Old Operating Systems Exam Questions

January 2001, Questions

1. Provide definitions for the following terms:

   - process;
   - process state;
   - multiprogramming;
   - time sharing.

   [4 marks]

2. Can a process make a transition from the *ready* state to the *blocked* state? Whay or why not? [2 marks]

3. In round-robin scheduling, new processes are placed at the end of the queue, rather than at the beginning. Suggest a reason for this. [5 marks]

4. What is the distinction between user mode and kernel mode? [1 mark]

Paper 3, 2001, Questions

1. What is the difference between local and global page allocation? What are their respective advantages and disadvantages? [6 marks]

2. Consider the following factors:

   (a) internal fragmentation,

   (b) process table size,

   (c) I/O overhead,

   (d) locality of reference.

   Which of these factors could be used to argue for a large page size, and which could be used to argue for a smaller page size? Why? [4 marks]

3. Define the notion of deadlock. What is the difference between a deadlocked and an unsafe state? [5 marks]

4. Which file-allocation method would you use for a system whose main task is database management? Why? [5 marks]

Paper 4, 2001, Questions

1. What is the difference between a process and a thread? Describe some benefits using threads. [5 marks]

2. Why is it a good idea to write the disk-scheduling algorithm as a separate module of the operating system? [6 marks]

3. Assume that n processes apply the fully distributed approach for mutual exclusion in a distributed system. How many messages are required to be sent and received by a single process to enter its critical section? Explain your answer. [4 marks]

January 2002, Questions

1. What are the advantages of using multithreading? Give an example when multithreading is preferable to using single-threaded processes. [4 marks]

2. In a batch system, there are five jobs A through E with run times 2, 4, 1, 1, 1 seconds, respectively. Their arrival times are 0, 0, 3, 3, 3 seconds. What is the turnaround time using the shortest-job-first scheduling algorithm? Is this the optimal turnaround time among the nonpreemptive runs? [5 marks]

3. Consider the Dining Philosophers problem with n philosophers but with $n+1$ forks; the extra fork is in the middle of the table and can be used by any philosopher (but only by one of them at a time). Is deadlock possible? Explain your answer. [3 marks]

Paper 3, 2003, Questions

1. Describe how system calls work. [5 marks]

2. Outline paging in Windows 2000. [6 marks]

3. Explain what hard and symbolic links are. [4 marks]

4. Describe how access control lists and capability lists are used as protection mechanisms. [5 marks]

Paper 4, 2003, Questions

1. Explain the difference between internal and external fragmentations. Why should they be avoided ? [4 marks]

2. Explain how monitors are used in interprocess communication. [6 marks]

3. Why is it a good idea in general to separate policy from mechanism? How can it be achieved in process scheduling? [6 marks]

4. The FAT-16 uses 16 bit addresses. What are the maximum partition sizes for 2, 4 and 8 KB block sizes, respectively? [4 marks]

January, 2004, Questions

1. Describe the difference between preemptive and non-preemptive scheduling algorithms. Which one is more suitable for a timesharing system? [4 marks]

2. Consider the following program:

   while(TRUE)fork();

   Describe what the danger is in running this program on a UNIX machine, and propose a way to limit its ill effect. [4 marks]

3. Name two advantages of using binary semaphores to achieve mutual exclusion among several processes over Peterson's solution. [4 marks]

Paper 3, 2004, Questions

1. There are five processes A to E to run. Their arrival times are 0, 1, 3, 9 and 12 second, respectively. Their processing times are 3, 5, 2, 5 and 5 seconds, respectively. What is the average turnaround time using first-come-first-served, shortest-job-first and round-robin (with 1 second quantum) scheduling? [6 marks]

2. Explain the concept of *working set*, and describe how it can be used in page replacement algorithms. [4 marks]

3. In Linux, semaphores are numbered, and a process can ask for semaphores only in ascending order. How does this help to prevent deadlock? [6 marks]

4. What are the respective advantages and disadvantages of implementing threads in kernel space and in user space? [4 marks]

Paper 4, 2004, Questions

1. Briefly describe how paging is implemented in UNIX. [5 marks]

2. Consider the following sequence of disk track requests: 27, 129, 110, 186, 147, 41, 10, 64, 120. Assume that initially the head is at track 30 and is moving in the direction of decreasing track number. Compute the number of tracks the head traverses using FIFO, SSF and elevator algorithms. [5 marks]

3. Describe the difference between mode switch and process switch. [4 marks]

4. Consider the dining philosopher problem. Assume that some philosophers always pick up their left forks first (a "lefty") and some philosophers always pick up their right forks first (a "righty"). Also assume that there is at least one lefty and one righty at the table. Can deadlock occur? Is starvation possible (assuming a fair scheduling policy)? [6 marks]

January 2005, Questions

1. Briefly describe what data is stored in the process and thread tables. [3 marks]
2. There are five jobs with arrival times 0, 0, 2, 2, and 4 minutes and running times 2, 2, 1, 2, and 2 minutes, respectively. What is the average turnaround time using round-robin scheduling? [5 marks]
3. Explain how linearly ordering resources helps to prevent deadlock. [4 marks]

