Student Name:..……………………KEY…………………………………. UIN:………………………………..

**Student Score / 55**

True/False Questions [20 pts] **I highlighted only the false answers**

1. The executable image of a program must be loaded into the main memory first before executing
2. An Operating System (OS) does not trust application programs because they can be either buggy or malicious
3. There was no concept of OS in first generation computers
4. The PC register of a CPU points to the next instruction to execute in the main memory
5. Second generation computers still executed programs in a sequential/batch manner
6. Time sharing computers gave a fixed time quantum to each program
7. An OS resides in-between the hardware and application programs
8. The primary goal of OS is to make application programming convenient
9. Context switching does not contribute much to the OS overhead
10. Main Memory access is slower than register/cache access because it is physically outside the CPU
11. Multiprogramming cannot work without Direct Memory Access (DMA) mechanism
12. Interrupts are necessary for asynchronous event handling in a CPU
13. A program can be kicked out of a CPU when it requests I/O operation, or when another Interrupt occurs
14. A program error can kick a program out of CPU
15. Interrupts are necessary to bring a program back to CPU if it was previously kicked out
16. The “Illusionist” role of the CPU allows a programmer write programs that are agnostic of other programs running in the system
17. Modern operating systems come with many utility services that are analogous to the “Glue” role of the OS
18. Networking service is not a core OS part, rather a common service included with most OS
19. Resource allocation and Isolation are not part of the core OS, rather common services included with OS
20. Efficiency is the secondary goal of an OS

Short Questions

1. [5 pts] Define multiprogramming. How is this better than sequential program execution?

*It is a technique to keep the CPU utilized when the current program requests I/O operation. For this method to work, multiple programs must stay loaded in the memory so that they can brought in quickly. Another important factor to avoid CPU involvement in every I/O step is the DMA controller, which takes care of the back-and-forth with the I/O device for sending commands.*

*It is better than sequential execution because the CPU and I/O device utilization is higher in this model resulting in higher throughput (i.e., more things getting done in the same time period).*

1. [20 pts] In a single CPU single core system, schedule the following jobs to take the full advantage of multiprogramming. The following table shows how the jobs would look like if they ran in isolation.

|  |  |  |  |
| --- | --- | --- | --- |
|  | JOB1 | JOB2 | JOB3 |
| Type of job | Full CPU | Only I/O | Only I/O |
| Duration | 5 min | 15 min | 10 min |
| Memory required | 50MB | 100MB | 75MB |
| Needs disk? | No | No | Yes |
| Needs terminal? | No | Yes | No |

1. Fill out the multiprogramming column in the following table (i.e., when the jobs are scheduled in multiprogramming). Assume that the system’s physical memory is 256MB.

|  |  |  |
| --- | --- | --- |
| Average Resource Use | Sequential | Multiprogramming |
| Processor | 5/30 = 16.67% |  |
| Memory | 32.55% |  |
| Disk | 33.33% |  |
| Terminal | 50% |  |

Memory usage is computed as follows: (5minx50MB + 15minx100MB + 10minx75MB) / (30minx256MB) = 32.55%

Other resources are fully utilized during the time they are utilized. So, you compute utilization only based on the duration they are used.

1. What is the total time for completion for all jobs in sequential and multiprogrammed model.

**Answer:**

*The following is the schedule of the above 3 jobs in a multiprogrammed system:*

*So, the completion time is 15 mins as follows:*

|  |  |  |  |
| --- | --- | --- | --- |
| *Time (min) 🡪* | *0 5* | *6 10* | *11 15* |
| *CPU* | *Job1* |  |  |
| *Memory(total 256MB)* | *225 MB* | *175 MB* | *100 MB* |
| *Terminal* | *Job2* | | |
| *Disk* | *Job3* | |  |

*The following is utilization of different resources:*

|  |  |  |
| --- | --- | --- |
| *Average Resource Use* | *Sequential* | *Multiprogramming* |
| *Processor* | *5/30 = 16.67%* | *5/15 = 33.33%* |
| *Memory* | *32.55%* | *65.10% \** |
| *Disk* | *33.33%* | *66.67%* |
| *Terminal* | *50%* | *100%* |

*\*Memory Utilization = (5minx225MB + 5minx175MB + 5minx100MB) / (15minx256MB) = 65.10%*

*Alternate Answer:*

*If you assume that since the CPU is fully utilized for the first 5 mins and you cannot even start Job2 or Job3 within that time period, you would be correct as well. The completion time is 20 min in that case where the schedule is as follows:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Time (min) 🡪* | *0 5* | *6 10* | *11 15* | *16 20* |
| *CPU* | *Job1* |  |  |  |
| *Memory* | *50 MB* | *175 MB* | | *100 MB* |
| *Terminal* |  | *Job2* | | |
| *Disk* |  | *Job3* | |  |

*Using the above, the utilizations now look like the following:*

|  |  |  |
| --- | --- | --- |
| *Average Resource Use* | *Sequential* | *Multiprogramming* |
| *Processor* | *5/30 = 16.67%* | *5/20 = 25%* |
| *Memory* | *32.55%* | *48.82% +* |
| *Disk* | *33.33%* | *50%* |
| *Terminal* | *50%* | *75%* |

*+Memory Utilization = (5minx50MB + 10minx175MB + 5minx100MB) / (20minx256MB) = 48.82%*

1. [4 pts] Give an example of the “Referee” role of the OS.

*Memory protection*

1. [6 pts] Give an example of a “Reliable but Unavailable” system, and one for a “Unpredictable” system.
   1. *A system restarts all the time*
   2. *A system that takes different amount of time to sort some data. A random CPU scheduler may sometimes be lucky scheduling jobs correctly. But at other times, such scheduler may take much longer.*