Fall 2017 COMP 3511 Homework Assignment #1
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 #1**. 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

    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

    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 pf how to load and communicate with other modules.
    D) Modules do not need to invoke message passing in order to communicate.

    4) Which one in the following uses inter-process communication (IPC) system?
    A) POSIX Shared Memory
    B) Mach
    C) Windows

D) All of the above

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

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

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

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

9) _____ provide(s) an interface to the services provided by an operating system.
  A) Shared memory
  B) System calls
  C) Simulators
  D) Communication

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

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)

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

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

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

```c
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;
}
```

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)

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.

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

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

4) (5 points) What is the main advantage of the layered approach to system design? What are the disadvantages of using the layered approach?

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.

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

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

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

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

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 inter-process communication mechanism? If YES, give the reason. If NO, give the reason and a suitable communication mechanism.