QUIZ 1

1. What is the purpose of interrupts? What are the differences between a trap and an interrupt?  
   **Interrupts alert the operating system of the occurrence of an event in the system. They consist of a signal to the CPU generated by hardware (interrupt) or software (trap).**
2. What are multiprogramming and multitasking and what is the difference between them?  
   **Multiprogramming is keeping multiple jobs/processes in memory so one is already ready to execute. Multitasking is an extension of multiprogramming where the CPU frequently switches between the jobs/processes to give the impression that they are running at the same time. Multiprogramming alone doesn't imply this switching.**
3. What is the difference between a system call and a system program?  
   **A system call is a routine providing an operating system service, typically available as an API. A system program (or system utility) is a program that also provides operating system services (or other useful functionality) but with a more convenient interface for the user than the equivalent system calls; a system program is typically an executable file and/or shell command.**

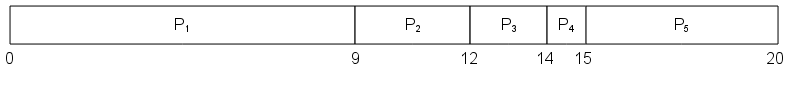
QUIZ 2

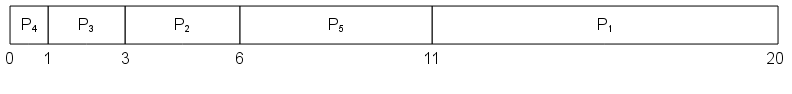
* Which of the following components of program state are shared across threads in a multithreaded process?  
  a. Register values  
  b. Heap memory  
  c. Global variables  
  d. Stack memory  
  **b. Heap memory and c. Global variables are shared across threads.**
* What are the two models of interprocess communication? What are the strengths and weaknesses of the two approaches?  
  **The two models are shared memory and message-passing. Shared memory allows direct exchanges of information, but requires processes to manage access to that information to prevent conflicts from occurring. Message-passing prevents conflicts, but also incurs additional overhead.**
* What will be the output of the following program (assume there are no errors in execution)?  
  int main()  
  {  
    pid\_t pid, pid1;  
    pid = fork();  
    if (pid == 0) {  
      pid1 = getpid();  
      printf("child: pid = %d, pid1 = %d", pid, pid1);  
    } else {  
      pid1 = getpid();  
      printf("parent: pid = %d, pid1 = %d, pid, pid1);  
      wait(NULL);  
    }  
    return 0;  
  }  
  **child: pid = 0, pid1 = <child pid>**  
  **parent: pid = <child pid>, pid1 = <parent pid>**

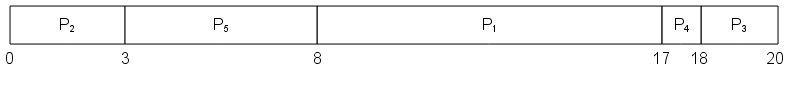
QUIZ 3

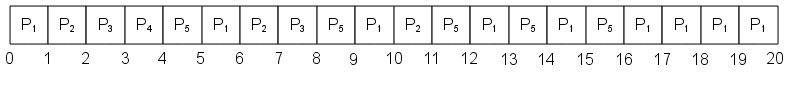
Circle the scheduling algorithm(s) that can result in starvation:  
a. First-come, first-served  
b. Shortest job first  
c. Round robin  
d. Priority  
**b. Shortest job first and d. Priority can result in starvation.**

1. Describe how the Swap() instruction can be used to provide mutual exclusion. Include code/pseudocode.  
   **Swap() can be used to atomically switch two values, so that when a shared variable changes to a value allowing a waiting process to proceed, only one of the waiting processes actually sees this value. The following pseudocode shows how processes should use Swap() with a shared variable to protect access to a critical section.  
   boolean global = false; // shared variable  
   ...  
   boolean local = true; // local variable  
   while (local)  
     Swap(&global, &local);  
   // critical section  
   global = false;**
2. Consider the following set of processes, with the length of the CPU burst given in milliseconds:  
   Process    Burst Time    Priority  
       P1                9                 3  
       P2                3                 1  
       P3                2                 5  
       P4                1                 4  
       P5                5                 2  
   The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0.  
   a. Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1).  
   **FCFS**

**SJF**

**nonpreemptive priority**

**RR**

b. What is the turnaround time of each process for each of the scheduling algorithms in part a?  
**FCFS  
P1 = 9, P2 = 12, P3 = 14, P4 = 15, P5 = 20**  
**SJF**  
**P1 = 20, P2 = 6, P3 = 3, P4 = 1, P5 = 11**  
**nonpreemptive priority**  
**P1 = 17, P2 = 3, P3 = 20, P4 = 18, P5 = 8**  
**RR**  
**P1 = 20, P2 = 11, P3 = 8, P4 = 4, P5 = 16**  
c. What is the waiting time of each process for each of these scheduling algorithms?  
**FCFS  
P1 = 0, P2 = 9, P3 = 12, P4 = 14, P5 = 15**  
**SJF**  
**P1 = 11, P2 = 3, P3 = 1, P4 = 0, P5 = 6**  
**nonpreemptive priority**  
**P1 = 8, P2 = 0, P3 = 18, P4 = 17, P5 = 3**  
**RR**  
**P1 = 11, P2 = 8, P3 = 6, P4 = 3, P5 = 11**  
d. Which of the algorithms results in the minimum average waiting time (over all processes)?  
**SJF results in the minimum average wait time.**

