

# Spring 2019 COMP 3511 Homework Assignment

Handout Date: Feb. 16, 2019 Due Date: Mar. 4, 2019

Name: \_\_\_\_\_ ID: \_\_\_\_\_ E-Mail: \_\_\_\_\_

Please read the following instructions carefully before answering the questions:

- You should finish the homework assignment **individually**.
- There are total **5** 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.
- **Homework Submission**: the homework is submitted to **assignment #1** on CASS

1. [20 points] Multiple choices.

- 1) The main function of the command interpreter is
  - A) to get and execute the next user-specified command
  - B) to provide the interface between the API and application program
  - C) to handle the files in operating system
  - D) none of the mentioned
- 2) Embedded systems typically run on a \_\_\_\_ operating system.
  - A) real-time
  - B) Windows XP
  - C) network
  - D) clustered
- 3) 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
- 4) DMA is used for:
  - A) High speed devices (disks and communications network)
  - B) Low speed devices
  - C) Utilizing CPU cycles
  - D) All of the mentioned
- 5) An interrupt vector
  - A) is an address that is indexed to an interrupt handler
  - B) is a unique device number that is indexed by an address
  - C) is a unique identity given to an interrupt
  - D) none of the mentioned
- 6) Two important design issues for the cache memory are \_\_\_\_.
  - A) speed and volatility
  - B) size and replacement policy

- C) power consumption and reusability
  - D) size and access privileges
- 7) What is a long-term scheduler supposed to do?
    - A) It selects which process has to be brought into the ready queue
    - B) It selects which process has to be executed next and allocates CPU
    - C) It selects which process to remove from memory by swapping
    - D) None of the mentioned
  - 8) The child process can :
    - A) be a duplicate of the parent process
    - B) never be a duplicate of the parent process
    - C) cannot have another program loaded into it
    - D) never have another program loaded into it
  - 9) 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.
  - 10) In indirect communication between processes P and Q :
    - A) there is another process R to handle and pass on messages between P and Q
    - B) there is another machine between the two processes to help communication
    - C) there is a mailbox to help communication between P and Q
    - D) none of the mentioned
2. [10 points] The program, `process-run.py`, allows you to see how process states change as programs run and either use the CPU (e.g., perform an add instruction) or do I/O (e.g., send a request to a disk and wait for it to complete). See the README for details. Run `python process-run.py -l 5:100,5:100` and `python process-run.py -l 5:100,5:0`. What should the CPU utilizations be (e.g., the percent of time the CPU is in use)? What are the differences between the two runs? Please justify your answer. Attach screen captures of the two runs. Hint: Use the `-c` and `-p` flags.
  3. [10 points] Write a short program using `fork()`. The child process should print "hello"; the parent process should print "goodbye". You should try to ensure that the child process always prints first; can you do this without calling `wait()` in the parent? Attach your code. Hint: Do anything to delay the parent, e.g., looping, `sleep()`, or relinquish CPU temporarily
  4. [12 points] On `fork()`

- 1) Consider the following code segments, how many “forked\n” will be printed?  
Please elaborate. (6 points)

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
int main() {

    fork();
    fork() && fork();

    printf("forked\n");
    return 0;
}
```

- 2) Consider the following code segments, how many different copies of the variable c are there? What are their values, respectively? (6 points)

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
int main() {
    int child = fork();
    int c = 5;
    if(child == 0)
    {
        c += 5;
    }
    else
    {
        child = fork();
        c += 10;
        if(child)
            c += 5;
    }
    printf("%d",c);
    return 0;
}
```

5. [48 points] Please answer the following questions in a few sentences.
- 1) What are the main reasons for separating kernel mode and user mode in operating system? Can you name the ways that the CPU mode changes from user model to kernel mode? (6 points)
  - 2) What do we mean by the temporal locality and spatial locality in a caching system? How do they affect the average access time (Hint: the average access time = hit rate x cache access time + (1 - hit rate) x memory access time) (6 points)
  - 3) Commercial operating system usually adopts a hybrid approach in the design. Can you illustrate such an approach using Apple Mac OS X? (6 points)
  - 4) What does linker do? What does loader do? (6 points)
  - 5) Please compare and contrast `fork()` in Unix and `CreateProcess()` in Microsoft Window operating system (6 points)
  - 6) What are the distinctive features in a client-server communication? What are the four pieces of information (4-tuples) to determine a socket communication? (6 points)
  - 7) What do we refer as an orphan process? How does Unix operating system handle orphan processes? (6 points)
  - 8) Please briefly explain how system calls are used in the following diagram including `fork()`, `exec()`, `wait()` and `exit()`. (6 points)

