COSC 450 Operating System Mini-Test #1

1.

- Running state – a process is currently 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 blocked since it need wait for an I/O
- Transaction 2 – a process time term is over, need be selected again by CPU scheduler.
- Transaction 3 – Since CPU become available, CPU scheduler select a process in ready queue.
- Transaction 4 – I/O become available, a process move to ready queue and ready to run.

2.

a) similar jobs are collected and 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 in a process are using same address space.

e) to support multiprogramming, 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, 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)

Sol) Since instruction cycle are three steps: fetch, decoding, and execute

fetch	decode	execute		
	fetch	decode	execute	
		fetch	decode	execute

m) multiple terminals are connected to a host computer and each user on a terminal 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: process management, memory management and deadlock management) 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)

- Mutual exclusion
- Circular Wait
- Hold and Wait
- No Preemption

s)

- Keep track of which parts of memory are currently being used by which process
- Allocate and deallocate memory space as needed for each process
- Decide which processes are to be loaded into memory when memory space become available.

t)

- Free-space management
- Storage allocation
- Disk scheduling

u)

- OS stops what currently doing and immediately transfers execution to a fixed location where the service routine for the interrupt is located.

v)

- CPU detect interrupt from the interrupt-request line and it reads the interrupt number and jump to the interrupt handler routine

w)

- DMA

x)

- Due to heavy data transfer, bus becomes a bottleneck.

y)

- Increased latency when a CPU must access remote memory across the system interconnect, creating a possible performance penalty-