QUIZ 4

Assuming a byte-addressed system with 16-bit logical and physical addresses and 8KB pages, translate the following logical addresses into physical addresses using the provided page table:  
page    frame  
  0           0  
  1  
  2           7  
  3           4  
  4  
  5           1  
  6           3  
  7           2  
a. 0xAAC4  
    **0x2AC4**  
b. 0x02C8  
    **0x02C8**  
c. 0xE310  
    **0x4310**

1. Name the types of fragmentation created by contiguous memory allocation and paging, respectively (one for each), and describe the difference between them (the two kinds of fragmentation).  
   **Contiguous memory allocation causes external fragmentation. Paging causes internal fragmentation. Both internal and external fragmentation involve wasted memory. External fragmentation consists of memory holes (sections of unallocated memory between sections of allocated memory) that are too small to fulfil memory requests. Internal fragmentation consists of the memory that gets allocated to processes that exceeds the processes' actual memory needs (due to memory being allocated in fixed-size chunks).**
2. What is the difference between the signal operation invoked on condition variables within monitors and the signal operation invoked on semaphores?  
   **The signal operation invoked on a condition variable within a monitor wakes up one process waiting on the condition variable if any such process exists; otherwise it does nothing. The signal operation invoked on a semaphore wakes up one process waiting on the semaphore if any such process exists; otherwise it allows the next process that calls wait on the semaphore to proceed immediately (or allows one additional process to do so if the semaphore has a count greater than zero).**

QUIZ 5

Under what circumstances do page faults occur? Describe the actions taken by the operating system when a page fault occurs.  
**A page fault occurs when a process tries to access a memory location for which the page is not mapped to a physical frame. This causes a page fault exception, triggering the page-fault service routine. This routine first finds the page on disk. It then locates a free frame. If no frames are free, then the page-replacement algorithm is executed to free a frame, scheduling a disk operation to write any modified data back to disk if necessary. Then a disk I/O operation is scheduled to read the new page into the newly freed frame. During the I/O time, other processes may be allocated the CPU. Once the page has been read into the free frame, the process's page table is updated with the page-to-frame mapping (including setting the valid-invalid bit to valid) and the process is restarted at the instruction that generated the page fault.**

1. Consider the following page reference string:  
   1, 3, 2, 4, 3, 5, 6, 7, 2, 3, 2, 1, 7, 6, 5, 4, 7, 2, 5, 6  
   How many page faults would occur for the following replacement algorithms, assuming one, four, and seven frames? Remember that all frames are initially empty, so your first unique pages will cost one fault each.  
   LRU replacement  
   **1 frame: 20**  
   **4 frames: 15**  
   **7 frames: 7**  
   FIFO replacement  
   **1 frame: 20**  
   **4 frames: 16**  
   **7 frames: 7**  
   Optimal replacement  
   **1 frame: 20**  
   **4 frames: 11**  
   **7 frames: 7**
2. Consider the parameter Δ used to define the working-set window in the working-set model. What is the effect of setting Δ to a small value on the page-fault frequency and the number of active (nonsuspended) processes currently executing in the system? What is the effect when Δ is set to a very high value?  
   **Setting Δ to a small value will increase the page-fault frequency and increase the number of active processes, since each process will have fewer frames allocated to it. Setting Δ to a very high value will have the opposite effect.**

QUIZ 6

What are the two access modes in which a file can be opened and what is the difference between them?  
**The two access modes are direct access and sequential access. With direct access, a process can read or write any location of the file. With sequential access, the process has a file pointer that determines where the next read or write will occur. The file pointer may be moved with the seek operation.**

1. What are the advantages of the variant of linked allocation that uses a FAT to chain together the blocks of a file?  
   **As with linked allocation, the file allocation table eliminates external fragmentation by allowing any block to be used to satisfy a storage allocation request. At the same time, it does not reduce the effective size of blocks by requiring them to contain block pointers. It also keeps track of free blocks. Finally, by keeping the block pointers all together, it allows more of them to be cached in main memory.**
2. Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4999, with the read-write head currently over cylinder 999. The queue of pending requests, in order of arrival, is:  
   57, 3543, 913, 2774, 2948, 1509  
   Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms (assuming the head's initial movement is toward higher numbered cylinders for c-f)?a. FCFS **10532**b. SSTF **6202**c. SCAN **8942**d. LOOK **6030**e. C-SCAN **9912**f. C-LOOK **6886**