COMP 3000B: Operating Systems

Winter 2007 Mid-Term Exam Solutions

February 15, 2007

1. Processors have memory management hardware primarily to:

   (a) increase the amount of physical memory available to the processor.
   (b) accelerate physical memory access.
   (c) **accelerate virtual address space computations.**
   (d) improve security.

2. "Zombie" processes are caused by:

   (a) An execve of a non-existent program
   (b) **A failure to wait for terminating processes**
   (c) A fork when system resources are low
   (d) All of the above

3. When a regular process is running on a single-CPU system, all of the following are true **except**:

   (a) The kernel is not running.
   (b) Disks can be writing data to memory.
   (c) The CPU will generate an exception (software interrupt) if certain addresses are accessed.
   (d) **The CPU is in supervisor mode.**

4. Monitors:

   (a) use condition variables for synchronization
   (b) prevent multiple threads from executing monitor code at the same time
   (c) hide mutual exclusion details from calling routines
   (d) **all of the above**

5. System calls must be used to:

   (a) modify global variables
   (b) call a user-written function
   (c) **write to a file**
   (d) All of the above

6. Which of the following is an example of a bounded-buffer producer consumer problem?

   (a) **UNIX pipes, e.g. the command "ls | more"**
   (b) two threads sharing access to `struct foo`
   (c) browsing and buying a plane ticket online
   (d) All of the above

7. A modern monolithic operating system kernel typically implements

   (a) scheduling & address space management

   (b) some device drivers

   (c) a TCP/IP stack

   (d) **All of the above**

8. When a UNIX shell such as bash executes an external command (such as ls), what system calls does it typically execute?

   (a) fork

   (b) wait

   (c) **Both (a) and (b)**

   (d) None of the above

9. Which of the following strategies would prevent deadlock with respect to shared data structures A, B, and C?

   (a) Organizing A, B, and C such that concurrent access to them is always safe.

   (b) A background watchdog thread that automatically releases locks on A,B, or C if they are held but not used for 60 seconds.

   (c) A strictly-enforced coding convention where A, B, and C are always locked in reverse-alphabetical order

   (d) **All of the above**

10. An attacker wants to disable a multi-user system but he only has ssh access as the regular user "fred" (i.e. he can't get root access). If the system isn't properly configured, he may be able to prevent others from productively using the system by:

    (a) Creating lots of processes

    (b) Running several programs that consume lots of memory

    (c) Creating many, many large files on disk

    (d) **All of the above**

11. A typical executing application is best described as a:

    (a) kernel

    (b) **process**

    (c) thread

    (d) system call

12. Special machine instructions such as "test and set" can be used to protect data structures shared between:

    (a) processes running on multiple networked computers

    (b) **threads running on a single CPU**

    (c) both (a) and (b)

    (d) none of the above

13. Pipelining. . .

    (a) . . .involves dividing machine instructions into smaller units of work.

    (b) . . .allows modern CPUs to execute more than one instruction in parallel.

    (c) . . .slows down context switches and increases interrupt latency.

    (d) **All of the above**

14. DMA. . .

    (a) . . . stands for "direct memory access."

    (b) . . . can be slower than programmed I/O.

    (c) . . . is used extensively by current desktop computers.

    (d) **All of the above.**

15. A batch system shares CPU resources by executing programs one at a time, with each running to completion. Compared with a timesharing OS system, batch operating systems:

    (a) make more efficient use of CPU resources (i.e. less is wasted in overhead)

    (b) are easier to implement

    (c) are used today on mainframe systems

    (d) **All of the above**

16. When choosing whether to implement a device driver as a block or a serial device, which of the following information about the new device is **most** relevant?

    (a) **The same data is likely to be read multiple times.**

    (b) The device returns data in fixed-sized chunks.

    (c) The device is fast.

    (d) The device is attached by USB.

17. If the same non-preemptible resources are allocated in varying orders by multiple threads or processes, can deadlock (involving resources) ever happen?

    (a) **yes, deadlock can happen**

    (b) no, deadlock can never happen

18. When a running program X requests data from a file F whose contents are on disk, the OS will:

    (a) save X's current state
    (b) schedule a disk request for F's data blocks
    (c) load the state of another ready-to-run program Y (which may be X)
    (d) **All of the above**

19. Which UNIX command will show you a list of currently running processes?

    (a) chmod
    (b) **ps**
    (c) more
    (d) ls

20. When a web browser requests a web page, it is performing a type of inter-process communication. This communication is best described as:

    (a) **message passing**
    (b) shared memory IPC
    (c) semaphore-based synchronization
    (d) None of the above

21. Disabled interrupts are effective at enforcing mutual exclusion in which of the following contexts?

    (a) an OS kernel on a a symmetric multiprocessor (SMP) system
    (b) an OS kernel on a cluster of networked computers
    (c) **an OS kernel on a single processor system**
    (d) a web-based distributed application

22. "Elevator" scheduling is used by:

    (a) Ethernet cards
    (b) Flash drive controllers
    (c) **Hard disk controllers**
    (d) Operating system CPU schedulers

23. Producer/consumer programs use mutual exclusion mechanisms (semaphores, monitors, etc.) to:

    (a) slow down consumers that run faster than producers
    (b) slow down producers that run faster than consumers
    (c) prevent the shared queue from becoming corrupted
    (d) **all of the above**

24. "ls -l hello" outputs the following:

```
-rw-r-xr--  2 soma sys 82021 2005-11-20 10:06 hello
```

Based on this information, which of the following is true about hello?

(a) The user soma may execute hello.

(b) **Members of the group "sys" may execute hello.**

(c) The file hello was created on November 20, 2005.

