Quiz 1- Rubric

Subjective Questions

Q1. Consider a parent process P that has forked a child process C in the program below.

```
 int a = 5;
  int fd = open(...) //opening a file
  int ret = fork();
  if(ret >0) {
   close(f;
    a = 6;
    ...
 }
  else if(ret==0) {
   printf("a=%d\n", ;
   read(fd, something);
  }
```

After the new process is forked, suppose that the parent process is scheduled first, before the child process. Once the parent resumes after the fork, it closes the file descriptor and changes the value of a variable, as shown above. Assume that the child process is scheduled for the first time only after the parent completes these two changes.

(a) What is the value of the variable a as printed in the child process when it is scheduled next? Explain.

(b) Will the attempt to read from the file descriptor succeed in the child? Explain.

Ans:

(a) 5. The value is only changed in the parent.
(b) Yes, the file is only closed in the parent.

Q2. For what types of workloads does SJF deliver the same turnaround times as FIFO?

Ans. SJF delivers the same turnaround times as FIFO when all processes have the same burst time. In this scenario, SJF and FIFO produce identical scheduling results because each process's burst time is the same

Q3. A context switch can occur only after processing a timer interrupt but not after any other system call or interrupt. [True/False with Justification]

Ans: False, a context switch can also occur after a blocking system call for example.

Q4. Answer yes/no, and provide a brief explanation.

(a) Can two processes that are not parent/child be concurrently executing the same program executable?
(b) Can two running processes share the complete process image in physical memory (not just parts of it)?

Ans:

(a) Yes, two processes can run the same program executable, which happens when we run the same executable from the terminal twice.
(b) No. In general, each process has its own memory image.

Q5. Consider the following events that happen during a context switch from (user mode of) process P to (user mode of) process Q, triggered by a timer interrupt that occurred when P was executing, in a Unix-like operating system design studied in class. Arrange the events in chronological order, starting from the earliest to the latest.
(A) The CPU program counter moves from the kernel address space of P to the kernel address space of Q.
(B) The CPU executing process P moves from user mode to kernel mode.
(C) The CPU stack pointer moves from the kernel stack of P to the kernel stack of Q.
(D) The CPU program counter moves from the kernel address space of Q to the user address space of Q.
(E) The OS scheduler code is invoked.

Ans:

BECAD

Q6. Consider a parent process that has forked a child in the code snippet below.

```
int count = 0;
 ret = fork();
 if(ret == 0)  {
 printf("count in child=%d\n", count);
 }
else  {
 count = 1;
 }
```

The parent executes the statement "count = 1" before the child executes for the first time. Now, what is the value of count printed by the code above?

Ans: 0 (the child has its own copy of the variable).

Q7. A process in user mode cannot execute certain privileged hardware instructions. [True/False with Explanation]

Ans: True, some instructions in every CPU's instruction set architecture can only be executed when the CPU runs in a privileged mode.

Objective Questions

Q1. Consider a simple linux shell implementing the command sleep 100. Which of the following is an accurate ordered list of system calls invoked by the shell from the time the user enters this command to the time the shell comes back and asks the user for the next input?

A. wait-exec-fork
B. exec-wait-fork
C. **fork-exec-wait**
D. wait-fork-exec

Q1. Ans: (c)

Q2. A parent process uses the wait system call on its child process. Which of the following statements is true?

A. The parent will always block.
B. The parent will never block.
C. The parent will sometimes block and sometimes does not.
D. None of the above.

Q2. Ans: (A)

Q3. In which of the following situations does a context switch occur in an operating system?

A. **When multiple processes are timesharing a CPU**
B. When a process requests I/O
C. When a hardware device signals the CPU for attention
D. When a process has begun to run according to its schedule.

Q3. Ans: (A, B, C)

Q4. Which of the following is a characteristic feature of the Multilevel Feedback Queue (MLFQ) scheduling algorithm?

A. Fixed priority levels without dynamic adjustments
B. All processes are executed in a strict round-robin manner
C. Processes can have their priority changed dynamically based on their behaviour
D. Processes are always executed in the order they arrive

Q4. Ans: (c)

Q5. In an MLFQ scheduling algorithm, how does a process typically move from one queue to another?

