

Fall 2017 COMP 3511 Homework Assignment #1 Solution

Handout Date: September 14, 2017 Due Date: September 28, 2017

Name: _____ ID: _____ E-Mail: _____

Please read the following instructions carefully before answering the questions:

- You should finish the homework assignment **individually**.
- There are a total of **4** questions.
- When you write your answers, please try to be precise and concise.
- Fill in your name, student ID, email at the top of this page.
- Please fill in your answers in the space provided, or you can type your answers in the Microsoft Word file.
- **Homework Collection:** the **hardcopy** is required and the homework is collected in **collection box #20**. The collection boxes locate outside Room 4210, near lift 21 (there are labels attached on the boxes).

1. (20 points) Multiple choices.

1) After setting up _____ for the I/O device, the device controller transfers an entire block of data directly to or from its own buffer storage to memory, with no intervention by the CPU.

- A) buffers
- B) pointers
- C) counters
- D) all of the above

Answer: D

2) Which of following is not true about asymmetric multiprocessing?

- A) Each processor performs all tasks within the operating system.
- B) Asymmetric multiprocessing defines a boss-worker relationship.
- C) Some processors have predefined tasks.
- D) A boss processor controls the system

Answer: A

3) Which of the following is NOT true about loadable kernel modules?

- A) Kernel provides core services while other services are implemented dynamically, as the kernel is running.
- B) Each kernel section has defined, protected interfaces, but a single module cannot call another module.
- C) The primary module has only core functions and knowledge of how to load and communicate with other modules.
- D) Modules do not need to invoke message passing in order to communicate.

Answer: B

4) Which one in the following uses interprocess communication (IPC) system?

- A) POSIX Shared Memory
- B) Mach
- C) Windows
- D) All of the above

Answer: D

5) Which is include in PCB when context switch occurs?

- A) the value of CPU registers
- B) the process state
- C) memory-management information
- D) all of the above

Answer: D

6) Which is the method for passing parameters to the OS?

- A) passing in registers
- B) OS gets parameters' address from block or table in memory
- C) Parameters are placed, or pushed, onto the stack by the program, and popped off the stack by the OS.
- D) All of the above

Answer: D

7) A zombie process is ____.

- A) a process that has terminated, but whose parent has not yet called wait()
- B) a process whose parent terminates without first calling wait()
- C) a process that periodically calls wait(), which allows any resources allocated to terminated processes to be reclaimed by the operating system
- D) a process that terminates the execution of its children processes

Answer: A

8) Two important design issues for cache memory are ____.

- A) speed and volatility
- B) size and replacement policy
- C) power consumption and reusability
- D) size and access privileges

Answer: B

9) ____ provide(s) an interface to the services provided by an operating system.

- A) Shared memory
- B) System calls
- C) Simulators
- D) Communication

Answer: B

10) When a child process is created, which of the following is a possibility in terms of the execution or address space of the child process?

- A) The child process runs concurrently with the parent.
- B) The child process has a new program loaded into it.
- C) The child is a duplicate of the parent.
- D) All of the above

Answer: D

2. (25 points) Simple questions on fork().

- 1) Consider the following code segments, what is the total number of processes (including the initial process)? Please elaborate. (10 points)

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>

int main() {
    if (fork() == 0) {
        fork();
        fork();
    }
    else
        fork();
    return 0;
}
```

Answer: There are total 6 processes. The parent process is A. The first fork() creates 1 child process A. The forth fork() creates 1 child process B. The second fork() creates 1 child process C. The third fork() creates 2 child processes D and E. And there is one parent process, totally 6 processes.

Consider the following code segments, what is the output of this segment? Please elaborate (10 points)

```
int var1 = -5;
int main()
{
    int var2=5;

