CMR INSTITUTE OF TECHNOLOGY

USN | | | | | | | | | |

Internal Assessment Test II – November 2016

Sub: Operating Systems **Code:** 10CS53

Date: 03/11/2016 Duration: 90 mins Max Marks: 50 **Sem:** V **Branch:** 5 CSE A,B,C

Note: Answer FIVE full Questions, selecting at least TWO from each part. **Total marks: 50**

PART-A

#	QUESTIONS	Marks	OBE CO	OBE RBT
1	What is monitor? Explain the solution to classical dining philosopher's problem, using monitor.	10M	CO2	L2
2	Explain paging with simple example. Mention the problem faced by paging and how it is overcome by implementing TLB. Explain with supporting diagram and example.	10M	CO3	L2
3a	Explain the necessary conditions for deadlock to occur.	4M	CO2	L2
3b	Compare relative advantages and disadvantages of Semaphores and monitors.	6M	CO2	L4
4a	Explain internal and external fragmentation with example.	4M	CO3	L2
4b	For the following page references 5, 1, 2, 3, 4, 5, 3, 4, 1, 2, 3, 4, 5 with 3 frames and pure demand paging, find the total page faults using i) FIFO ii) Optimal iii) LRU page replacement algorithms.	6M	CO2	L3

PART-B

#	QUESTIONS	Marks	OBE CO	OBE RBT
5a	Explain the differences between i) paging and segmentation ii) physical and logical address space.	4M	CO3	L2
5b	Consider a paging hardware with a TLB. Assume that the entire page table and all the pages are in the physical memory. It takes 10 milliseconds to search the TLB and 80 milliseconds to access the physical memory. If the TLB hit ratio is 0.6, what is the effective memory access time(in milliseconds) ?	6M	CO3	L3
6a	What is resource allocation graph (RAG). Draw the RAG i) with deadlock ii) with a cycle but no deadlock.	4M	CO2	L2
6b	For the snapshot below, find the i) Need matrix ii) safe sequence if exists using Bankers Algorithm.	6M	CO2	L3

	Allocation A	Allocation B	Allocation C	Allocation D	Max A	Max B	Max C	Max D
P_1	0	0	1	2	0	0	1	2
P_2	1	0	0	0	1	7	5	0
P_3	1	3	5	4	2	3	5	6
P_4	0	6	3	2	0	6	5	2
P_5	0	0	1	4	0	6	5	6

Available:

A	B	C	D
1	5	2	0

7a	What is wait for graph? Explain how it is useful for detection of deadlock.	4M	CO2	L2
7b	Consider six memory partitions of size 200 KB, 400 KB, 600 KB, 500 KB, 300 KB, and 250 KB, where KB refers to kilobyte. These partitions need to be allotted to four processes of sizes 357 KB, 210 KB, 468 KB and 491 KB in that order. Which partitions are NOT allotted to any process using i) Best Fit ii) Worst Fit iii) First Fit ?	6M	CO3	L3
8a	What is thrashing? Explain working set model with example.	4M	CO3	L2
8b	Consider a system with byte-addressable memory, 32 bit logical addresses, 4 kilobyte page size and page table entries of 4 bytes each. i) What is the size of the page table in the system. ii) Find the logical address space, if two level page table is used by dividing memory into 8 sections.	6M	CO3	L3

Course Outcomes		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1:	Describe different functions, structures and design issues associated with OS.	3	-	-	-	-	-	-	-	-	-	-	2
CO2:	Demonstrate various process management concepts including scheduling, synchronizing and deadlocks.	3	2	-	-	-	-	-	-	-	-	-	2
CO3:	Explain memory management techniques like Paging, Segmentation and virtual memory.	3	-	-	-	-	-	-	-	-	-	-	-
CO4:	Explain the management techniques of file systems, protection and security.	3	-	-	-	-	-	-	-	-	-	-	-
CO5:	Demonstrate various secondary storage concepts including disk scheduling methods	3	-	-	-	-	-	-	-	-	-	-	-
CO6:	Explain the functionality with Linux operating system as case study.	3	-	-	-	-	-	-	-	-	-	-	-

Cognitive level	KEYWORDS
L1	List, define, tell, describe, identify, show, label, collect, examine, tabulate, quote, name, who, when, where, etc.
L2	summarize, describe, interpret, contrast, predict, associate, distinguish, estimate, differentiate, discuss, extend
L3	Apply, demonstrate, calculate, complete, illustrate, show, solve, examine, modify, relate, change, classify, experiment, discover.
L4	Analyze, separate, order, explain, connect, classify, arrange, divide, compare, select, explain, infer.
L5	Assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare, summarize.

