BITS, Pilani-Dubai Campus,
Academic city,Dubai
III rd Year First Semester 2007-08
Degree:B.E.(Hons) Branch:C.S.E

COURSE NO. : CS UC372 **COURSE TITLE : : Operating Systems**
Date : 3-1-08

Comprehensive exam (closed book) Total marks=60 (Closed book) Weightage=40%

Part-A answer all the questions- All questions carry equal marks (5 *2=10)

Q1. How condition variables of monitor are different from wait and signal methods of semaphore ? (4 marks)

Q2. With simple example explain how thread scheduling is done using non-preemptive scheduling?

Q3. Why the operating system has to maintain a processor state information for a process?

Q4.What is meant by page fault in virtual memory management?

Q5.Outline the steps involved in deadlock prevention.

Part_B answer all the questions-All questions carry equal marks (5 *10=50M)

Q1 a).Suppose that a disk drive has 5000 tracks numbered 0 to 4999. The drive is currently serving a request at cylinder 143 and the previous request was cylinder at 125. The queue of pending requests in FIFO order is
86, 1470, 913,1774,948,1509,1022,1750,130.Starting from the current track what is the total distance in terms of tracks covered by the Read/write head for the following disk scheduling policies (4+4)

1)FCFS
2)SCAN

b)What factors decide the rotational latency of a HDD ? (2M)

Q2. a) Outline how dynamic partition leads to holes in main memory of the system(3M)
b)with the help of diagrams outline how using appropriate page based virtual memory the above drawback can be overcome ?(7M)

Bounded Buffer (3)

```
Semaphore mutex = 1;  // mutual exclusion to shared set of buffers
Semaphore empty = N;  // count of empty buffers (all empty to start)
Semaphore full = 0;   // count of full buffers (none full to start)
```

```
producer {
 while (1) {
  Produce new resource;
  wait(empty); // wait for empty buffer
  wait(mutex); // lock buffer list
  Add resource to an empty buffer
  signal(mutex); // unlock buffer list
  signal(full);  // note a full buffer
 }
}
```

```
consumer {
 while (1) {
  wait(full);   // wait for a full buffer
  wait(mutex);  // lock buffer list
  Remove resource from a full buffer;
  signal(mutex); // unlock buffer list
  signal(empty); // note an empty buffer
  Consume resource
 }
}
```

Q3.a)W.R.T the above diagram outline clearly what will be the effect of interchanging the wait statements in consumer code by a programmer (3M)

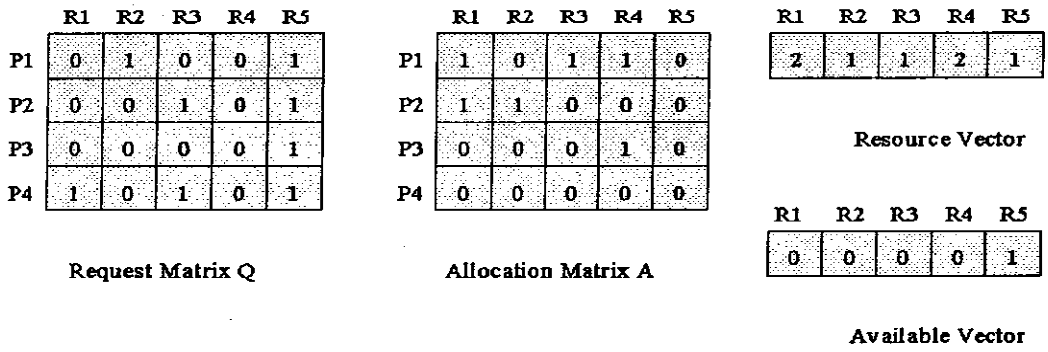

Figure 6.9 Example for Deadlock Detection

b)
Using appropriate steps identify whether any dead lock has happened in the above scenario.(3M)

c)What should be the minimum resources available with the resource vector to prevent happening of dead lock in the above scenario?(4M)

Q4.a)Outline the merits of DMA based data transfer compared to interrupt driven for transferring data from hard disk drive to memory ?(6)

b)Outline the advantages of using any two different types of buffers for data transfer between a process and an I/O device? (4M)

Q5. Suppose we have to implement a function to handle withdrawals from a bank account:

```
withdraw (account, amount) {
balance = get_balance(account);
balance = balance – amount;
put_balance(account, balance);
return balance;
}
```

Now suppose that you and your significant other share a bank account with a balance of $1000. Then you each go to separate ATM machines and simultaneously withdraw $100 from the account.(3+7)

a) Outline clearly what will happen if 2 threads corresponding to withdraw run without any co-ordination ?

b)How the above problem in a) can be solved using semaphore and monitor ?