Paper 3, 2005, Questions

1. Briefly explain how software traps work. [4 marks]

2. How many disk accesses are needed to read a small file (i.e., smaller than the size of a block) from hard disk in UNIX and in WINDOWS 2000. Explain your answer. [4 marks]

3. In the solution for the bounded-buffer problem, we used two counting semaphores, empty and full (besides the binary semaphore mutex). Explain why it is better to use them than keeping track of the empty slots by using an ordinary variable. [6 marks]

4. A small computer has 4 page frames. A process makes the following list of page references: 1, 2, 3, 4, 1, 5, 2, 3, 1, 2. How many page faults occur using FIFO, second chance, and least-recently-used page replacement algorithms? [6 marks]

Paper 4, 2005, Questions

1. Linux supports seven different file types. Name three of them. All the information about a file is stored in a data structure called an i-node (index node). At least nine attributes must be present. Name five of them. When a process opens a file, a data structure called a file object is created. What does this achieve? [6 marks]

2. Both WINDOWS and UNIX boost the priority of I/O-bound processes. Explain why this is a good idea. Can this lead to starvation of CPU-bound processes? [6 marks]

3. What is Belady's anomaly? [2 marks]

4. Consider the following two statements about IPC problems.

   (a) A starvation-free solution is also deadlock-free.

   (b) A deadlock-free solution is starvation-free.

   Are they true or false in general? Explain your answer. [6 marks]

January 2006, Questions

1. Assume a scheduling policy that requires that an unblocked process should get the CPU next; still when a process blocks it is put into the ready state instead of the running state directly. Describe a scheduling mechanism that achieves this goal. [4 marks]

2. Assume that a process is in its critical section guarded by a mutex when it creates a fatal error that causes the process to be killed. Explain how this could effect other processes. Suggest a way the OS can eliminate the problem. [4 marks]

3. Desribe an allocation of resources that yields an unsafe yet not deadlocked state. Show why it is not guaranteed that deadlock will not occur. [4 marks]

Paper 3, 2006, Questions

1. Each process is created with an address space that defines access to every memory-mapped resource in the process. Explain how a process can access objects that are not in its address space (e.g., a file). [4 marks]

2. There are five processes with arrival times 0, 0, 2, 3, 3 seconds and running times 1, 2, 2, 2, 2 seconds, respectively. What is the average turnaround time using

   (a) round-robin [4 marks]

   (b) first-come-first-served [2 mark]

   scheduling?

3. A system has four processes p_1, p_2, p_3, p_4 and three types of dedicated resources R_1, R_2, R_3. The existence vector is $E = (3, 2, 2)$.

   - Process p_1 holds one unit of R_1 and requests one unit of R_2;
   - Process p_2 holds two units of R_2 and requests two units of R_1 and one unit of R_3;
   - Process p_3 holds one unit of R_1 and requests one unit of R_2;
   - Process p_4 holds two units of R_3 and requests one unit of R_1.

   (a) Compute the availibility vector. [1 mark]

   (b) Explain whether the system is deadlocked? [2 marks]

   (c) Determine whether this state of the system is safe? [3 marks]

4. Argue for conditions under which the access control method is inferior to the capability list approach for implementing the protection matrix. [4 marks]

Paper 4, 2006, Questions

1. Both Unix and Windows2000 re-calculate the priorities of processes from time to time. Briefly describe how this dynamic re-allocation of priorities works and explain its benefits. [4 marks]

2. Consider the following pseudo-code:

```
shared boolean flag[2];
flag[0] = flag[1] = FALSE;
proc(int i) {
  while (TRUE) {
    while (flag[(i+1) mod 2] == TRUE);
    flag[i] = TRUE;
    critical_section;
    flag[i] = FALSE;
    }
}
```

   Explain whether this code solves the critical section problem for two processes. [6 marks]

3. In a paging system, page boundaries are transparent to the programmer. Explain how a loop might cause thrashing in a static allocation paging system when the memory allocation is too small. [4 marks]

4. Assume that Windows2000 stores a file on the hard disk using the following runs: (track 12, sectors 1–4), (track 18, sectors 4–8), (track 23, sectors 2–10), (track 35, sectors 3-7). How much seek time is needed to read this file using

   (a) the elevator [2 marks]

   (b) the shortest-seek-first [2 marks]

   algorithm; in both cases the initial position of the head is at track 20, the initial direction of the head is down, and the track-to-track seek time is 5ms. Explain whether this computation would change if, instead of using runs, the sectors storing the file were randomly placed in tracks 12–34? [2 marks]

Paper 3, 2007, Questions

1. Briefly describe how three types of page tables work and compare their benefits. [6 marks]

2. Compute the maximum partition sizes using FAT-16 for block sizes 2 KB and 4 KB. [4 marks]

3. Compute the turnaround time for the following five processes using the multiple queues scheduling algorithm. Processes A, B, and C have priority 1, and processes D and E priority 0 (the higher the priority, the sooner the process is scheduled). Their arrival times are 0, 1, 2, 0 and 5 seconds, and their run times are 1.5, 1, 1.5, 1.5 and 1.5 seconds, respectively. [6 marks]