A. Based on the arrival time of the process
B. Based on the process's execution time and behaviour
C. Based on the process's priority level
D. Based on the I/O operations performed by the process

Q5. Ans: (B, C)

Q6. Process are classified into different groups in _____

A. Shortest job scheduling algorithm
B. Round robin scheduling algorithm
C. Lottery scheduling algorithm
D. Multilevel queue scheduling algorithm

Q6. Ans: (d)

Q7. Which of the following scheduling algorithms could result in starvation?

A. First-Come, First-Served (FCFS)
B. Shortest Job First (SJF)
C. Round Robin (RR)
D. None of the above

Q7. Ans: (b)

Q8. In a Round Robin (RR) scheduling algorithm, what happens when the time quantum is too small?

A. All processes are terminated.
B. There are too many context switches.
C. The response time is increased.
D. The processes are completed faster.

Q8. Ans: (b)

Q9. Which of the following is true regarding the Multilevel Feedback Queue Scheduling?

A. It is designed for systems with very few processes
B. It is suitable for real-time systems
C. Each queue may have its own scheduling algorithm
D. None of the above

Q9. Ans: (c)

Q10. When a process is in the "terminated" state, what happens to its Process Control Block (PCB)?

A. It is suspended in memory for future use.
B. It is deleted and resources are freed.
C. It is moved to the ready queue.
D. It is stored in the CPU cache for quick access.

Q10. Ans: (b)

Q11. What is the primary purpose of saving the process's state during a context switch?

A. To increase the CPU speed by optimizing register use
B. To allow the CPU to execute a different process while preserving the state of the current process
C. To enhance the performance of I/O operations by reducing interrupt frequency
D. To reset the CPU registers to their initial state for each new process

Q11. Ans: (b)

Q12. When a process enters the "blocked" state, it means:

A. The process is waiting for a CPU time slice.
B. The process is waiting for an I/O operation to complete.
C. The process has completed execution.
D. The process is waiting for its child process to terminate.

Q12. Ans: (b)

Q13.Which of the following system calls is used to replace the current process image with a new one in UNIX-based systems?

A. fork()
B. exec()
C. wait()
D. signal()

Q13. Ans: (b)

Q14. What is the primary reason for context switching in an operating system?

A. To save energy by putting the CPU into idle mode
B. To ensure fair allocation of CPU time among processes
C. To handle I/O operations in the background
D. To reset the CPU registers to their initial state

Q14. Ans: (b)

Q15. During a context switch, which of the following operations is performed first?

A. The CPU registers of the current process are saved.
B. The memory pages of the current process are swapped out.
C. The process control block (PCB) of the new process is loaded.
D. The stack of the current process is cleared.

Q15. Ans: (a)

Q16.Which of the following can reduce the number of context switches in a system?

A. Decreasing the size of time quanta in a Round Robin scheduler
B. Increasing the time slice allocated to each process
C. Increasing the number of processes in the system
D. Switching to a priority-based non-preemptive scheduling algorithm

Q16. Ans: (B, D)

Q17 .What is the primary function of the Process Control Block (PCB) in process execution?

A. To manage memory allocation for processes

B. To store the state of the CPU registers and process information
C. To handle I/O operations for a process
D. To perform context switching between processes

Q17. Ans: (b)

Q18. In which state does a process reside when it is waiting for an event to occur, such as the completion of I/O?

A. Running
B. Ready
C. Blocked
D. Terminated

Q18. Ans: (c)

Q19. Which of the following best describes preemptive process scheduling?

A. The operating system gives each process a fixed time to run and forcibly switches to another process when the time expires.
B. Processes run until they voluntarily give up the CPU.
C. Processes run in the background without any CPU scheduling.
D. Only one process can run at a time in the CPU.

Q19. Ans: (a)

Q20. Which of the following is true about the "fork" system call in UNIX-based systems?

A. It is used to terminate a process.
B. It creates a new process by duplicating the calling process.
C. It replaces the current process image with a new one.
D. It waits for a child process to terminate.

Q20. Ans: (b)

Q21. Which of the following is NOT a function of the operating system?

A. Process management
B. Memory management
C. File management
D. Data encryption

Q21. Ans: (d)

Q22. If a process calls fork() and successfully creates a new child process, what will be the return value of fork() in the child process?

A. 0
B. -1
C. The Process ID (PID) of the parent
D. The Process ID (PID) of the child

Q22. Ans: (a)

Q23. What is the primary purpose of a system call in an operating system?

A. To request services from the CPU
B. To provide an interface between user programs and the operating system
C. To manage file systems
D. To run the process scheduler

Q23. Ans: (b)