Student Name:Sahil Palnitkar UIN:726001905

**Student Score / 55**

True/False Questions [20 pts]

1. The executable image of a program must be loaded into the main memory first before executing

**True**

1. An Operating System (OS) does not trust application programs because they can be either buggy or malicious

**True**

1. There was no concept of OS in first generation computers

**True**

1. The PC register of a CPU points to the next instruction to execute in the main memory

**True**

1. Second generation computers still executed programs in a sequential/batch manner

**True**

1. Time sharing computers gave a fixed time quantum to each program

**True**

1. An OS resides in-between the hardware and application programs

**True**

1. The primary goal of OS is to make application programming convenient

**False**

1. Context switching does not contribute much to the OS overhead

**False**

1. Main Memory access is slower than register/cache access because it is physically outside the CPU

**True**

1. Multiprogramming cannot work without Direct Memory Access (DMA) mechanism

**True**

1. Interrupts are necessary for asynchronous event handling in a CPU

**True**

1. A program can be kicked out of a CPU when it requests I/O operation, or when another Interrupt occurs

**True**

1. A program error can kick a program out of CPU

**True**

1. Interrupts are necessary to bring a program back to CPU if it was previously kicked out

**True**

1. The “Illusionist” role of the CPU allows a programmer write programs that are agnostic of other programs running in the system

**True**

1. Modern operating systems come with many utility services that are analogous to the “Glue” role of the OS

**True**

1. Networking service is not a core OS part, rather a common service included with most OS

**True**

1. Resource allocation and Isolation are not part of the core OS, rather common services included with OS

**False**

1. Efficiency is the secondary goal of an OS

**True**

Short Questions

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

**Multiprogramming is to run many programs in parallel on a multiple processor. Multiprogramming was introduced in third generation computers.**

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% | **33.33%** |
| Memory | 32.55% | **61.85%** |
| Disk | 33.33% | **66.67%** |
| Terminal | 50% | **100&** |

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.

**15 minutes**

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

**Deciding how much memory every program is allowed to use is an example of the referee role of the OS.**

1. [6 pts] Give an example of a “Reliable but Unavailable” system, and one for a “Unpredictable” system.

**A system that freezes and hangs regularly but doesn’t affect the user files is reliable but unavailable. An unpredictable system example is when the operating system runs normally, but the files are corrupted.**