CS365 Operating System Midterm Exam

Name: ______________________________

Question Number	Score
1	
2	
3	
4	
5	
6	
7	
Total:	

Question1:
Explain the following terminology: (1pt each)

1. **System Call**
2. **Process**
3. **I/O-bound Process**
4. **Context Switch**
5. **LWP**
6. **Symmetric Multiprocessing**

1. **System Call:** is how a program requests service from an operating system's kernel
2. **Process:** a program in execution.
3. **I/O-bound process:** spends more time doing I/O than computations, many short CPU bursts.
4. **Context Switch:** is the process of storing and restoring the state (the context) of a process or thread so that execution can be resumed from the same point at a later time.
5. **LWP:** is an intermediate data structure between user and kernel threads, which appears to be a virtual processor on which process can schedule user thread to run on a kernel thread.
6. **Symmetric Multiprocessing:** In the symmetric multiprocessor system with two or more identical processors, each processor is self-scheduling and each processor may have its own ready queue.

Question2:

2.1 Explain and describe 3 different types of process schedulers. (3 pts)

- **Long-term scheduler**

(or job scheduler) –selects which processes should be brought into the ready queue

- **Short-term scheduler**

(or CPU scheduler) –selects which process should be executed next and allocates CPU

- **Midterm Scheduler**

Sometimes it can be advantage to remove process from memory and thus decrease the degree of multiprogrammimg. This scheme is called **swapping**

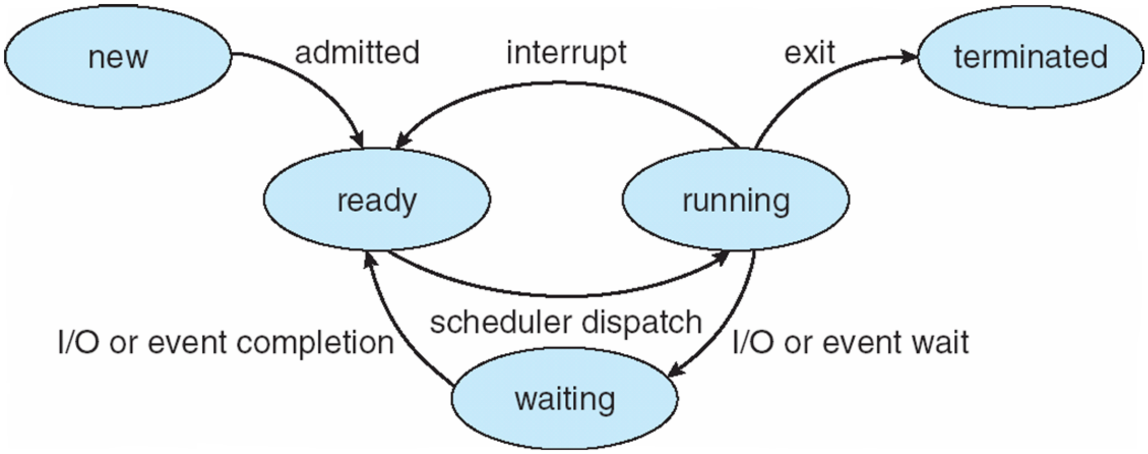
2.2 Describe 3 different multithreading models for mapping user threads to kernel threads. Also, describe at least **one** advantage or disadvantage for each model. (5 pts)

Many to one: It maps many user-level threads to one kernel thread. The entire system may block makes a block system call.

One to one: Each user-level thread has one corresponding kernel thread. The only drawback is that creating a user thread requires creating the corresponding kernel thread

Many to many: Multiplexes many user-level threads to a smaller or equal number of kernel threads. User can create as many threads as they want

2.3 Draw the state diagram of a process from its creation to termination, including all transitions, and briefly elaborate every state and every transition . (5 pts)



Question3:

According to the bounded buffer code in shared memory system we discussed in the class, draw the ring structure with size 4 and indicate the location of “in” and “out” for each time space. (If the buffer is full or empty, indicates how many jobs are waiting) (10pts, 2 pts for each)

Producer View

```
while (true) {
   /* produce an item and put in next Produced*/
   while (count == BUFFER_SIZE)
        ; // do nothing

   buffer [in] = nextProduced;
   in = (in + 1) % BUFFER_SIZE;
   count++;
}
```

