# THE UNIVERSITY OF AUCKLAND

**Second SEMESTER, 2013**
**Campus: City**

**COMPUTER SCIENCE & SOFTWARE ENGINEERING**

## TEST

**Operating Systems**

**(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 models can be applied to operating systems?

(a) The manager model
(b) The onion model
(c) The dustbin model
(d) All of the above

**Question 2**
[1 mark] The MS-DOS operating system was really which type of system?

(a) A time-sharing system
(b) A resident monitor
(c) A microkernel
(d) A strictly layered system

**Question 3**
[1 mark] Which of the following is NOT an advantage of an all-in-one operating system design?

(a) It can have the most functionality in the least space
(b) Access to different components within the operating system is efficient
(c) It is initially simpler to implement
(d) It is easier to verify the functioning of the operating system

**Question 4**
[1 mark] What was the major task of job control languages?

(a) To suspend and resume jobs
(b) To give commands to the operating system about processes and data
(c) To take control away from the operating system and give it to user level jobs
(d) To prevent jobs getting into infinite loops and executing continuously

**Question 5**
[1 mark] Which of the following is a requirement in order to create a time-sharing system?

(a) A relatively large amount of memory
(b) Interruptible processors
(c) Memory protection
(d) All of the above

**Question 6**
[1 mark] Early smartphones had which of the following features?

(a) Fast processors
(b) Virtual memory
(c) Large amounts of memory
(d) None of the above

**Question 7**
[1 mark] Which of the following is NOT an advantage of virtual machines?

(a) Individual servers can be provided on a smaller number of real machines.
(b) Multiple operating systems can run simultaneously on the same hardware.
(c) There is a large increase in flexibility for data centres using virtual machines.
(d) Virtual machines simplify the overall design of operating systems.

**Question 8**
[1 mark] Which of the following was NOT included in Popek and Goldberg's requirements of virtualization?

(a) Performance - most instructions in the virtual machine should be run directly on the hardware.
(b) Fidelity - the software should run identically on the virtual machine as on a real machine.
(c) Simplicity - the implementation of the virtual machine monitor must be clear and simple.
(d) Safety - each virtual machine, and the virtual machine monitor should be safe from actions in another virtual machine.

**Question 9**
[1 mark] Instructions which query the running state of a processor (i.e. kernel or user mode) make virtualization difficult. Which of the following best explains why?

(a) An instruction running in the kernel of a virtual machine would report that it was running in user mode.
(b) An instruction running in the kernel of the host operating system would report that it was running in user mode.
(c) An instruction running in user mode of a virtual machine would report that it was running in kernel mode.
(d) An instruction running in the kernel of the host operating system would report that it was running in kernel mode.

**Question 10**
[1 mark] Even with the Global Interpreter Lock (GIL) in standard Python which means that only one thread can run at a time there is still a need for locks (as in the first assignment). Which of the following reasons best explains why this is so?

(a) The GIL is associated with threads, the locks which are still required are associated with processes.
(b) A thread could still be interrupted (and another thread scheduled) while it is accessing a resource.
(c) If run on a multicore machine more than one thread can run simultaneously in the Python program.
(d) The GIL sometimes doesn't work allowing multiple threads to run simultaneously.

**Question 11**
[1 mark] Spin locks are required in some situations. Which of the following reasons justifies the use of spin locks?

(a) To provide higher level synchronization constructs such as semaphores with queues to put threads to sleep safely, we can make the operations appear atomic by locking parts of them with spin locks.
(b) If the spin part of the spin lock is guaranteed to only take a very short period of time.
(c) In a multicore machine a spin lock is not so bad because work can still be done on the other cores.
(d) All of the above

**Question 12**
[1 mark] Here is the lecture version of Peterson's software solution to the two thread lock problem:

```
lock:
      flag[i] = true
      turn = j
      while (flag[j] && turn == j)
      end

unlock:
      flag[i] = false
```

Which of the following statements about this lock is TRUE?

(a) It can be generalized to more than two threads by simply extending the size of the flag array.
(b) The lock could allow one thread to enter the critical section twice if the other thread is waiting to enter the critical section.
(c) The `i` variable in this code represents an identifier associated with the other thread.
(d) The lock relies on the fact that the line "turn = j" is executed atomically by each thread.

**Question 13**
[1 mark] Which of the following is NOT an advantage system level threads have over user level threads?

(a) They don't block threads in the same process when one of them blocks in the kernel.
(b) On a multicore machine system level threads from the same process can run simultaneously on the processor cores.
(c) They are simpler to create and switch between.
(d) Each system level thread can be scheduled independently by the operating system.

