

COSC 450 Operating System Mini-Test #1

1.

- Running state – a process is using CPU for a calculation
- Ready state – a process is waiting for CPU (short-term scheduler)
- Blocked state – a process is waiting for I/O finish
- Transaction 1 – a process need I/O
- Transaction 2 – a process time out
- Transaction 3 – short-term scheduler select a process to run
- Transaction 4 – a process finish I/O and ready to run

2.

- a) Similar job are collected, save in a magnetic tape, and implement one by one sequentially.
- b) jobs for processes (I/O jobs) are saved in a file and executed one by one (i.e. network printer)
- c) Several jobs are loaded into RAM and OS support pseudo-parallelism.
- d) Since threads are using same address space.
- e) Limited size of memory, big size processes.
- f) Mechanical component (device itself), electrical component (device controller) and device driver.
- g) Since fetch cycle is much slower than execute cycle by CPU, there are bottleneck.
- h) Save the address of next instruction.
- i) Process status, snapshot of CPU, scheduling information, memory management information I/O status information.
- j) Fetching an instruction to CPU, decode the instruction then execute the instruction.
- k)
 - a. Protection between jobs
 - b. Job scheduling
 - c. virtual memory
- l)

Since instruction cycle are three steps: fetch, encoding, and execute
- m) multiple monitors are connected to a host computer and each user are shared system
- n)
 - a. Process & Thread Management
 - b. Memory Management
 - c. File Management
 - d. Deadlock Management
 - e. Input/Output Management

- o) Since limited number resources which must be shared between processes.
- p) To achieve high reliability, OS is broken into small well-defined module. Only one module (Microkernel) run in kernel mode and the rest run as user mode.
- q)
 - Issue I/O command to devices
 - catch interrupts from each I/O devices
 - handle errors
- r) (0.5 pt.) What are four necessary conditions for a deadlock?
 - Mutual exclusion
 - Circular Wait
 - Hold and Wait
 - No Preemption