands run in series.
(d) It starts all of the commands in a pipeline and runs them one after another in the shell process. As each command finishes it passes its output on as input to the next command to run. The commands run in parallel.

Question 17
[1 mark] What is the purpose of line A "`if current_pid == 0:`"?

(a) It checks if the process id returned from the fork system call is that of the init process.
(b) It checks whether the fork system call succeeded.
(c) It checks the number of processes started by the call to fork.
(d) It checks whether the process is the child or the parent process after the call to fork.

Question 18
[1 mark] What is the purpose of line B "`os.dup2(rd, sys.stdin.fileno()`"?

(a) It changes the read end of the most recently created pipe to be the standard input file of the parent process. So reading from the pipe will come from the keyboard.
(b) It changes the standard input file of the parent process to be the read end of the most recently created pipe. So standard input will come from the pipe.
(c) It changes the read end of the most recently created pipe to be the standard input file of the child process. So reading from the pipe will come from the keyboard.
(d) It changes the standard input file of the child process to be the read end of the most recently created pipe. So standard input will come from the pipe.

Here is a very simple attempt at implementing a lock:

```
lock:
      while locked
      end
      locked = true

unlock:
      locked = false
```

Question 19
[1 mark] Which of the following statements about the above lock code is FALSE?

(a) It is unfair, there is no guarantee a thread will progress through the lock.
(b) It doesn't work, deadlock can occur very easily.
(c) It doesn't work, multiple threads could gain the lock simultaneously.
(d) It wastes CPU cycles checking the value of the locked variable.

Question 20
[1 mark] Which of the following is the best definition of an atomic instruction?

(a) The instruction stops all other processes or threads from working until it has completed.
(b) As an atomic instruction executes the operand values are only accessible from within privileged code such as the kernel.
(c) The instruction executes without any of its operands being accessible by other threads until the instruction has completed.
(d) Atomic instructions can only be broken down into smaller sub-atomic instructions.

Question 21
[1 mark] Which of the following best describes priority inversion and why it occurs?

(a) A process X with better priority is blocked waiting for a resource held by a process Y with worse priority because the resource is locked by process Y.
(b) A process X with better priority is blocked waiting for a resource held by a process Y with worse priority because process Y is not scheduled as there are other runnable processes with better priorities.
(c) A process X with worse priority is blocked indefinitely by a process Y with better priority because process Y has locked a resource needed by process X.
(d) A process X with worse priority has its priority improved because a process Y with better priority is waiting for a resource held by process X.

Question 22
[1 mark] Which of the following statements about semaphores is TRUE?

(a) A binary semaphore can be used in the same way as a simple lock.
(b) A semaphore is an integer count with some indivisible operations and an initialisation.
(c) Returning a resource when no process is waiting causes the semaphore value to increase.
(d) All of the above.

Use this information for the next two questions. Here are some possible schedules for two real-time processes A (2, 5, 5) and B (5, 9, 9) - where the numbers of compute time, period and deadline are all in milliseconds. The tables show only the first 10 milliseconds of the calculated schedules. Each millisecond is used to recalculate the priorities.

X)

A	A	B	B	B	B	B	A	A	B

Y)

A	A	B	B	B	A	A	B	B	B

Z)

A	A	B	B	B	B	A	B	A	B

Question 23
[2 marks] Which of the schedules is generated by Earliest Deadline First?

(a) Y
(b) Z
(c) X
(d) None of the above.

Question 24
[2 marks] Which of the schedules is generated by Least Slack Time? If the slack time is the same for both A and B give the priority to the currently running process, if there is no currently running process choose A.

(a) X
(b) Z
(c) Y
(d) None of the above.

Question 25
[1 mark] Which of the following is NOT a component of a monitor?

(a) a shared resource which is protected by the monitor
(b) publicly accessible entry points
(c) a deadlock detector
(d) a scheduler

Here is a pseudocode semaphore implementation of the Dining Philosophers' problem. The semaphores are initialised to 1.

```
do forever:
  status = "waiting"
  wait(left)
  wait(right)
  status = "eating"
  signal(left)
  signal(right)
  status = "thinking"
```

Question 26
[1 mark] In the above pseudocode solution to the Dining Philosophers' problem, what could go wrong?

(a) An unlucky process might never be able to get both left and right forks simultaneously, even though other processes are eating.
(b) All processes might pick up one fork causing deadlock.
(c) Nothing is wrong, this is a good solution to the problem.
(d) Some processes will get extra turns to eat on a regular basis, violating the principle of fair treatment.

Question 27
[1 mark] Which of the following statements about concurrency constructs is TRUE?

(a) Locks are at least as powerful as monitors.
(b) Semaphores are at least as powerful as locks.
(c) Monitors are at least as powerful as semaphores.
(d) All of the above.

Rough Working – This page will not be marked

(You may detach this page from the question booklet and use it for rough working)