**Question 14**

[1 mark] Given the following code what output comes from the child process? (The “end” keyword used in the print function means the output is separated by spaces rather than newlines.)

```
import os

def test():
    print(1, end=' ')

pid = os.fork()

if pid == 0:
    print(2, end=' ')
    test()
else:
    test()
    print(3, end=' ')

print(4, end=' ')
```

(a) 2 1 1 3 4
(b) 2 1 4
(c) 1 3 4
(d) 1 3 4 2 1 4

**Question 15**

[1 mark] How many times does the ps command get called from the following program? Remember that in Python 0 is equivalent to false.

```
import os

count = 0
os.system('ps')
while count < 2:
    if os.fork():
        os.system('ps')
        if os.fork():
            os.system('ps')
    count += 1
```

(a) 9
(b) 7
(c) 5
(d) 3

**Question 16**

[1 mark] In assignment 1 part 1 which two threads/processes could access the ready_list at the same time?

(a) The scheduler thread and the simple process thread
(b) The high and low priority processes
(c) The scheduler and controller threads
(d) The main process and the high priority process

**Question 17**

[1 mark] In Assignment 1 what best describes the purpose of the following line of code?

```
os.waitpid(current_process.pid, os.WNOHANG) != (0, 0)
```

(a) It waits until the current process has finished.
(b) It checks to see if the current process is not hanging.
(c) It waits until the current process is no longer hanging.
(d) It checks to see if the current process has finished.

**Question 18**

[2 marks] The following output from assignment 1 shows that a low priority process gets the resource for a second and third time even though two higher priority processes have requested the resource. What is the best explanation for this behaviour?

```
Low priority: 67130 - just about to request resource
owner pid: 67130
Low priority: 67130 - got resource
High priority: 67131 - just about to request resource
owner pid: 67130
High priority: 67132 - just about to request resource
owner pid: 67130
Low priority: 67130 - just about to request resource
owner pid: 67130
Low priority: 67130 - got resource
Low priority: 67130 - just about to request resource
owner pid: 67130
Low priority: 67130 - got resource
```

(a) The low priority process has had its priority improved because of the waiting higher priority processes.
(b) The high priority processes are letting the low priority process complete in order to get it out of the way as soon as possible.
(c) The low priority process was first in the queue and hence gets the resource repeatedly.
(d) The low priority process requests the resource while it is still holding the resource.

**Question 19**

[1 mark] What is true about the following lines of code from assignment 1?

```
r, w = os.pipe()
controller_read = os.fdopen(r)
controller_write = os.fdopen(w, mode='w', buffering=1)
```

(a) This pipe needs to be created before any simple processes which require the resource are forked.
(b) This pipe is used to convey request-reply-release messages to the processes which require the resource.
(c) This pipe is used to coordinate access to the ready_list.
(d) The controller needs to be able to write to this pipe.

**Question 20**

[1 mark] Which of the following would you NOT normally expect to be in a Process Control Block?

(a) References to other processes
(b) References to allocated memory
(c) References to device drivers
(d) References to open files

**Question 21**

[1 mark] Which of the following statements about the Unix exec system call is FALSE?

(a) Exec checks to see if the file passed to it is executable.
(b) Exec copies the currently running process then replaces it with a new program.
(c) All files which were open in the process which called exec remain open in the process after the exec.
(d) Calling exec throws away the memory associated with the process.

**Question 22**

[1 mark] Which of the following statements concerning the Windows operating system is FALSE?

(a) The Windows equivalent of a Process Control Block is stored in a variety of data structures.
(b) Windows DLLs are loaded within the context of the newly created process.
(c) Windows process creation is a version of Unix process creation.
(d) The Windows NT Kernel can use different environmental subsystems to run user applications.

**Question 23**

[1 mark] Why is suspending a thread NOT generally regarded as a good idea?

(a) If a suspended thread holds locks it could cause deadlock.
(b) Suspended threads take up unnecessary CPU cycles.
(c) Suspended threads release any locks they are holding, possibly leaving resources in inconsistent states.
(d) If a suspended thread releases locks it could cause lifelock.

**Question 24**

[1 mark] Which of the following statements concerning the Unix operating system is FALSE?

(a) A waiting process is placed on a queue associated with the kernel address of the resource the process is waiting for.
(b) A process with a priority of 40 has a better priority than a process with priority 20.
(c) Signals are used to notify processes when they have done something wrong.
(d) A process only stays a zombie if its parent is still alive.

**Question 25**

[1 mark] In the following pseudocode solution to the Dining Philosopher's problem, what could go wrong? The simultaneous_wait only returns when both parameters are available.

```
do forever:
  status = "waiting"
  simultaneous_wait(left, right)
  status = "eating"
  simultaneous_signal(left, right)
  status = "thinking"
```

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

**Question 26**

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

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

**Question 27**

[2 marks] Given three processes A, B, and C with corresponding burst times 20, 5, and 4, what is the average waiting time with a round-robin scheduler using a time slice of 5? The ready list is initially A, B, then C.

(a) 3
(b) 15
(c) 8
(d) 9.66...

**Question 28**

[1 mark] With a logical clock system and assuming that the clocks work in integers what should the local logical clock of a system be changed to if its current value is 4 and it receives a message from another system with a time stamp of 6.

(a) 6
(b) 5
(c) 3
(d) None 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)