Consumer View

```
while (true) {
   while (count == 0)
        ; // do nothing

   nextConsumed= buffer[out];
   out = (out + 1) % BUFFER_SIZE;
   count--;
   /* consume the item in next Consumed
}
```

Time	Action	waiting:0	count:0	in:1	out:1
1	Producer produces 2 jobs	0	2	3	1
2	Consumer consumes 1 job	0	1	3	2
3	Producer produces 3 jobs	0	4	2	2
4	Consumer consumes 3 jobs	0	1	2	1
5	Consumer consumes 3 jobs	1	0	2	2

Question4:

By “**Multilevel Queue Fixed priority**” scheduling algorithm, draw the CPU scheduling Gantt chart (5 pts) and complete the table for the given processes information. (10 pts)

	Process	Burst time	Arriving time	Algorithm
■ 1st Foreground	P1	50	0.0	RR interval:20
	P2	15	30.0	(RR is a non-preemptive
	P3	45	30.0	algorithm)
■ 2nd Foreground	P4	40	0.0	SJF Preemptive
	P5	10	120.0	
■ Background	P6	30	60.0	FCFS
	P7	20	130.0	

Gantt Chart:

P1	P1	P2	P3	P1	P3	P3	P4	P5	P4	P6	P7

0 20 40 55 75 85 105 110 120 130 160 190 210

Scheduling criteria:

	P1	P2	P3	P4	P5	P6	P7
Waiting time	35	10	35	120	0	100	60
Turnaround time	85	25	80	160	10	130	80
Response time	0	10	25	110	0	100	60

Question5:

Circle the best answer. (14 pts, 1 pt for each)

AAACBDBBBDDCDB

1. The major difficulty in designing a layered operating system approach is ____.

[✓ A)] appropriately defining the various layers

B) making sure that each layer hides certain data structures, hardware, and operations from higher-level layers
C) debugging a particular layer
D) making sure each layer is easily converted to modules

2. A boot block ____.
[✓ A) typically only knows the location and length of the rest of the bootstrap program]
B) typically is sophisticated enough to load the operating system and begin its execution
C) is composed of multiple disk blocks
D) is composed of multiple disk cylinders

3. Microkernels use _____ for communication.
[✓ A) message passing]
B) shared memory
C) system calls
D) virtualization

4. The UNIX fork() system call creates a new process. What is the equivalent system call in WINDOWS:
A) NTCreateProcess()
B) process()
[✓ C) CreateProcess ()]
D) getpid()

5. All access to POSIX shared memory requires a system call.
A) True
[✓ B) False]

6. When a child process is created, which of the following is a possibility in terms of the execution or address space of the child process?
A) The child process runs concurrently with the parent.
B) The child process has a new program loaded into it.
C) The child is a duplicate of the parent.
[✓ D) All of the above]

7. Cancellation points are associated with ____ cancellation.
A) asynchronous
[✓ B) deferred]
C) synchronous
D) immediate

8. The _____ model allows a user-level thread to be bound to one kernel thread.
A) many-to-many
[✓ B) two-level]
C) one-to-one
D) many-to-one

9. Which of the following is true of cooperative scheduling?
A) It requires a timer.
[✓ B) A process keeps the CPU until it releases the CPU either by terminating or by switching to the waiting state.]
C) It incurs a cost associated with access to shared data.
D) A process switches from the running state to the ready state when an interrupt occurs.

10. ____ is the number of processes that are completed per time unit.
A) CPU utilization
B) Response time
C) Turnaround time
[✓ D) Throughput]

11. ____ scheduling is approximated by predicting the next CPU burst with an exponential average of the measured lengths of previous CPU bursts.
A) Multilevel queue
B) RR
C) FCFS
[✓ D) SJF]

12. The ____ scheduling algorithm is designed especially for time-sharing systems.
A) SJF
B) FCFS
[✓ C) RR]

13. Which of the following scheduling algorithms must be nonpreemptive?
A) SJF
B) RR
[✓ C) FCFS]
D) priority algorithms

14. Which of the following is true of multilevel queue scheduling?
A) Processes can move between queues.
[✓ B) Each queue has its own scheduling algorithm.]
C) A queue cannot have absolute priority over lower-priority queues.
D) It is the most general CPU-scheduling algorithm.