PO1 - *Engineering knowledge*; PO2 - *Problem analysis*; PO3 - *Design/development of solutions*; PO4 - *Conduct investigations of complex problems*; PO5 - *Modern tool usage*; PO6 - *The Engineer and society*; PO7- *Environment and sustainability*; PO8 – *Ethics*; PO9 - *Individual and team work*; PO10 - *Communication*; PO11 - *Project management and finance*; PO12 - *Life-long learning*

CMR INSTITUTE OF TECHNOLOGY

USN | | | | | | | | | | |

Internal Assessment Test II – November 2016

Sub:	**Operating Systems**	**Code:**	10CS53
Date:	03/11/16 Duration: 90 mins Max Marks: 50 **Sem:** V	**Branch:**	5CS A,B, C

Note: Answer FIVE full Questions,selecting at least TWO from each part. **Total marks: 50**

Question No.	Description	Distribution of marks	Total Marks
1	Definition of Monitor State Dining philosophers problem Solution using monitors	1 3 6	10
2	Paging with example Problems in paging TLB explanation	5 1 4	10
3a	One line explanation of 4 necessary conditions	4	10
3b	Advantages and disadvantages of semaphores Advantages and disadvantages of Monitors	3 3	
4a	Internal fragmentation with example External fragmentation with example	2+2	10
4b	i) FIFO = 11 ii) Optimal = 8 iii) LRU = 11	2 2 2	
5a	Two differences between paging and segmentation Two differences between physical and logical addresses	2 2	10
5b	Solving problem with formula, steps and correct answer	6	
6a	Define RAG i) with deadlock graph ii) without deadlock with cycle graph	1 1.5 1.5	10
6b	i) Find need matrix ii) find safe sequence with steps and correct answer	2 4	
7a	Definition of Wait for graph How it is used to detect deadlock	1 3	10
7b	Solve problem i) Best Fit ii) Worst Fit iii) First Fit	 2 2 2	
8a	Define thrashing Explain working set model	1 3	10
8b	i) find size of page table ii) calculate logical address space	4 2	

Solutions

1 High-level abstraction that provides a convenient and effective mechanism for process synchronization. Only one process may be active within the monitor at a time.

```
Philosophers monitor DP
{
enum { THINKING; HUNGRY, EATING) state [5] ;
condition self [5];
void pickup (int i) {
state[i]            =
HUNGRY; test(i);
if (state[i] != EATING) self [i].wait;
}
void putdown (int i) {
state[i] = THINKING;
// test left and right
neighbors test((i + 4) % 5);
test((i + 1) % 5);
}
void test (int i) {
if ( (state[(i + 4) % 5] != EATING)
&& (state[i] == HUNGRY) &&
(state[(i + 1) % 5] != EATING) )
{ state[i] = EATING ;
self[i].signal () ;
}
}
initialization_code() {
for (int i = 0; i < 5; i++)
  state[i] = THINKING;
  }
  }
  ->Each philosopher I invokes the operations pickup()
  and putdown() in the following sequence:
  dp.pickup (i)
  EAT dp.putdown (i)
```