    if (fork() == 0) {
        var1++;
    }
```

```

        var2++;
        printf(" var1 = %d, var2 = %d\n", var1, var2);
    }
    else {
        var1++;
        var2--;
        printf("var1 = %d, var2 = %d\n", var1, var2);
        wait(NULL);
    }
    return 0;
}

```

Answer:

var1 = -4, var2 = 4

var1 = -4, var2 = 6

Line 1: The child process consists of a copy of the address space of the parent. After the parent process forks the child process, the values of the variables "var1" and "var2" being exchanged, it has "var1=-4" and "var2=4". Then it waits for the child process. The child process's copy of "var1" and "var2" are also been exchanged, it has "var1=-4" and "var2=6".

- 2) Both of the parent process and the child process continue execution at the instruction after the fork(). What's the different value between the parent and the child. (5 points)

Answer: The return code for the fork() is zero for the new(child) process, whereas the (nonzero) process identifier of the child is returned to the parent.

3. (30 points) Please answer the following questions in 3-4 sentences.

- 1) (5 points) Which of the following instructions should be privileged?
 - a. Set value of timer.
 - b. Read the clock.
 - c. Clear memory.
 - d. Issue a trap instruction.
 - e. Turn off interrupts.

f. Modify entries in device-status table.

g. Switch from user to kernel mode.

h. Access I/O device.

Answer:

The following operations need to be privileged: Set value of timer, clear memory, turn off interrupts, modify entries in device-status table, access I/O device. The rest can be performed in user mode.

- 2) (5 points) Distinguish between the client – server and peer-to-peer models of distributed systems.

Answer:

The client-server model firmly distinguishes the roles of the client and server. Under this model, the client initiates and requests services that are provided by the server, which has to be available at all time. The peer-to-peer model does not have such strict roles. In fact, all nodes in the system are considered peers and thus may act as either clients or servers – or both. A node may request a service from another peer, or the node may in fact provide such a service to other peers in the system.

- 3) (5 points) What are the three major activities of an operating system with regard to memory management?

Answer:

The three major activities are:

- a. Keep track of which parts of memory are currently being used and by whom.
 - b. Decide which processes are to be loaded into memory when memory space becomes available.
 - c. Allocate and deallocate memory space as needed.
- 4) (5 points) What is the main advantage of the layered approach to system design? What are the disadvantages of using the layered approach?

Answer:

As in all cases of modular design, designing an operating system in a modular way has several advantages. The system is easier to debug and modify because changes affect only limited sections of the system rather than touching all sections

of the operating system. Information is kept only where it is needed and is accessible only within a defined and restricted area, so any bugs affecting that data must be limited to a specific module or layer.

- 5) (5 points) Why does Java provide the ability to call from a Java program native methods that are written in, say, C or C++? Provide an example of a situation in which a native method is useful.

Answer: Java programs are intended to be platform I/O independent. Therefore, the language does not provide access to most specific system resources such as reading from I/O devices or ports. To perform a system I/O specific operation, you must write it in a language that supports such features (such as C or C++.) Keep in mind that a Java program that calls a native method written in another language will no longer be architecture-neutral.

4. (30 points) Process.

- 1) (6 points) When a process creates a new process using the fork() operation, which of the following state is shared between the parent process and the child process?
- a. Stack
 - b. Heap
 - c. Shared memory segments

Answer:

Only the shared memory segments are shared between the parent process and the newly forked child process. Copies of the stack and the heap are made for the newly created process.

- 2) (6 points) Please briefly explain the five process states, and give a diagram to illustrate the transition between them.

Answer:

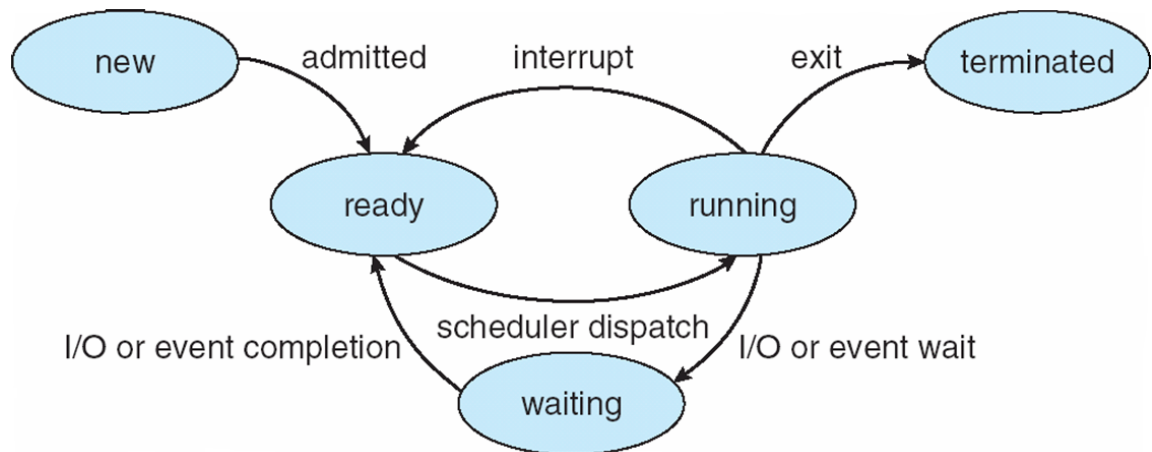
New: The process is being created—resources being allocated, such as memory

Running: Instructions are being executed

Waiting: The process is waiting for some event to occur (such as I/O completion)

Ready: The process is waiting to be assigned to a processor (CPU)

Terminated: The process has finished execution—to release resources



- 3) (6 points) Please describe the key actions taken by the kernel during context switch.

Answer: In general, the operating system must save the state of the currently running process to PCB and restore the state of the process scheduled to be run from PCB next. Saving the state of a process typically includes the values of all the CPU registers in addition to memory allocation. Context switches must also perform many architecture-specific operations, including flushing data and instruction caches.

- 4) (6 points) Describe the differences among short-term, medium-term, and long-term scheduling.

Answer:

- Short-term (CPU scheduler) – selects from jobs in memory those jobs that are ready to execute and allocates the CPU to them.

- Medium-term – used especially with time-sharing systems as an intermediate scheduling level. A swapping scheme is implemented to remove partially run programs from memory and reinstate them later to continue where they left off.

- Long-term (job scheduler) – determines which jobs are brought into memory for processing.

The primary difference is in the frequency of their execution. The short-term must select a new process quite often. Long-term is used much less often since it handles placing jobs in the system and may wait a while for a job to finish before it admits another one.

- 5) (6 points) Consider the situation where a process reads a log and many processes writes to the same log. Is it appropriate to use an ordinary pipe as interprocess communication mechanism? If YES, give the reason. If NO, give the reason and a suitable communication mechanism.

Answer: NO. Ordinary pipe is unidirectional. It only allows communication in a standard producer-consumer fashion-pipe. Those writing processes cannot communicate via the ordinary pipe. Named pipe is suitable. When writing processes wish to write a message, they write it to the named pipe. When the reading process wants to read a message, it reads from the named pipe.