BITS, Pilani-Dubai
Academic city,Dubai
III rd Year First Semester 2007-08
Degree:B.E.(Hons) Branch:C.S.E

COURSE NO. : CS UC372 **COURSE TITLE : : Operating Systems**
Date : 18-11-2007

Test 2 **Total marks=20 (Open book) Weightage=20%**

Q1.Justify whether a multithreaded solution using multiple user-level threads achieve better performance on a multiprocessor system than on a single processor system.(3M)

Q2.Under what circumstances does a multithreaded solution using multiple kernel threads provide better performance than a single threaded solution on a single-processor system.(3M)

Q3.Consider a multiprocessor system and multithreaded program written using the many-to many threading model. LET THE NUMBER OF USER-LEVEL THREADS IN THE PROGRAM BE MORE THAN THE NUMBER OF PROCESSORS IN THE SYSTEM. Discuss the performance implications of the following scenarios.(4M)

Q4. Explain why spin locks are not appropriate for single-processor systems yet are often used in multiprocessor systems compared to Disable interrupt type of lock.(4 M)

Q5.Outline the problem in using a lock having the following design in critical sections and how the problem can be resolved using an improved design of lock.(6M)

```
Struct lock {

int held=0;

}
Void acquire (lock)
{
While (lock->held);
Lock->held=1;
}
Void release(lock)
{
Lock->held=0;
}
```

In the above question proper explanation has to be provided about how Critical section can exist within the lock environtment and the way to avoid the above flaw using atomic instructions for reading the current status of lock variable and resetting the same appropriately.

BITS, Pilani-Dubai Campus, Knowledge Village,Dubai
III rd Year First Semester 2007-08
Degree:B.E.(Hons) Branch:C.S.E

COURSE NO. : CS UC372 **COURSE TITLE : : Operating Systems**
Date : 30-9-2007

Test 1 **Total marks=25 (Closed book) Weightage=20%**

Answer all the questions

Q1a).Outline how a multitasking environment is better than multiprogramming environment using appropriate case study.(5M)
During multitasking, choosing the correct quantum size is important to the effective operation of an operating system. Consider a single processor timesharing system that supports a large number of interactive users. Each time a process gets the processor , the interrupting clock is set to interrupt after the quantum expires. Assume a single quantum time interval for all processes on the system.
b) What would be the effect of setting the quantum at a very large value, say ten minutes ? (2 marks)
c)What if the quantum were set to a very small value , say a few processor cycles ? (3 marks)

Q2.Outline the effect of following hardware features on functioning of OS:

a) Hardware does not support kernel mode of operation (2 marks)

b) Hardware does not support timers(2 marks)

Q3.What is the main difference between ready state and blocked state of a process? Justify whether a process can go to running state directly from blocked state.(5M)

Q4. To which type of kernel design (monolithic or microkernel or both) do each of the following problems/advantages correspond? Why?

a) Resilience to errant kernel modules (2 marks)

b) Additional latencies (2 marks)

c) Each part of kernel has access to every other part (2 marks)

BITS, Pilani-Dubai,Campus

Academic City,Dubai

IIIrd Year FIRST Semester 2007-2008

Degree:B.E.(HONS) Branch:C.S.E

COURSE NO. : CS UC372 QUIZ1

COURSE TITLE: Operating systems

Time :15 mts Date : 1-10-2007 Total: 5 Marks

Answer all the questions(closed book) All questions carry equal marks

Answering and marking scheme

Q1. Outline how the OS switches the CPU among process A,process B,process C and a scheduler in main memory ?

Scheduler-process A-scheduler-Process B-scheduler process C .At the expiry of quantum interval or when a process requests for I/O the the scheduler will be invoked

Q2.What is meant by a system call and how it is handled by OS?

When user program want to access the I/O resource it generates a software interrupt Called Trap or system call. In response to that a ISR routine will be invoked which may access the I/O device .

Q3.What is the difference between symmetric multiprocessing and asymmetric multiprocessing?

