

THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2012

Campus: City

Computer Science / Software Engineering

TEST

Operating Systems

(Time Allowed: 45 minutes)

Note:

- The use of calculators is NOT permitted.
- Compare the test version number on the Teleform sheet supplied with the version number above. If they do not match, ask the supervisor for a new sheet.
- Enter your name and student ID on the Teleform sheet. Your name should be entered left aligned. If your name is longer than the number of boxes provided, truncate it.
- Answer all **Multiple-choice** questions on the Teleform answer sheet provided. Attempt all questions. There are no negative marks.
- Use a dark pencil to mark your answers in the multiple choice answer boxes on the Teleform sheet. If you spoil your sheet, ask the supervisor for a replacement.
- There are **30 marks** in this test.

CONTINUED

Choose the BEST answer from the lectures and recommended reading.

Question 1

[1 mark] Which of the following is NOT an operating system theme or model as defined in lectures?

- (a) The donkey model
- (b) The onion model
- (c) The resource allocator model
- (d) The manager model

Question 2

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

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

Question 3

[1 mark] Which of the following best describes the task of a resident monitor?

- (a) It supplied a disk operating system with spooling of input and output.
- (b) It enabled several people to use the computer at the same time from multiple terminals.
- (c) It cleared memory used by one program, automatically loaded the next program and jumped to its starting point.
- (d) It kept several programs in memory at once and scheduled them according to time slices.

Question 4

[1 mark] What change contributed most to the development of time-sharing systems?

- (a) Spooling - simultaneous peripheral operation online.
- (b) Multiple scheduling queues.
- (c) Disk drives became widely available.
- (d) Computers became cheaper.

CONTINUED

Question 5

[1 mark] How did a time-sharing operating system differ from a batch operating system?

- (a) The response time was slower on a time-sharing system.
- (b) Security was simpler in time-sharing systems.
- (c) Time-sharing systems allowed multiple processes to be running simultaneously.
- (d) Compared to a batch system many of the scheduling decisions could no longer be made by the operating system.

Question 6

[1 mark] The first 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) Batch systems
- (d) Resident monitors

Question 7

[1 mark] Which of the following is NOT a reason that virtual machines are popular?

- (a) Virtual machines allow you to run programs from one operating system on top of another operating system.
- (b) Multiple servers, each in its own virtual machine, can run on one computer.
- (c) Virtual machines can emulate any CPU architecture.
- (d) Virtual machines can be easily moved from one machine to another.

Question 8

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

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

Question 9

[1 mark] Which of the following best describes what application virtualization is?

- (a) An API library layer is provided which allows an application designed to run on one operating system to run on another operating system.
- (b) A guest operating system runs as an application on a host operating system.
- (c) Modifications are made to the source code of an application so that it can run efficiently on another operating system.
- (d) Multiple virtual machines with the same kernel are implemented as applications on a host machine also running the same kernel.

Question 10

Here is a very simple attempt at implementing a lock:

```
lock:
    while locked
    end
    locked = true

unlock:
    locked = false
```

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

- (a) It doesn't work, multiple threads could gain the lock simultaneously.
- (b) It is unfair, there is no guarantee a thread will progress through the lock.
- (c) It doesn't work, if a thread holds the lock and goes to sleep no other threads can gain access.
- (d) It wastes CPU cycles checking the value of the locked variable.

Question 11

[1 mark] Why are spin locks particularly bad when run on a single core processor?

- (a) Running in the spin lock is pointless as the thread holding the lock cannot run and therefore release the lock at the same time.
- (b) As only one thread can run on a single core processor there is no need for a lock of any kind, including spin locks.
- (c) Because time-slice preemption is the only way another thread can run on a single core, it is possible for the spin lock to fail.
- (d) Single core processors get overheated more easily if a thread is running in a spin lock.

Question 12

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

- (a) As an atomic instruction executes, intermediate states of the operand values are only accessible from within the same process.
- (b) Atomic instructions can only be broken down into smaller sub-atomic instructions.
- (c) The instruction stops all other processes or threads from working until it has completed.
- (d) The instruction executes without any of its operands being visible or modifiable by outside changes until the instruction has completed.

