

# Spring 2015 COMP 3511 Homework Assignment #1 Solution

Handout Date: Feb. 13, 2015 Due Date: Feb. 27, 2015

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 each page.
- Please fill in your answers in the space provided, or you can type your answers in the MS Word file.
- **Homework Collection: the hardcopy** is required and the homework is collected in **collection box #16**. The collection boxes locate outside Room 4210, near lift 21 (there are labels attached on the boxes).

1. (20 points) Multiple choices

- 1) A(n) \_\_\_\_\_ is used to prevent a user program from getting stuck in an infinite loop or never returning control to the OS.

- A) Program counter
- B) Timer
- C) Interrupt
- D) CPU scheduler

**Answer: B**

- 2) Which of the following statements is TRUE?

- A) In an SMP-type system, there is one master CPU and the remainder CPUs are slaves.
- B) A Web browser is a system program.
- C) A trap can be used to call operating system routines or to catch arithmetic errors.
- D) The two separate modes of operating in a system are supervisor mode and system mode.

**Answer: C**

- 3) Which of the following statements is NOT TRUE about DMA (Direct Memory Access)?

- A) DMA is used for high-speed I/O devices in order to avoid increasing the CPU's execution load.

- B) CPU initiates a DMA controller, which instructs a device controller to move data between the devices and main memory.
- C) After the completion of the DMA transfer, the CPU is notified by interrupt signal.
- D) The CPU is not allowed to execute other programs while the DMA controller is transferring data.

**Answer: D**

4) Which of the following should NOT be part of a microkernel?

- A) File system service
- B) Inter-process communication
- C) CPU scheduling
- D) Address space management

**Answer: A**

5) Which of the following is NOT contained in a Process Control Block (PCB)?

- A) The process ID
- B) The process state
- C) The scheduler ready queue
- D) The location of the process in memory

**Answer: C**

6) Which of the following methods is used for passing parameters to the operating system?

- A) Registers pass starting addresses of blocks of parameters.
- B) Pass parameters in registers.
- C) Parameters can be placed or pushed onto the stack by the program, and popped off the stack by the operating system.
- D) All of the above.

**Answer: D**

7) Long-term scheduler (or job scheduler) \_\_\_\_\_.

- A) selects which processes should be executed next and allocate CPU
- B) is invoked very frequently, in milliseconds
- C) is sometimes the only scheduler in a system
- D) controls the degree of multiprogramming

**Answer: D**

8) 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**

9) What are the two basic models of inter-process communication?

- A) Message-passing model and shared-memory model.
- B) Direct communication model and indirect communication model.
- C) User model and system model.
- D) Physical model and logical model.

**Answer: A**

10) Which of the following statements is NOT true about pipes?

- A) Name pipes do not require parent-child relationships.
- B) An ordinary pipe can be accessed from outside the process that created the pipe.
- C) Name pipes allow multiple processes to use it for communications and multiple processes can write to it.
- D) Ordinary pipes allow two processes to communicate in a standard producer and consumer fashion.

**Answer: B**

2. (20 points) Simple C programs 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() {
    pid_t pid;
    pid = fork();
    if (pid == 0) {
        fork();
        fork();
    }
}
```

**Answer:** There are total 5 processes. The first fork() creates one child (1), the second fork() creates one child (2), and the third fork() run by two processes, the first child process (1) and second child process (2), creates two more child (3 and 4).

- 2) 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() {
    for (i = 0; i < 10; i++)
        if (fork() > 0)
            fork();
}
```

**Answer:** There are total  $3^{10}$  processes. In each loop, there are 2 fork() operations. The first fork() creates one child (1). The second fork() is run by the original process(0), and thus creates the second child process (2). So after the first loop there will be 3 processes in total. After the second loop, each of these 3 processes will have 2 more child processes. Thus after the second loop, there will be  $3^2$  processes. Thus in the end, after the 10 loops, there will be  $3^{10}$  processes in total.

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

- 1) (6 points) Describe the differences between symmetric and asymmetric multiprocessing. What are the advantage and disadvantage in multiprocessor systems?

**Answer:** Symmetric multiprocessing treats all processors as equals, and I/O can be processed on any CPU. Asymmetric multiprocessing has one master CPU and the remainder CPUs are slaves. The master distributes tasks among the slaves, and I/O is usually done by the master only.

Multiprocessors can save money by not duplicating power supplies, housings, and peripherals. They can execute programs more quickly and can have

increased reliability. They are also more complex in both hardware and software than uniprocessor systems.

(Advantages: Increased throughput, economy of scale, increased reliability. Disadvantage: complex in hardware and software.)

- 2) (6 points) What is the purpose of interrupts? What are the differences between a trap and an interrupt? Can traps be generated intentionally by a user program? If so, for what purpose?

**Answer:** An interrupt is a hardware-generated change of flow within the system. An interrupt handler is summoned to deal with the cause of the interrupt; control is then returned to the interrupted context and instruction. A trap is a software-generated interrupt. An interrupt can be used to signal the completion of an I/O to obviate the need for device polling. A trap can be used to call operating system routines or to catch arithmetic errors.

- 3) (6 points) What are the memory components in memory hierarchy? Briefly describes property of each memory components (e.g. size, volatility, implementation technology).

**Answer:** register, cache, main memory, disk storage

register: < 1KB, volatile, CMOS

cache: >16MB, volatile, SRAM

memory: >16GB, volatile, DRAM disk

storage: >100GB, non-volatile, magnetic disk

- 4) (6 points) Describe the relationship between an API, the system-call interface, and the operating system.

**Answer:** The system-call interface of a programming language serves as a link to system calls made available by the operating system. This interface intercepts function calls in the API and invokes the necessary system call within the operating system. Thus, most of the details of the operating-system interface are hidden from the programmer by the API and are managed by the run-time support library.

- 5) (6 points) What are the advantages of using loadable kernel modules?

**Answer:** It is difficult to predict what features an operating system will need when it is being designed. The advantage of using loadable kernel modules is that functionality can be added to and removed from the kernel while it is running. There is no need to either recompile or reboot the kernel.

4. (30 points) Process

- 1) (6 points) List all possible states of a process. What are the 4 main step in creating a process?

**Answer:**

Possible states: new, running, waiting, ready, terminated.

Creating process: construct a new PCB, set up new pages table for address space or simply set up memory space, copy data from parent process, copy I/O status.

- 2) (6 points) Describe the actions taken by a kernel to context-switch between processes.

**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.

- 3) (6 points) What do we mean by concurrency in a uniprocessor system? How is that different from a parallel system with multiple processors?

**Answer:** A concurrent system supports more than one task by allowing multiple tasks to make progress. Or within any period of time, there are usually multiple processes running on the CPU at different time instant. Or there are multiple processes within the system (or memory) can run interleaved. This is different from a parallel system in which system can perform more than one task simultaneously on different CPUs, while in a concurrent uniprocessor system, at most one process can run at one time.

- 4) (6 points) What is an orphan process? How does UNIX handle that?

**Answer:** A parent terminates without first calling wait(), its children are considered as orphan processes. UNIX assigns the init process as the new parent

of orphan processes and init periodically calls wait(), which allows any resources allocated to terminated processes to be reclaimed by the operating system.

- 5) (6 points) What is the primary distinction between short-term scheduler and long-term scheduler? Please briefly explain the reason.

**Answer:** The primary distinction between short-term scheduler and long-term scheduler is the frequency of their execution. The short-term scheduler must select a new process quite often, while long-term scheduler is used much less often since it handles placing jobs in the system and may wait a while for jobs to finish before it admits another one.