Problem Set 3: CPU Scheduling and Synchronization[1]

Answers to all questions should be in one file named **ps3.pdf.** You may use Microsoft Word or any other editor, and then convert the document to PDF.

1. Consider the following set of processes, with the length of the CPU burst given in miliseconds:

Process	Burst Time	Priority
P1	2	2
P2	1	1
P3	8	4
P4	4	2
P5	5	3

The process are assumed to have arrived in the order P1 through P5 all at time 0.

   a. Draw four GANTT charts that illustrate the execution of these processes using the following scheduling algorithms:
      1. FCFS
      2. SJF
      3. Nonpreemptive priority (a larger priority number implies a higher priority)
      4. RR (quantum = 2)
   b. What is the turnaround time for each process for each of the scheduling algorithms in part a?
   c. What is the waiting time for each process for each of the scheduling algorithms?
   d. Which of the algorithms results in the minimum average waiting time (over all processes)?

2. Consider a system running ten I/O bound tasks and one CPU-bound task. Assume that the I/O-bound tasks issue an I/O operation once for each millisecond of CPU computing (1ms CPU burst), that each I/O operation takes 10 milliseconds to complete, and all I/O operations are blocking (release CPU) and independent (for example, using different devices). Assume the CPU task does no I/O and uses all CPU time it is given. Also assume that the context-switching overhead is 0.1 milliseconds and that all process are long-running tasks. Describe the CPU utilization for a RR scheduler when:
   a. The time quantum is 1 milliseconds.
   b. The time quantum is 10 milliseconds.

3. For each of the following scheduling algorithms, explain how it favours short processes (if at all):
   a. FCFS
   b. RR
   c. Multilevel feedback queues

[1] Created by Mohamed Hefeeda, modified by Brian Fraser. Some parts provided by Prof. Wei Tsang Ooi of National University of Singapore. Some questions are from Operating Systems Concepts (OSC) by Silberschatz, Galvin and Gagne.

4. Describe the circumstances under which one should use each of the following synchronization tools. You may want to consider how long a lock will be held, if it is a multiprocessor system, and any other factors which seem relevant.
   a. spinlocks
   b. mutex locks
   c. semaphores
   d. condition variables
   e. reader-writer locks
5. On a single processor system, answer each of the following questions about disabling interrupts:
   a. Describe how disabling interrupts can be used to ensure mutual exclusion.
   b. Discuss the problems of allowing programs to disable interrupts in user mode without executing a sys-call.
6. Consider the code shown below which can be used for allocating resource numbers. For example, it could be used in the kernel to allocate process numbers.

```
1. #define MAX_RESOURCES 5
2. int resources_allocated = 0;
3.
4. // Allocate a resource to the calling code.
5. // Return true if successful, false if unable to allocate.
6. _Bool allocate_resource(void)
7. {
8.     if (resources_allocated >= MAX_RESOURCES) {
9.         return false;
10.        } else {
11.                resources_allocated++;
12.                return true;
13.        }
14.    }
15.
16.    // Release the resource
17.    void release_resource(void)
18.    {
19.        resources_allocated--;
20.    }
```

   a. Identify the race condition(s) by describing which piece(s) of code, when running concurrently, can trigger the race condition.
   b. Assume you have a mutex lock named `mutex` with operations acquire() and release(). Indicate where you would need to lock and unlock the mutex. In your answer, you may indicate "between lines 3 and 4" by saying "at line 3.5 add ..."
   c. Atomic variables are special variable types that ensure that increment, decrement and addition operations are performed atomically. Could the variable at line 2

```
int resources_allocated = 0;
```

   be replaced with an atomic integer variable

```
atomic_t resources_allocated = 0;
```

   to correctly prevent all race conditions in the code?