OPERATING SYSTEMS (COMP 310 / ECSE 427) – Midterm – February 27, 2014

Name: Solutions **Student ID:**

Part I –Some multiple-choice questions can have **more than one correct** answer. Circle *all and only correct* answers. (Answer **all 10 questions – 6 points each**)

1. What are two important differences between a system call and a function call?

1. System call runs code part of OS

2. System call runs in privileged mode

2. How does address space virtualization (or memory virtualization) help multi-processing?
   a. Allows the different processes to load their segments to the same virtual address locations
   b. Isolates the address space between two processes that are resident in the memory at the same time
   c. Allows the CPU to access the memory faster
   d. Resolves memory contention in a fair manner
   e. None of the above

3. Consider a process that is being context switched out of the CPU (i.e., the process is getting evicted from the CPU and will be scheduled in a later point in time). Which of the following should be always true for the process.
   a. The process will never run on the CPU again
   b. The process is waiting on a semaphore
   c. The process is waiting for a disk operation to complete
   d. The process has been using too much CPU time
   e. None of the above.

A little confusing...

} One possible answer

Another choice

4. Two processes are created by an application. Which of the segments associated with the processes could be same between the two processes?

The text segment could be the same.

5. A process has multiple threads. The threads are implemented at the kernel level. Which of the following statements are true?
   a. The text segment is common across all the threads
   b. The data segment is common across all the threads
   c. All threads share the same stack for all their activities such as function invocation.
   d. The memory map segment is different for the different threads

goes on per-thread Stack

6. A process and its child process are writing their outputs to a file. The output from the parent and child processes do not clobber each other – that is, do not overwrite each other. Which of the following is true regarding the implementation of this feature?
   a. The parent and child processes have the same file descriptor table
   b. The parent and child processes are synchronized so that one writes after the other
   c. The file system table is common to both processes
   d. The parent and child processes have their own entries in the file system table corresponding to the open file each have for outputting data
   e. None of the above
7. A program takes 200 seconds to run. An enhancement is applied to a portion of the program to make it run faster. Only 30% of the program as measured in terms of run time cannot be enhanced by this modification. The modification can improve the run time by a factor of 10. What is the overall speedup?

Enhancement applied to 140s; New runtime = $\frac{140}{10} + 60$

Speedup = $\frac{200}{74} = 2.7$

8. Consider two ways of implementing threads: user-level versus kernel-level. Which of the following statements are true?
   a. With user-level threads blocking system calls can stop the whole application – that is no part of the application is running
   b. With kernel-level threads system calls take longer to execute
   c. With user-level threads the execution time of the application can be shorter because the context switching overhead is small
   d. With kernel-level threads more CPU time is actually used by the application and it could run faster
   e. None of the above
9. Provide the Peterson's algorithm for synchronizing two processes.

Process 0

```
turn = 1
flag[0] = true
while (turn == 1 &&
       flag[1]);
<.....>
flag[0] = false
```

Process 1

```
turn = 0
flag[1] = true
while (turn == 0 &&
       flag[0]);
<.....>
flag[1] = false
```

10. What is priority inversion? Briefly explain.

A low priority process is able to prevent a high priority from executing by holding a lock that is needed by the high priority process. Low priority process is unable to release the lock because the CPU is held by the high priority process busy waiting.

Part II – Long form Questions (Answer **both questions – provide the shortest possible answer**)

1. **(20 points)** Consider the readers and writers problem. We have many readers and writers and readers have priority. A writer needs to wait until all readers have completed before it could start. A reader need not wait for all writers. All readers can access the data object concurrently. Writers have to access the object serially – one after the other. Provide a monitor that provides the following update functions.

```
reader () {
    while(true) {
        // Do something
        get_reader_lock()
        // Do reading
        release_reader_lock()
        // Do something
    }
}
```

```
writer () {
    while(true) {
        // Do something
        get_writer_lock()
        // Do writing
        release_writer_lock()
        // Do something
    }
}
```

```
Monitor RWlock {
condvar RWQueue
bool reading = false
bool writing = false
int readerCnt = 0

get_reader_lock() {
    readerCnt ++
    if (reading)
        return;
    else if (writing)
        RWQueue.wait();
    reading = true
    return;
}

release_reader_lock() {
    readerCnt--
    if (readerCnt == 0)
        reading = false
        RWQueue.signal()
}
```

```
get_writer_lock() {
    if (reading ||
        writing)
    RWQueue.wait();
    writing = true;
}

release_writer_lock() {
    writing = false
    RWQueue.signal();
}
```

changed to [4 3] in the exam

Give full credit students who did with old value...

2. **(20 points)** We have four processes and two different types of resources in a computing system. The maximum claim matrix (C) and current allocation (H) are given below. The availability vector is [3, 2]. Using Banker's algorithm find:
   a. A value for x and a value for y for which the system is safe.
   b. A value for x and a value for y for which the system is unsafe.

Note: there is no unique answer.

$$C = \begin{bmatrix} 8 & 9 \\ x & 6 \\ 7 & 5 \\ 6 & 4 \end{bmatrix}, \quad H = \begin{bmatrix} 3 & y \\ 1 & 2 \\ 2 & 2 \\ 2 & 1 \end{bmatrix}$$

$$\text{Need} = \begin{bmatrix} 8 & 9 \\ x & 6 \\ 7 & 5 \\ 6 & 4 \end{bmatrix} - \begin{bmatrix} 3 & y \\ 1 & 2 \\ 2 & 2 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 5 & 9-y \\ x-1 & 4 \\ 5 & 3 \\ 4 & 3 \end{bmatrix} \begin{matrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{matrix}$$

Avail [4 3]

P_3 can be completed.

We have [6 4]

P_2 can be completed

We have [8 6]

There are other possible answers as well

$x = 15$ (safe no matter value of y)

$x = 9$ (safe no matter the value of y)

Unsafe $x = 15$, $y = 5$

Safe $x = 9$ $y = 7$