In symmetric multiprocessing any CPU can run any part of the OS depending upon the need whereas in the case of asymmetric multiprocessing there is master CPU which will allocate work to the slave CPUS. Each slave will execute a pre determined task and thus lacks flexibility compared to symmetric multiprocessing.

Q4. Q4 In layered OS .what sort of abstraction is provided by virtual memory manager to the layers above that?

Above the virual memory manager users can develop programmes without worrying whether their programme is currently in the main memory or secondary storage.

Q5.How an operating system using microkernel approach give rise to a distributed operating system

In microkernel approach most of the OS modules like file server memory manager process manager run in user mode. Now we can form a distributed OS by relocating the various modules(running in user mode) in different machines and have a copy of kernel on every machine.

BITS, Pilani-Dubai Campus,
Academic city,Dubai
III rd Year First Semester 2007-08
Degree:B.E.(Hons) Branch:C.S.E

COURSE NO. : CS UC372 **COURSE TITLE : : Operating Systems**
Date : 6-11-07

Quiz2 **Total marks=5 (Closed book) Weightage=5%**

Answer all the questions .All questions carry equal marks

1. Why there is a need for thread pool?

2. What is meant by many-to many multithreading model?

3. How a thread_yield function can be used in pre-emptive and non-preemptive multithreading?

4. With the help of an example bring out the difference between mode switch and process switch in a multitasking architecture?

5. Suppose I want to implement
   Result= (x*y*z)/(a*b*c)
I want to perform concurrent operations for performing the numerator and

Denominator. Specify any two approaches with their relative merits and demerits for the same assuming that multiple CPUS are there. Make any valid assumption if necessary.

BITS, Pilani-Dubai Campus,
Academic city,Dubai
III rd Year First Semester 2007-08
Degree:B.E.(Hons) Branch:C.S.E

COURSE NO. : CS UC372 **COURSE TITLE : : Operating Systems**
Date : 6-11-07

Quiz2 **Total marks=5 (Closed book) Weightage=5%**

Answer all the questions .All questions carry equal marks

1. Why there is a need for thread pool?

   a) To avoid the time overhead of creating a new thread dynamically depending on the request
   b) To control the max number of threads supported by the system

2. What is meant by many-to many multithreading model?

Users create threads using thread library running in the user space.

Definite kernel threads time multiplex the user threads

3. How a thread_yield function can be used in pre-emptive and non-preemptive multithreading?
In pre –emptive multithreading thread yield function is called within ISR which is invoked forcibly at the end of thread quantum interval.
In the case of non-pre emptive multithreading a thread will itself call thread-yield when it wants to surrender the CPU

4. With the help of an example bring out the difference between mode switch and process switch in a multitasking architecture?

Suppose a process is running. Let the process be interrupted due to firing of an external random event. In that case mode switch takes place and at the end of ISR the process will be continued. Here the overhead is very less.
At regular quantum interval an interrupt happens and in respone to that Old process may be driven to ready and the new process will be in running state.This is called process

switch and leads to saving PCB for the outgoing and retrieving the PCB of the incoming process. Overhead is high in case 2.

5. Suppose I want to implement

Result= (x*y*z)/(a*b*c)

I want to perform concurrent operations for performing the numerator and Denominator. Specify any two approaches with their relative merits and demerits for the same assuming that multiple CPUS are there. Make any valid assumption if necessary.

Multithreading:

Allow thread1 to run the func1 for computing numerator and thread2 for computing denominator (func2). One of the thread can exchange its result and the final result can be obtained.

Muliple processes:

Allow two processes for computing func1 and func2. Here even though concurrency takes place overhead is very much because inter process communication takes long time. Further thread creation is less overhead instead of process creation..

BITS, Pilani-Dubai

International Academic city,Dubai

IIIrd Year FIRST Semester 2007-2008

Degree:B.E.(HONS) Branch:C.S.E

COURSE NO. : CS UC372

COURSE TITLE: Operating systems

Time :15 mts Date : 13-12-2007 Total:5 Marks Quiz3

Q1.What are the main advantages of semaphore compared to locks ?

Q2.What is the difference between concurrency and synchronization w.r.t processes P1 and P2?

Q3.W.r.t the below diagram justify whether a deadlock has happened or not ?

