Operating Systems Comprehensive Exam

Spring 2007

Student ID # _____

3/15/2007

You must complete all of part I (60%)

You must complete two of the three sections in part II (20% each)

In Part I, circle or select only one answer for each question, unless notified otherwise.

If you need more space to answer a question, use the back of a page and indicate which question is being answered.

Part I: You `must` complete `all` of this section.

1. Which of the following is used in a page table to indicate that a memory page has been modified.

   valid/invalid bit **write bit** **dirty bit**

2. A UNIX process calls `fork()` to create a child process as shown: `pid = fork();`

   a) What value will be assigned to `pid` in the parent process by the call to `fork()`?

   the parent's process id **the child's process id** **zero** **none of these**

   b) What value will be assigned to `pid` in the child process by the call to `fork()`?

   the parent's process id **the child's process id** **zero** **none of these**

3. Which of the following process scheduling algorithms can, in some cases, delay requests for an indefinite amount of time (i.e., starvation can occur)?

   FCFS **SJF** **Round-Robin** **none of these is correct**

4. Allows I/O devices to transfer data directly into main memory without passing it through the CPU.

   page faults **DMA** **base registers** **threads** **none of these**

5. If a system is in an unsafe state, it is guaranteed that a deadlock will occur. **True** **False**

   Briefly explain your choice, i.e., why is that answer correct?

6. Deadlock _____ only allocates resources to a process if that allocation

   won't cause a deadlock, while deadlock _____ ensures that at least one of the necessary conditions will not occur.

7. _____, _____, _____ and _____ are the four necessary conditions for a deadlock.

_____, _____ and _____ are the three requirements for a valid critical section.

A. mutual exclusion	**B. circular wait**	**C. progress**
D. recovery	**E. semaphores**	**F. bounded waiting**
G. preemption	**H. non-preemption**	**I. hold and wait**

Matching: choose the best answer for each question from the list below:

8. The term _____ refers to a multi-processor system where one processor assigns tasks to the other processors.

9. When it is not known at compile time where a process will reside in memory, _____ code must be generated.

10. _____ occurs when a process spends more time paging than executing.

11. _____ processes can affect or be affected by the execution of another process.

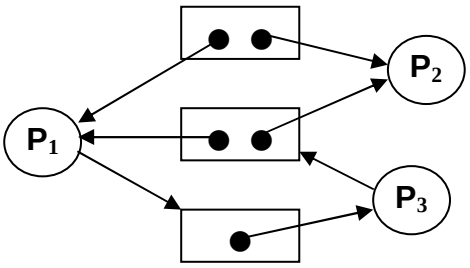
12. _____ is a file allocation method that stores all file block pointers in one block.

13. _____ is commonly used to implement virtual memory.

A. absolute	**B. clustered**	**C. demand paging**
D. cooperative	**E. compaction**	**F. asymmetric**
G. symmetric	**H. static**	**J. thrashing**
K. indexed	**L. fragmentation**	**M. linked**
N. deadlock	**O. asynchronous**	**P. relocatable**
Q. dependent	**R. collaborative**	**S. recovery**

14. Does the following resource allocation graph indicate that a deadlock has occurred?

Yes **No**



15. Does the following pair of operations correctly implement a Semaphore?

```
wait(S)  {
   while(S >= 0);
   S++;
}
```

```
signal(S)  {
   S--;
}
```

Yes **No** If not, briefly explain the error:

16. For each of the following, indicate the type of fragmentation (if any) that is possible:

a) **linked file allocation** internal external both neither

b) **contiguous memory allocation** internal external both neither

c) **memory paging** internal external both neither

d) **memory segmentation** internal external both neither

e) **contiguous file allocation** internal external both neither

f) **indexed file allocation** internal external both neither

17. Briefly describe one advantage and one disadvantage of increasing the round-robin quantum (time slice).

18. You are told to design an operating system that supports threads. One requirement is: if any one user thread makes a system call to do I/O, it can not cause any other user threads to be blocked.

    a) Which threading model would you use? **many-to-one** **one-to-one** **many-to-many**

    b) Briefly explain how the model you chose satisfies the requirement.