2

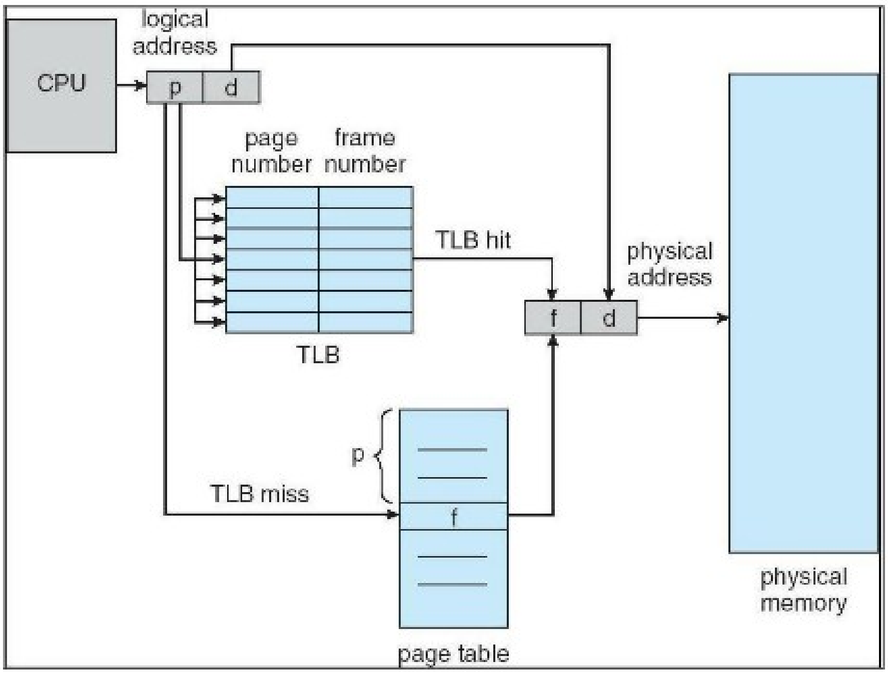
Paging:-

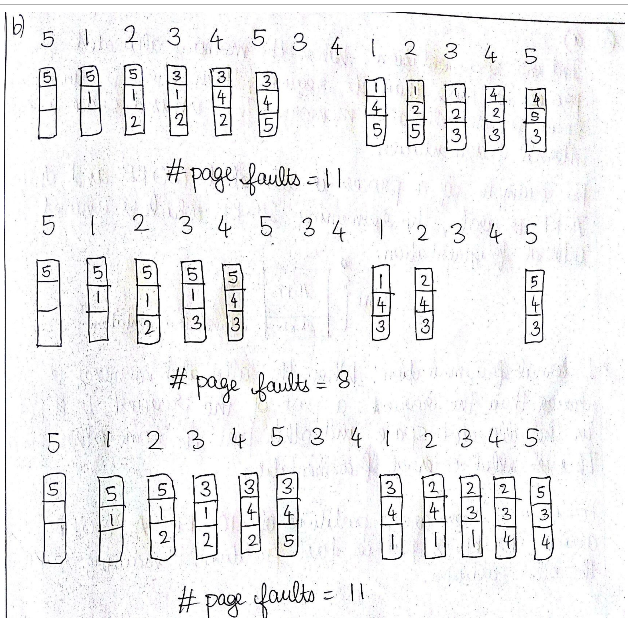
- Paging is a memory management scheme that permits the physical address space of a process to benon-contiguous. Support for paging is handled by hardware.
- It is used to avoid external fragmentation.
- Paging avoids the considerable problem of fitting the varying sized memory chunks on to the backingstore.
- When some code or date residing in main memory need to be swapped out, space must be found onbacking store.

Hardware Support for Paging:-

The hardware implementation of the page table can be done in several ways:-

1. The simplest method is that the page table is implemented as a set of dedicated registers. Theseregisters must be built with very high speed logic for making paging address translation . Everyaccessed memory must go through paging map. The use of registers for page table is satisfactory if thepage table is small.
2. If the page table is large then the use of registers is not visible. So the page table is kept in the mainmemory and a page table base register [PTBR] points to the page table. Changing the page tablerequires only one register which reduces the context switching type. The problem with this approach isthe time required to access memory location. To access a location [i] first we have to index the pagetable using PTBR offset. It gives the frame number which is combined with the page offset to produce the actual address. Thus we need two memory accesses for a byte.
3. The only solution is to use special, fast, lookup hardware cache called translation look aside buffer[TLB] or associative register.
4. TLB is built with associative register with high speed memory. Each register contains two paths akey and a value.



