Chapter 3 exercises

1. A multithreaded web server wishes to keep track of the number of requests it services (known as hits). Consider the two following strategies to prevent a race condition on the variable hits . The first strategy is to use a basic mutex lock when updating hits :

```
int hits;
mutex_lock hit_lock;
hit_lock.acquire();
hits++;
hit_lock.release();
```

A second strategy is to use an atomic integer:

```
atomic_t hits;
atomic_inc(&hits);
```

Explain which of these two strategies is more efficient.

2. Consider the code example for allocating and releasing processes shown in the following figure.

```
#define MAX_PROCESSES 255
int number_of_processes = 0;
/* the implementation of fork() calls this function */
int allocate process() {
  int new_pid;
  if (number_of_processes == MAX PROCESSES)
    return -1;
  else {
    /* allocate necessary process resources */
    ++number_of_processes;
  }
}
return new_pid;
/* the implementation of exit() calls this function */
void release_process() {
  /* release process resources */
  --number_of_processes;
}
```

a. Identify the race condition(s).
b. Assume you have a mutex lock named mutex with the operations acquire() and release() . Indicate where the locking needs to be placed to prevent the race condition(s).
c. Could we replace the integer variable

```
int number of processes = 0
```

with the atomic integer

```
Atomic_t number_of_processes = 0
```

to prevent the race condition(s)?

3. Servers can be designed to limit the number of open connections. For example, a server may wish to have only N socket connections at any point in time. As soon as N connections are made, the server will not accept another incoming connection until an existing connection is released. Explain how semaphores can be used by a server to limit the number of concurrent connections.

4. Show how to implement the wait() and signal() semaphore operations in multiprocessor environments using the test and set() instruction. The solution should exhibit minimal busy waiting.

5. Explain the difference between preemptive and nonpreemptive scheduling.

6. Suppose that the following processes arrive for execution at the times indicated. Each process will run for the amount of time listed. In answering the questions, use nonpreemptive scheduling, and base all decisions on the information you have at the time the decision must be made.

Process	Arrival Time	Burst Time
P 1	0.0	8
P 2	0.4	4
P 3	1.0	1

a. What is the average turnaround time for these processes with the FCFS scheduling algorithm?
b. What is the average turnaround time for these processes with the SJF scheduling algorithm?
c. The SJF algorithm is supposed to improve performance, but notice that we chose to run process P 1 at time 0 because we did not know that two shorter processes would arrive soon. Compute what the average turnaround time will be if the CPU is left idle for the first 1 unit and then SJF scheduling is used. Remember that processes P 1 and P 2 are waiting during this idle time, so their waiting time may increase. This algorithm could be called future-knowledge scheduling.

7. What advantage is there in having different time-quantum sizes at different levels of a multilevel queueing system?

8. Suppose that a scheduling algorithm (at the level of short-term CPU scheduling) favors those processes that have used the least processor time in the recent past. Why will this algorithm favor I/O-bound programs and yet not permanently starve CPU-bound programs?

9. Why is it important for the scheduler to distinguish I/O -bound programs from CPU -bound programs?

10. Discuss how the following pairs of scheduling criteria conflict in certain settings.
a. CPU utilization and response time
b. Average turnaround time and maximum waiting time
c. I/O device utilization and CPU utilization

11. Consider the following set of processes, with the length of the CPU burst given in milliseconds:

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

The processes are assumed to have arrived in the order P1 , P2 , P3 , P4 , P5 , all at time 0.
a. Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS , SJF , nonpreemptive priority (a larger priority number implies a higher priority), and RR (quantum = 2).
b. What is the turnaround time of each process for each of the scheduling algorithms in part a?
c. What is the waiting time of each process for each of these scheduling algorithms?
d. Which of the algorithms results in the minimum average waiting time (over all processes)?

12. The following processes are being scheduled using a preemptive, round-robin scheduling algorithm. Each process is assigned a numerical priority, with a higher number indicating a higher relative priority. In addition to the processes listed below, the system also has an idle task (which consumes no CPU resources and is identified as P idle). This task has priority 0 and is scheduled whenever the system has no other available processes to run. The length of a time quantum is 10 units. If a process is preempted by a higher-priority process, the preempted process is placed at

the end of the queue.

Thread	Priority	Burst	Arrival
P_1	40	20	0
P_2	30	25	25
P_3	30	25	30
P_4	35	15	60
P_5	5	10	100
P_6	10	10	105

a. Show the scheduling order of the processes using a Gantt chart.
b. What is the turnaround time for each process?
c. What is the waiting time for each process?
d. What is the CPU utilization rate?

13. Which of the following scheduling algorithms could result in starvation?
a. First-come, first-served
b. Shortest job first
c. Round robin
d. Priority

14. 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 every millisecond of CPU computing and that each I/O operation takes 10 milliseconds to complete. Also assume that the context-switching overhead is 0.1 millisecond and that all processes are long-running tasks. Describe the CPU utilization for a round-robin scheduler when:
a. The time quantum is 1 millisecond
b. The time quantum is 10 milliseconds

15. Explain the differences in how much the following scheduling algorithms discriminate in favor of short processes:
a. FCFS
b. RR
c. Multilevel feedback queues

16. Suppose that a system is in an unsafe state. Show that it is possible for the processes to complete their execution without entering a deadlocked state.

17. Consider the following snapshot of a system:

	Allocation	*Max*	*Available*
	A B C D	*A B C D*	*A B C D*
P_0	0 0 1 2	0 0 1 2	1 5 2 0
P_1	1 0 0 0	1 7 5 0	
P_2	1 3 5 4	2 3 5 6	
P_3	0 6 3 2	0 6 5 2	
P_4	0 0 1 4	0 6 5 6	

Answer the following questions using the banker's algorithm:
a. What is the content of the matrix Need?
b. Is the system in a safe state?
c. If a request from process P_1 arrives for (0,4,2,0), can the request be granted immediately?

18. Consider a system consisting of four resources of the same type that are shared by three processes, each of which needs at most two resources. Show that the system is deadlock free.

19. Consider a system consisting of m resources of the same type being shared by n processes. A process can request or release only one resource at a time. Show that the system is deadlock free if the following two conditions hold:
a. The maximum need of each process is between one resource and m resources.
b. The sum of all maximum needs is less than m + n.