

Spring 99

ECES 338 Midterm #2 Name: KEY  
April 19, 1999 Ozsoyoglu,G.

The exam is worth 100 points. There are 16 questions. Answer all questions.

You are to use **ONLY** the space provided, **NOT** the back of the page.

Please be **brief, succinct, and to-the-point** in your answers.

You will loose points for **irrelevant AND incorrect statements** even with correct answers!

### INTRO

(12 pts) 1) Circle the correct statements.

- T (1) A process is an independent execution of a program on resources assigned to it.
- T (2) The ability of an O.S. to run a program with different IO devices is called device independence.
- T (3) Spooling essentially uses the disk as a very large buffer, for reading as far ahead as possible on input devices and for storing output files until the output devices are able to accept them.
- T (4) A trap is a software-generated interrupt.
- F (5) A daemon is a generic name for memory management processes in operating systems.
- T (6) Swapping technique was used for rudimentary multiprogramming in early OSs.

### UNIX

(4 pts) 2) Circle the correct statements.

- F (1) UNIX 4.3BSD uses First-Come-First-Serve CPU scheduling with no process priorities.
- F (2) UNIX shell is a command interpreter located inside the UNIX kernel, and does I/O.

(3 pts) 3) State the differences between fork and vfork system calls.

*fork() spawns a child process, and gets a new stack+data space (a copy of the parent's) for the new process.*

*vfork() spawns a child process, and the child shares with the parent the same stack+data space.*

(3 pts) 4) What is the *effective user id* of a process, and how does a process get (acquire) it?

*Effective user id provides a mechanism through which a process accesses resources that are not available to the owner of the process. When a "setuid" binary file (i.e., a file with a set "setuid" bit) is execve'd by a process, the user id of the file's owner becomes the effective user id of the process. Then the process can access all the resources accessible to the file's owner.*

*The effective user id can also be set by the system call seteuid().*

### DEADLOCKS

**(6 pts) 5)** Circle the true statements.

- T** (1) In a resource allocation graph with multiple instances of resource types, if there are no cycles then no deadlocks exist.
- F** (2) In a resource allocation graph with multiple instances of resource types, if there is a cycle then a deadlock exists.
- F** (3) In an environment with multiple instances of resource types, one sufficient condition for the existence of deadlocks is the existence of a circular hold-and-wait condition.

**(3 pts) 6)** State three necessary conditions for deadlocks.

*Hold-and-wait, no preemption, mutual exclusion, circular wait.*

**(10 pts) 7)** (a) What is a "safe state" in deadlock avoidance techniques?

*It is a state of the system where the system can \*sequentially\* execute \*all\* the processes in some order, allocating resources up to the max requested by each process, and deallocating all the resources once the process completes. If all the processes can complete in this manner then the system is in a safe state.*

(b) Consider the Banker's algorithm and the following snapshot of processes ( $p_i$ 's) and resources ( $r_j$ 's):

Allocation Matrix					Max Matrix					Available Matrix				
	$r_1$	$r_2$	$r_3$	$r_4$	$r_1$	$r_2$	$r_3$	$r_4$	$r_1$	$r_2$	$r_3$	$r_4$		
$p_1$	0	0	1	2	3	0	4	2	1	5	2	0		
$p_2$	1	0	0	0	1	7	5	0						
$p_3$	1	3	5	4	3	3	5	6						
$p_4$	0	6	3	2	0	6	5	2						
$p_5$	0	0	1	4	0	6	5	8						

Is the system in a safe state? Explain.

*Yes, there is a safe sequence: ( $p_4, p_2, p_3, p_5, p_1$ )*

## DEVICE AND SECONDARY STORAGE MANAGEMENT

**(5 pts) 8)** The most general directory structure used in Operating Systems today is an acyclic directed graph structure where nodes with no outgoing edges represent files, and the rest of the nodes are directories. List two problems associated with using arbitrary graph structures as directory structures in operating systems.