(d) All of the above

25. [4] A system administrator would like to allow an apache web server process to log incoming requests; however, she also wants to configure the system such that if the web server process is compromised by an attacker, that attacker won't be able to modify or erase past transactions.

(a) [2] What UNIX permission(s) would the web server process need on its logfile? What Windows permission(s) would it need?

The web server would need write access on both UNIX and Windows.

(b) [2] Would these permissions prevent a compromised web server from erasing the log of past events? Explain.

No, because write access would allow the web server to change or erase past entries. What we would really want are "append-only" files—such file permissions do exist on some systems (Linux's ext3 filesystem supports append-only files); however, they are not available in standard UNIX or Windows OSs.

26. [2] What is the primary advantage to placing windowing system code in the OS kernel? What is the primary disadvantage?

Primary advantage: speed (i.e. reducing context switches). Primary disadvantage: loss of stability/security due to bugs. Note that it takes a lot of code to build a modern windowing system!

27. [2] UNIX pipes are an example of a bounded-buffer producer consumer problem. On what system call will the producing process block when it gets ahead of the consumer? On what system call will the consumer block when it gets ahead of the producer? (Hint: what system calls are used when a process accesses a pipe?)

Producers will block on write, while consumers will block on read.

28. [3] Answer the following questions on semaphores:

(a) [1] What is the basic strategy for using a binary semaphore (mutex) to prevent concurrent access to a shared data structure?

First grab the semaphore (lock it) before accessing the data structure (this may require sleeping if the lock isn't initially available). Then, release the semaphore (unlock it) when the access is complete.

(b) [2] If the semaphore is used incorrectly, what problems can arise? State two simple errors and the problems these errors can cause with the execution of concurrent threads and/or the state of the shared data structure.

If the semaphore isn't grabbed before accessing the data structure interleaved concurrent accesses may occur, corrupting the data structure. If the semaphore isn't released, other accessing threads will wait forever for access (they'll starve).

29. [4] Answer the following questions on UNIX environment variables.

(a) [1] Who ultimately determines the value of a new process's environment variables?

The parent process.

(b) [1] Can UNIX environment variables such as PATH be changed by a process at runtime, or are they constants?

They can be changed at runtime.

(c) [2] Where do you think UNIX environment variables stored—in kernel memory, or in the address space of individual processes? Explain how you came to this conclusion (or give evidence for why your answer is correct).

Environment variables are stored in the address space of the process. There are no restrictions on how many environment variables may be created, and any environment variable may be deleted (e.g. they can be manipulated using bash's "export" command). In general data is stored in kernel memory only if access to them is restricted. Since there appear to be no restrictions on environment variables, they should be stored in the address space of the process.

30. [5] Briefly answer the following questions on process and thread management:

(a) [1] A running program consists of one or more execution contexts plus an address space. In terms of execution contexts and address spaces, what is a process? What is a thread?

A process is one or more execution contexts plus an address space. A thread is an execution context (or, one execution context and an address space).

(b) [2] When a thread exits, does a process exit? When a process exits, does a thread exit? Explain.

When a thread exits, a process does not necessarily exit because it may have other threads executing. If a process exits, it implies that all of its threads exit (so at least one thread must exit).

(c) [2] Kernels in many operating systems support mulitple execution contexts within the kernel, i.e. part of the kernel may be blocked waiting for an I/O event while another is processing a system call. Are these "execution contexts" best thought of as threads or processes? Why?

They are better throught of as threads, as they all share a single address space.

31. [2] Older operating systems largely relied on static linking, while modern operating systems generally use dynamic linking. With static linking, only the needed parts of libraries are incorporated into the executable; with dynamic linking, entire libraries are incorporated into the process at runtime.

Dynamic linking has a number of runtime costs; however, it also saves memory on modern operating systems. How is it possible for dynamic linking to consume fewer memory resources than static linking? Explain.

Dynamic linking allows library code to be shared between running processes via shared read-only pages. Such sharing does not normally happen between processes running different executables is possible with static linking.

For example, only one copy of the code for printf() has to be present in all of physical memory, even if 20 processes running 10 different executables all all currently running (and all use printf). With static linking, we'd have at least 10 copies of the printf() code in memory.

32. [4] Most current portable electronics devices do not run modern operating systems. However, Apple has stated that the iPhone will run Mac OS X, which is a modern OS.

    (a) [1] What key piece of hardware will the iPhone need that other portable devices do not?

    The iPhone needs an MMU (memory management unit) to enable the use of virtual memory. Note that Windows CE (PocketPC) and Symbian also require an MMU; most portable devices (most mobile phones, MP3 players, PalmOS devices, etc.) do not—even if they run 3rd party programs.

    (b) [1] What is one compelling reason for Apple to choose a modern OS foundation for the iPhone?

    Compatibility with existing applications written for MacOS X and other UNIX variants is one reason. Probably just as compelling is the relative ease of development on a modern OS (better tools, better crash recovery, more robust services, etc.).

    (c) [2] Apple has stated that the iPhone is to be a "closed" device, meaning that users will not be able to install arbitrary applications on it; instead, they will only be able to install Apple-supplied (or authorized) programs.

    What are two capabilities that Apple will need to restrict on the iPhone that are available to users of conventional modern, personal operating systems (e.g. Windows XP, regular versions of Mac OS X)?

    Regular users must not be able to execute arbitrary binaries: the kernel must check that each program has an appropriate digital signature before execution. Also, regular users must not be allowed to modify the OS kernel in arbitrary ways (no custom device drivers or other extensions); otherwise, the signing checks could be bypassed.