3a	Necessary Conditions:- deadlock situation can occur if the following 4 conditions occur simultaneously in a system:- 1. Mutual Exclusion:- Only one process must hold the resource at a time. If any other process requestsfor the resource, the requesting process must be delayed until the resource has been released. 2. Hold and Wait:- A process must be holding at least one resource and waiting to acquire additionalresources that are currently being held by the other process. 3. No Preemption:- RESOURCES CAN'T BE PREEMPTED I.E., ONLY THE PROCESS HOLDING THE RESOURCES mustrelease it after the process has completed its task. 4. Circular Wait:- A SET {P0,P1… … ..PN} OF WAITING PROCESS MUST EXIST SUCH THAT P0 IS WAITING for aresource i.e., held by P1, P1 is waiting for a resource i.e., held by P2. Pn-1 is waiting for resourceheld by process Pn and Pn is waiting for the resource i.e., held by P1.
3b	Both Monitors and Semaphores are used for the same purpose – thread synchronization. But, monitors are simpler to use than semaphores because they handle all of the details of lock acquisition and release. An application using semaphores has to release any locks a thread has acquired when the application terminates – this must be done by the application itself. If the application does not do this, then any other thread that needs the shared resource will not be able to proceed. Another difference when using semaphores is that every routine accessing a shared resource has to explicitly acquire a a lock before using the resource. This can be easily forgotten when coding the routines dealing with multithreading . Monitors, unlike semaphores, automatically acquire the necessary locks.
4a	• Memory fragmentation can be of two types:x Internal Fragmentation x External Fragmentation • ⑩ In Internal Fragmentation there is wasted space internal to a portion due to the fact that block of • data loaded is smaller than the partition. *Eg*:-If there is a block of 50kb and if the process requests 40kb and if the block is allocated to the process then there will be 10kb of memory left. • External Fragmentation exists when there is enough memory space exists to satisfy the request, but it not contiguous i.e., storage is fragmented in to large number of small holes. • External Fragmentation may be either minor or a major problem. • One solution for over-coming external fragmentation is <u>compaction</u>. The goal is to move all the free memory together to form a large block. Compaction is not possible always. If the relocation is static and is done at load time then compaction is not possible. Compaction is possible if the re-location is dynamic and done at execution time. • Another possible solution to the external fragmentation problem is to permit the logical address space of a process to be non-contiguous, thus allowing the process to be allocated physical memory whenever the latter is available.
4b	

5a	x The address generated by the CPU is called logical address or virtual address. x The address seen by the memory unit i.e., the one loaded in to the memory register is called the physical address. x Compile time a load time address binding methods generate some logical and physical address. x The execution time addressing binding generate different logical and physical addre x Set of logical address space generated by the programs is the logical address space. x Set of phys address corresponding to these logical addresses is the physical address space. x The mapping of virtual address to physical address during run time is done by the hardware **Segmentation:** x Program is divided in to variable sized segments. x User is responsible for dividing the program in to segments. x Segmentation is slower than paging. x Visible to user. x Eliminates internal fragmentation. x Suffers from external fragmentation. x Process or user segment number, offset to calculate absolute address. **Paging**: x Programs are divided in to fixed size pages. x Division is performed by the OS. x Paging is faster than segmentation. x Invisible to user. x Suffers from internal fragmentation. x No external fragmentation. x Process or user page number, offset to calculate absolute address.

5b

b) TLB hit ratio = 0.6

TLB miss ratio = 1 − 0.6 = 0.4

When it is a TLB hit, total access time is TLB access time + memory access time = 10 + 80 = 90 ms

When it is a TLB miss, total access time is TLB access time + page table access time + memory access time = 10+80+80 = 170 ms

Effective access time = 0.6 X 90 + 0.4 X 170

= 54 + 68

= 122 ms

6a Definition of RAG, i)Any RAG with Deadlock ii) Any RAG with cycle and no deadlock.

6b

b)

	Allocation A B C D	Max A B C D	Need A B C D	Available A B C D	
P1	0 0 1 2	0 0 1 2	0 0 0 0 ✓	~~1 5 2 0~~	~~1 5 2 0~~
P2	1 0 0 0	1 7 5 0	0 7 5 0 ✓	~~1 5 3 2~~	~~0 5 3 0~~
P3	1 3 5 4	2 3 5 6	1 0 0 2 ✓	~~2 8 8 6~~	~~2 1 3 6~~
P4	0 6 3 2	0 6 5 2	0 0 2 0	~~3 8 8 6~~	~~3 8 6 6~~
P5	0 0 1 4	0 6 5 6	0 6 4 2	~~3 14 11 8~~	~~3 8 7 6~~
				3 14 12 12	

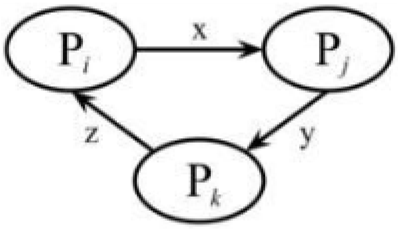
Safe sequence: < P1, P3, P2, P4, P5, >

Need

A	B	C	D
0	0	0	0
0	7	5	0
1	0	0	2
0	0	2	0
0	6	4	2

7a

A **wait-for graph** in computer science is a directed graph used for deadlock detection in operating Systems and relational database systems.