Question 13

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
```

[1 mark] Which of the following statements about this lock is FALSE?

- (a) The lock relies on the fact that the line "turn = j" is executed atomically by each thread.
- (b) The lock guarantees that each thread will eventually progress past the lock and enter the critical section of code.
- (c) The lock will not allow one thread to enter the critical section twice if the other thread is waiting to enter the critical section.
- (d) It can be generalized to more than two threads by simply extending the size of the flag array.

Question 14

[1 mark] Which of the following statements about the x86-64 xchg instruction is FALSE?

- (a) It can be used to implement correctly working locks.
- (b) It can be used to atomically swap the values of two memory locations.
- (c) It can be used to atomically swap the values of two registers.
- (d) It atomically swaps the values in its two operands.

Question 15

[1 mark] How does the Bakery algorithm guarantee a unique ordering of process requests to use a shared resource?

- (a) The algorithm uses the scheduling priority of each process as its unique ordering identifier.
- (b) A hash value is generated from the process id of the process and the ticket number distributed by the algorithm.
- (c) The ticket number distributed by the algorithm is kept unique by the use of a spin lock.
- (d) The process id of the process is appended to the ticket number distributed by the algorithm.

CONTINUED

Question 16

[1 mark] Which of the following problems does not occur as a possible consequence of locking resources?

- (a) Priority inversion
- (b) Deadlock
- (c) Indefinite postponement
- (d) Inconsistent data

Question 17

[1 mark] What is priority inheritance?

- (a) The solution to indefinite postponement, where a process waiting for a resource gets the same priority as the process currently using the resource.
- (b) The solution to priority inversion, where a process using a resource has its priority temporarily increased to that of the highest priority process waiting for the resource.
- (c) The solution to indefinite postponement, where all processes using or waiting for a resource get the maximum priority of all processes waiting for the resource.
- (d) The solution to priority inversion, where all processes waiting on a resource temporarily get the same priority as the process currently using the resource.

Question 18

[1 mark] Which of the following statements about semaphores is correct?

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

Question 19

```
#include <stdio.h>

int answer = 6;

void main() {
    int x = 3;
    int y = 5;
    printf("answer1: %d\n", answer);
    __asm__ ("addl %1, %2\n\t"
            "movl %2, %0"
            : "=m" (answer)
            : "r" (x),
            "r" (y)
            );
    printf("answer2: %d\n", answer);
}
```

[1 mark] What output is produced by the second printf function in the program above?

- (a) answer2: 2
- (b) answer2: 8
- (c) answer2: 5
- (d) answer2: 6

Question 20

[1 mark] In assignment 1 it was possible for the elapsed time of the program to be considerably shorter than the sum of the user and system times. What is the best explanation for this?

- (a) If there were multiple cores it was possible for more than one thread to be running simultaneously.
- (b) The system times and users times were only approximations of the actual values, whereas the elapsed time was correct.
- (c) If there were multiple threads it was possible for several threads to be blocked in a spin lock.
- (d) None of the above.

Question 21

[1 mark] Which of the following conditions would be expected to cause the greatest number of context switches when running the programs of assignment 1?

- (a) 8 threads running with the spin lock solution.
- (b) 1 thread running with the no lock solution.
- (c) 8 threads running with the no lock solution.
- (d) 1 thread running with the spin lock solution.

Question 22

[1 mark] The definition “an instance of a program in execution” best applies to which of the following?

- (a) a thread
- (b) a process
- (c) a system-level resource
- (d) a user-level resource

Question 23

[1 mark] Which of the following is NOT an advantage of system-level threads over user-level threads?

- (a) Threads from the same process can run simultaneously on different cores/processors
- (b) Each thread can be scheduled separately
- (c) Switching between threads is faster
- (d) Blocking threads don't stop the entire process

Question 24

[1 mark] Which of the following statements is FALSE concerning Solaris threads before version 9?

- (a) Each lightweight process was associated with one kernel thread.
- (b) There could be several kernel threads associated with one user-level thread at the same time.
- (c) Several user threads could be associated with one lightweight process.
- (d) Extra lightweight processes were added if necessary when a thread blocked in the kernel.

Question 25

[1 mark] With original Linux threads the clone system call created a new process which shared all of its memory with the original process. The new process was then used as a thread of the original process. Which of the following statements was TRUE about original Linux threads?

- (a) If one thread called exec to run another program, all threads in the same process would be killed.
- (b) If a thread made a blocking system call, the other threads in the same process could still continue.
- (c) If one thread in the process finished, all threads finished.
- (d) All of the above

Question 26

[1 mark] Which of the following statements about the producer/consumer problem is FALSE?

- (a) Each item of data produced must be consumed only once.
- (b) Each item of data produced must be consumed.
- (c) The order of data being consumed must match the order it was produced.
- (d) None of the above.

Question 27

[1 mark] Which of the following statements about multitasking is FALSE?

- (a) The use of cooperative multitasking over preemptive multitasking can make the design of the kernel simpler.
- (b) Cooperative multitasking means that all processes run to completion before allowing other processes to run.
- (c) Preemptive multitasking provides greater predictability of when processes will be scheduled compared to cooperative multitasking.
- (d) Multitasking means that multiple processes can take turns to execute on the same CPU in such a way that it appears that they are all running simultaneously.

Question 28

[1 mark] What is normally meant by a “context switch”?

- (a) Any change in state of a process as it executes.
- (b) Saving the state of one process and restoring the state of another process so that the CPU moves from running one process to running the other.
- (c) The switch between a process running in user mode to running in kernel mode. This can happen as the result of a system call or an exception.
- (d) The change which occurs on the stack of a running process as it makes a function call, including the allocation of memory for any parameters and the local variables.

Question 29

[2 marks] Given three processes A, B, and C with corresponding burst times 20, 5, and 2, what is the average waiting time with a first come first served (FCFS) scheduler? All processes are ready to run at the same time but they arrived in the order A, B, then C.

- (a) 3
- (b) 15
- (c) 9
- (d) 25

Rough Working – This page will not be marked