19. Answer the following questions based on a system that uses 18-bit memory addressing, a single-level page table and `2k` (i.e., `2048`) byte pages:

    a) what is the maximum number of memory pages for this system? _____

    b) how many bits in the address are used for the page offset? _____

    c) what is the size of a memory frame (in bytes)? _____

20. _____ algorithm is used for deadlock prevention.

    Belady's **Dijkstra's** **Banker's** **none of these is correct**

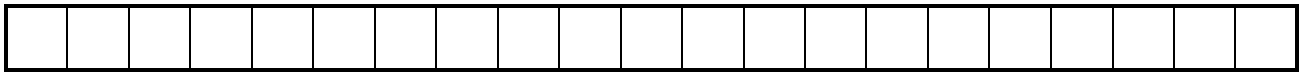
II. You must complete `two` of the following three sections. If you complete more than two sections, clearly indicate which two sections that you want graded. Otherwise, the first two sections will be graded and the third ignored.

A. Given the following set of processes, answer the questions below. Assume that each new process arrives after the interrupted process has been returned to the ready queue. If two processes arrive at the same time, or have the same remaining burst time, schedule them in process number order.

Process Id	Burst Time	Arrival Time
1	5	0
2	2	1
3	3	3
4	5	4

Fill in the following Gantt charts as specified and answer the questions associated with each part. Write the process number of the executing process in the cell for each time unit.

1) **First-Come-First-Served** (non-preemptive):

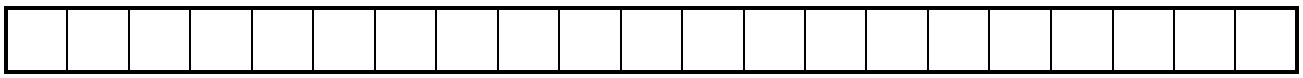


Average waiting time = **Which process had the shortest turnaround? _____**

2) **Shortest-job-first** (preemptive):

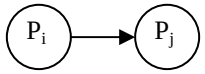
Average waiting time = **What was the average response time? _____**

3) **Round Robin** (time slice (quantum) is 1 time unit):



Average waiting time = **What was the turnaround time for process 3 _____**

B. Assume that $X_{i,j}$ is a semaphore synchronizing processes P_i and P_j, and that all $X_{i,j}$ are initialized to 0. Let S_i represent the statements defining process P_i. On the right of the page, construct a precedence graph showing the order in which the processes will execute. Represent processes P_i and P_j with circles and show a precedence relationship between those processes with an arrow.

Example: 

P_1: { wait($X_{4,1}$);
S_1;
signal($X_{1,2}$);
signal($X_{1,3}$); }

P_2: { signal($X_{2,4}$);
wait($X_{1,2}$);
S_2;
signal($X_{2,3}$); }

P_3: { wait($X_{1,3}$);
wait($X_{2,3}$);
S_3;
wait($X_{4,3}$); }

P_4: { wait($X_{2,4}$);
signal($X_{4,1}$);
S_4;
signal($X_{4,3}$); }

For the statements below, enter **T** if the statement is always TRUE, **F** if it is always FALSE and **U** if an answer can not be determined from the information given.

_____ S_1 completes before S_2 begins.

_____ S_2 completes before S_4 begins.

_____ S_2 completes before S_3 begins.

_____ S_1 completes before S_4 begins.

C. Given the following list of page references, in execution order:

```
4, 2, 5, 1, 5, 2, 3, 5, 1, 4, 2, 3
```

Given the number of available frames shown in the table below, how many page faults will occur for each of the following page replacement algorithms? (All pages are initially empty.)

You must **show your work** to receive partial credit, otherwise only your answers will count.

Algorithm	**# of page faults**
FIFO with 3 frames	
LRU with 4 frames	
Optimal with 4 frames	