*For arbitrary graphs, we may have the following problems:*

*(1) Searching and looping over a cycle needs to be recognized, and broken.*

*(2) Deleting a file directory which is pointed from other directories leaves "dangling" pointers that need to be eliminated.*

**(5 pts) 9)** How are disk devices protected from users' malicious/accidental destruction of other users' data on them?

*The OS (which is trusted and not malicious) performs the writes on behalf of users; and, before any write, (a) checks file boundaries, and (b) enforces access control specifications (e.g., r/w/x, group access controls, etc.).*

**(5 pts) 10)** A leading computer designer (Fred Brooks?) has stated: "there is no point in interrupting a CPU to handle I/O because the CPU won't service the interrupt until it is ready. It is better for the CPU, when ready, to poll for I/O." Do you agree? If so, why? If not, why not?

*Disagree:*

*(1) If an interrupt of a device is not serviced, other I/Os waiting for that device will not be serviced.*

*(2) Servicing the interrupt means that the associated process/thread will go into its ready queue earlier, possibly improving the CPU scheduling and CPU utilization.*

*Agree:*

*(1) Overhead due to forced context switching with interrupts will be eliminated.*

*(2) Polling overhead is much smaller.*

**(2 pts) 11)** Name three disk I/O scheduling techniques.

LOOK, SCAN, CSCAN

## MEMORY MANAGEMENT and CPU SCHEDULING

**(20 pts) 12)** Circle the correct statements.

- T** (1) Belady's anomaly illustrates a case such that, given the FIFO page replacement algorithm and the execution of the same program, one can have more page faults with more physical memory.
- T** (2) The local page replacement policy states that, when a page is needed to be brought to the main memory for a process, the candidate pages to be kicked out of the main memory are only those that belong to the program that the process is executing.
- F** (3) A process scheduling policy is preemptive if the CPU cannot be forcibly removed from a process.
- F** (4) Because the shortest-job-first policy gives preference to short jobs, it is useful in timesharing.
- T** (5) Real-time systems generally use preemptive CPU scheduling.
- T** (6) Timesharing systems generally use nonpreemptive CPU scheduling.
- T** (7) One weakness of priority schemes is that the system will faithfully honor the priorities, but the priorities themselves may not be meaningful.
- T** (8) The number of processes that are completed per time unit is called the throughput of the system.
- T** (9) Swapper is an O.S. process that chooses some user processes from among those in the ready queue to temporarily suspend.
- T** (10) Turnaround time of a process  $p$  is the actual clock time interval that  $p$  exists.

**(5 pts) 13)** Name three CPU scheduling policies that have a scheduling cost independent of the number of processes in the ready list.

*FCFS, RR, Multilevel Queues.*

**(5 pts) 14)** How do Operating Systems with multiprogramming prevent the accidental/malicious destruction of users' address spaces by each other?

*The OS (which is trusted and not malicious) performs the writes on behalf of users, and, before each write, checks address space boundaries. The boundary checking is facilitated by either upper-and-lower-bound registers, or base+limit registers.*

**(7 pts) 15)** Discuss **briefly** how fragmentation manifests itself in each of the following types of memory management mechanisms.

(a) Segmentation

*Space between segments will be fragmented (external fragmentation).*

(b) Paging

*On the average, 1/2 page per active process will be fragmented (internal fragmentation).*

(c) Paged Segmentation

*On the average, 1/2 page per active process will be fragmented (internal fragmentation).*

**(5 pts) 16)** Some paged memory management systems do I/O with systemwide buffers. Other systems do I/O directly with user buffer areas in each user's paged logical memory. Discuss the relative merits of each approach.

*System-Wide Buffers:*

*Advantages: Can be kept out of virtual memory management mechanisms (no page locking needed).*

*Disadvantages: There is an extra step of transferring the data to users' logical address space. Shared buffering is complicated.*

*User Buffers:*

*Advantages: Data is directly transferred to/from I/O devices (fast and simple).*

*Disadvantages: Requires page locking.*