	R1	R2	R3	R4	R5
P1	0	1	0	0	1
P2	0	0	1	0	1
P3	0	0	0	0	1
P4	1	0	1	0	1

Request Matrix Q

	R1	R2	R3	R4	R5
P1	1	0	1	1	0
P2	1	1	0	0	0
P3	0	0	0	1	0
P4	0	0	0	0	0

Allocation Matrix A

R1	R2	R3	R4	R5
2	1	1	2	1

Resource Vector

R1	R2	R3	R4	R5
0	0	0	0	1

Available Vector

Figure 6.9 Example for Deadlock Detection

4.

Q4.

```
static int a;
void printtid()
{
a= mytid();
cout << a;
}
```

Consider the above programme.Assume that there is a CPU in the system. Now I want to spawn threads 1 and 2 to run the function printed( ). What is the problem I may face ?

Q5. With an example outline How the above problem can be resolved ?

BITS, Pilani-Dubai

International Academic city,Dubai

IIIrd Year FIRST Semester 2007-2008

Degree:B.E.(HONS) Branch:C.S.E

COURSE NO. : CS UC372

COURSE TITLE: Operating systems

Time :15 mts Date : 13-12-2007 Total:5 Marks Quiz3

Q1.What are the main advantages of semaphore compared to locks ?
1.Thread waiting for critical section need not waste its CPU cycles
2.There will not be any need for OS to disable interrupts

Q2.What is the difference between concurrency and synchronization w.r.t processes P1 and P2?

Concurrency is nothing but more than one thread run different procedures of a process simultaneously.On the other hand synchronization is nothing but a thread1 will start running a procedure after another thread completed some other activity.
Q3.W.r.t the below diagram justify whether a deadlock has happened or not ?

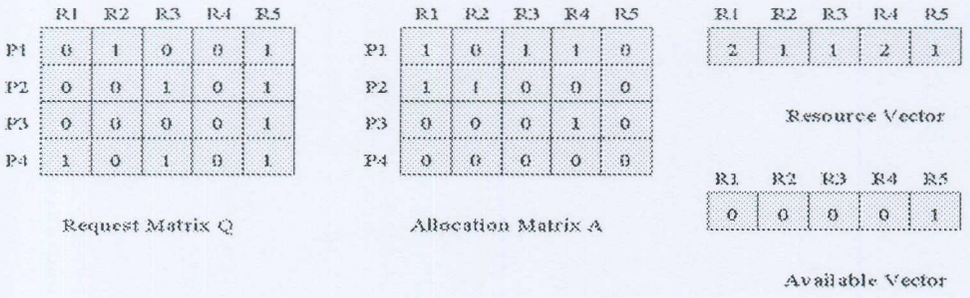

Figure 6.9 Example for Deadlock Detection

4.

Consider allocation matrix of P4. All are 0s.So p4 will not result in dead lock.
Consider request of P3. It can be satisfied by available vector..
Consider the request of P1 and P2.Their requests can not be handled by available vector. Thus dead lock happens because of P1 and P2

Q4.

```
static int a;
void printtid()
{
a= mytid();
cout << a;
}
```

Consider the above programme.Assume that there is a CPU in the system. Now I want to spawn threads 1 and 2 to run the function printed( ). What is the problem I may face ?

First thread let it stores its thread ID 100 in variable a. After that let thread switching takes place.now let thread2 has its execution for the entire function printed. In this case thread 2 will print its Id as 200 and store the same as 200 in variable a. When thread 1 resumes it will also print its Id as 200 instead of 100.

Q5. With an example outline How the above problem can be resolved ?

The above problem can be solved if the function PrintID( ) is declared as critical section and only one thread can enter and complete it. Then only another thread can enter. This is possible using locks, semaphore or monitor.

BITS, Pilani-Dubai,Campus

Academic city, Dubai

IV th Year FIRST Semester 2007

Degree:B.E.(HONS) Branch:C.S.E

COURSE NO. : CS UC372

COURSE TITLE: Operating sytems

Time :15 mts Date :17-12-2007 Marks: 5 Quiz4 Closed book

All questions carry equal marks

Q1. Why there is a need for virtual memory management ?

Q2. What is the difference between fixed and variable size partitions?

Q3.What is meant by compaction ?

Q4. How write protection enabled in VM management ?

Q5.What is the difference between external and internal fragmentation ?