4. For the producer–consumer problem we have seen solutions using semaphores and using message passing. Briefly compare the benefits of the two techniques. [4 marks]

Paper 4, 2007, Questions

1. Assume that the scheduler is implemented as a process. Describe a mechanism that would enable the operating system to decide when to run the scheduler process. [4 marks]

2. Can Belady's anomaly happen to the optimal page replacement algorithm? Explain your answer. [4 marks]

3. We discussed Tanenbaum's solution to the readers–writers IPC problem. It gives preference to readers: writers cannot enter the database as long as there are readers trying to enter the database. Briefly describe how to modify the solution to avoid starvation of writers. [6 marks]

4. A hard disk spins at 7200 rpm, and tracks contain 100 sectors of size 512 bytes. The average seek time between tracks is 6 msec. Compute how much time it takes in average to read a file of size 40 KB

   (a) if the blocks of the file occupy consecutive sectors on one track

   (b) if the file is stored in eight equal size runs; the runs are randomly distributed on the disk and a run consists of contiguous blocks of the same track.

   In the second case, assume that the read is done in the order the blocks appear in the file, i.e., using first-come-first-served algorithm. [6 marks]

Paper 3, 2008, Questions

1. Explain what interrupts are and briefly describe how the operating system handles them. [4 marks]

2. We have seen that round-robin scheduling can be modified by listing multiple occurrences of processes so that it gives the amount of CPU-time to each process according to its priority. Compute the average turnaround time for the following processes using this scheduling algorithm: Processes A, B, C, D and E are created at 0, 0, 4, 4 and 6 minutes, their runtimes are 3, 2, 2, 5 and 6 minutes and their priorities are 3, 1, 2, 1 and 4 (the higher the priority, the more CPU-time a process gets). [6 marks]

3. Explain why it makes sense to store recently used data (e.g., recently referenced pages) in cache. [4 marks]

4. Peterson's solution is used in distributed systems for achieving mutual exclusion. Yet, it suffers from the priority inversion problem. Describe what the problem is and explain how it is prevented in modern operating systems. (Hint: How do Unix and Windows2000 allocate priorities?) [6 marks]

Paper 4, 2008, Questions

1. Consider the following proposed solution to the dining philosophers problem.

```
semaphore fork[5];
semaphore room=4;

void philosopher(int i)
{while(TRUE)
 think();
 down(room);
 down(fork[i]);
 down(fork[i+1 mod 5]);
 eat();
 up(fork[i]);
 up(fork[i+1 mod 5]);
 up(room);
}
```

   Briefly explain whether deadlock and starvation can occur. [6 marks]

2. Explain what device controllers and device drivers are, and briefly describe their purposes. [4 marks]

3. Describe how deadlock prevention works by denying some of the necessary conditions of a deadlock. [6 marks]

4. Briefly compare the file allocation methods used by Unix and Windows2000. What are their respective advantages? [4 marks]

Paper 3, 2009, Questions

1. Explain what deadlock avoidance is and apply it to the following example. There are five processes (A to E) and four types of resources. Resources are assigned as follows: A: (3,0,1,0), B: (0,1,0,0), C: (1,1,1,0), D: (1,1,0,1), E: (0,0,0,0). The additional request are: A: (1,1,0,1), B: (0,1,1,2), C: (3,1,0,0), D: (0,0,1,0), E: (2,1,1,0). The availability vector is (1,0,2,1). Determine whether the request by A for one item of resource type 4 should be granted. [8 marks]

2. Explain the three fundamental ways I/O can be performed. For each method determine how much CPU time is needed to print a 1KB document. The initial set-up takes 20 millisecond in each case, it takes 50 microsecond to put a byte in the data register of the controller of the printer, and printing one byte takes 50 millisecond. Assume that each interrupt service procedure needed runs for 70 microsecond and that the data register of the controller of the printer can contain only one byte. [8 marks]

3. Briefly explain how messages can be used to achieve mutual exclusion. What is the main advantage of messages compared to semaphores and monitors? [4 marks]

Paper 4, 2009, Questions

1. The test-and-set atomic machine instruction is defined as follows:

```
boolean test_and_set (int i)
{
  if (i == 0)
   { i= 1;
   return true;
  }
  else return false;
}
```

   Explain how test-and-set can be used to achieve mutual exclusion. Does your solution avoid busy waiting? [8 marks]

2. Explain what multilevel (aka hierarchical) page tables are and their advantages over one-level page tables. How many pages can be in the virtual address space using a two-level page table where every table (both top-level and second-level) has 1K entries? [6 marks]

3. List the main criteria a (short-term) scheduling algorithm should satisfy in a time-sharing system. Assume that a process had the following CPU bursts: 8, 32, 16, 8, 8 microseconds. Compute the estimated length of the next CPU burst using simple and exponential (aka weighted) averaging with parameter 0.5. [6 marks]