Midterm
April 16, 2004 15.15–17.15
Good Luck!

Part A. Multiple Choice, only 1 answer must be marked (each 1 pt)

1 You are designing a highly reliable system. This means

a you are trying to minimize hardware failures

b you are trying to minimize software failures

c you are trying to minimize downtime

d you are trying to maximize throughput

e all of the above

f a, b and c

2 Compared with a synchronous I/O request

a an asynchronous request is easier to program

b an asynchronous request is more efficient

c an asynchronous request requires kernel support for threads

d all of the above

e none of the above

f a and b

3 The UNIX fork system call

a implies that a new process is the same as its parent

b is never an efficient way to implement parallel code

c can differ from its parent by checking the result of calling fork

d both b and c

e both a and c

f both a and b

4 Implementation of a real-time operating system

a implies limits to the memory hierarchy to guarantee execution times

b works best with a specialized kernel

d all of the above

e both a and b

f both a and c

5 For a preemptive SJF scheduler

a accurate implementation is impossible

b starvation is possible

c average wait time vs. nonpreemptive SJF is usually longer

d none of the above

e both a and b

f both a and c

6 An atomic memory modification and test instruction

a is essential for implementing synchronization

b is not needed for implementing synchronization

c makes implementing synchronization easier

d a and b

e a and c

f b and c

7 Which of the following are more like policies, and which are more like mechanisms? For each answer, circle policy or mechanism.

a The timer interrupt Policy / Mechanism

b How long a time quantum should be Policy / Mechanism

c Saving the register state of a process Policy / Mechanism

d Continuing to run the current process when a disk I/O interrupt occurs Policy / Mechanism

8 For a workload consisting of ten CPU-bound jobs of all the same length (each would run for 10 seconds in a dedicated environment), which policy would result in the lowest average response time?

a Round-robin with a 100 millisecond quantum

b Shortest Job First

d Round-robin with a 1000 nanosecond quantum

Part B. Short Answer, answer the questions with one or two sentences (each 3 pt)

1 What are the two main purposes of an operating system?

2 What is multiprogramming?

3 Which of the following instructions (or instruction sequences) should only be allowed in kernel mode?

1. Disable all interrupts.
2. Read the time of day clock.
3. Set the time of day clock.
4. Change the memory map.
5. Write to the hard disk controller register.
6. Write all buffered blocks associated with a file back to disk (fsync).

4 On UNIX, which of the following are considered system calls? Why?

1. `read()`
2. `printf()`
3. `malloc()`
4. `open()`

.
.
.

5 What are the differences between a trap and an interrupt? What is the use of each function?

a How can you use this to implement a critical section?

b Why does it work?

c Why is this generally a bad idea?

7 Processes (or threads) can be in one of three states: Running, Ready, or Blocked. For each of the following three examples, write down which state the process (or thread) is in:

a Waiting in Domain Read() for a message from some other process to arrive.

b Spin-waiting for a variable x to become non-zero.

c Having just completed an I/O, waiting to get scheduled again on the CPU.

8 Regarding Job scheduling, what is the convoy effect?

9 What is deadlock?

(each 5 pt)

1 An operating system runs in privileged mode, a hardware state where it has full access to machine resources. Why is such a mode needed, and why can t normal user processes and threads enter privileged mode?

2 Why caches are useful? What problems do they solve? What problems do they cause? If a cache can be made as large as the device for which it is caching (for instance, a cache as large as a disk), why not make it that large and eliminate the device?

3 There are two different levels at which threads can be implemented. What are they and how do they differ? What advantages and disadvantages do each have?

4 What advantages do threads have over multiple processes? Suggest one application that would benefit from the use of threads?

5 What is a Critical Region? List and explain the four conditions that need to be satisfied to solve the critical-region problem?

6 What is meant by the term *context switch*? What might cause a context switch to occur?

ducer threads to pass data to consumer threads), but that the buffer is unbounded; in other words, it does not have a limit as to how big it can get.

a How many condition variables will you need in order to implement this buffer properly, and why?

b How is this different than a standard bounded buffer implementation?

8 Consider the following sets of processes, with the length of the CPU-burst time given in milliseconds. Arrival time is the time at which the process is added to the ready queue.

Process	**Burst Time**	**Arrival Time**
P1	10	0
P2	5	0
P3	3	0
P4	4	0
P5	2	4
P6	1	6

a Draw appropriate charts illustrating the execution of these processes using FCFS, SJF non-preemptive, and SJF preemptive.

b Calculate the wait times of each process for each strategy. Calculate the average wait times under each strategy.

9 Three jobs (A, B, and C) arrive to the job scheduler at time 0. Job A needs 10 seconds of CPU time, Job B needs 20 seconds, and Job C needs 30 seconds.

a What is the average turnaround time for the jobs, assuming a shortest-job-first (SJF) scheduling policy?

c Which finishes first, Job C in SJF or Job A in LJF?

10 Describe the round-robin scheduling algorithm. Discuss what issues need to be consider to achieve good performance from this algorithm.

11 Describe four ways to *prevent* deadlock by attacking the conditions required for deadlock (see the next question).

12 For deadlock to occur, four conditions must hold: mutual exclusion, hold and wait, no preemption, and circular wait. If any one condition does not hold, no deadlock can occur. Assume we want to allow preemption , and thus get out of deadlocks; in other words, if a deadlock is detected, we will forcibly take a lock away from a thread; by repeatedly doing this, we will eventually undo the deadlock. What new problems are introduced by this preemptive approach?

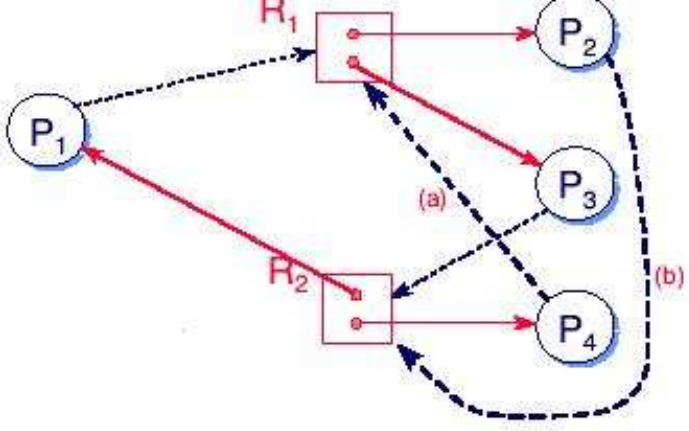

Figure 1: Figure

Part D.

1 The state of the system is with four processes and two multiple units resources described in resource allocation graph (see the Figure). Explain the cases that deadlock occurs and does not occur. (10 pts)

2 i. Write a C program that creates processes and prints out whether child or parent process. (Hint: Use fork(), getpid().) (7 pts)

ii. Write a C program that creates threads and calculates factorial of N by using threads. (Hint: Use phtread_create(), pthread_join(), pthread_exit().) (8 pts)