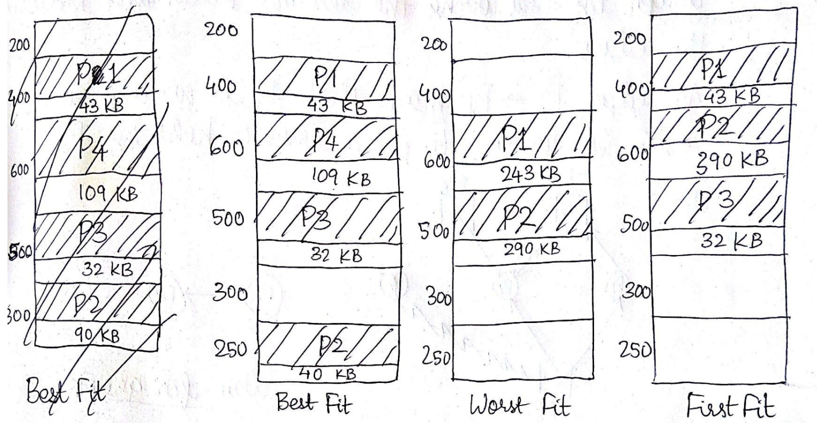


In computer science, a system that allows concurrent operation of multiple processes and locking of resources and which does not provide mechanisms to avoid or prevent deadlock must support a mechanism to detect deadlocks and an algorithm for recovering from them.

One such deadlock detection algorithm makes use of a wait-for graph to track which other processes a process is currently blocking on. In a wait-for graph, processes are represented as nodes, and an edge from process P_i to P_j implies P_j is holding a resource that P_i needs and thus P_i is waiting for P_j to release its lock on that resource. If the process is waiting for more than a single resource to become available (the trivial case), multiple edges may represent a conjunctive (and) or disjunctive (or) set of different resources or a certain number of equivalent resources from a collection. In the conjunctive case, graph cycles imply the possibility of a deadlock, whereas in the disjunctive case knots are indicative of deadlock possibility. There is no simple algorithm for detecting the possibility of deadlock in the final case.

7b

b) (P1, P2, P3, P4) – (357 KB, 210 KB, 468 KB, 491 KB)



~~i) In Best Fit~~

i) In Best Fit, 200 KB & 300 KB Partitions are NOT alloted to any ~~partition~~ process.

ii) In Worst Fit, 200 KB, 400 KB, 300 KB, 250 KB partitions are not allotted.

ii) In First Fit, 200 KB, 300 KB, 350 KB, partitions are NOT alloted

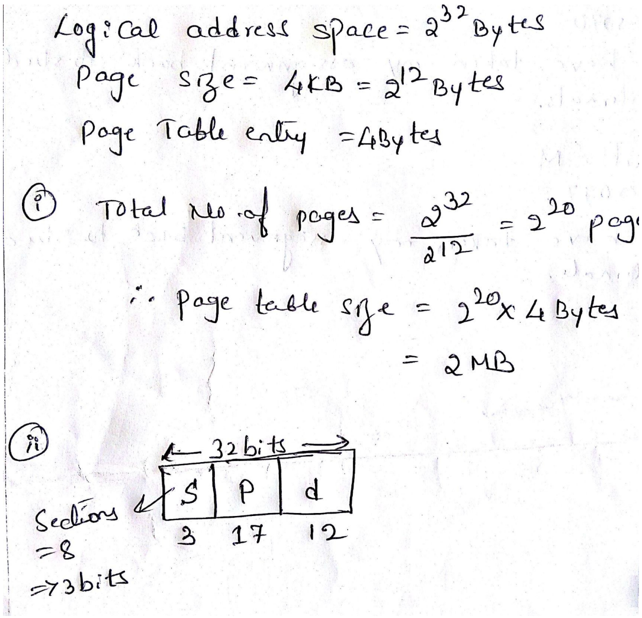
8a

If the number of frames allocated to a low-priority process falls below the minimum number required by the computer architecture then we suspend the process execution.A process is thrashing if it is spending more time in paging than executing.If the processes do not have enough number of frames, it will quickly page fault.During this it must replace some page that is not currently in useConsequently it quickly faults again and again.The process continues to fault, replacing pages for which it then faults and brings back. This high paging activity is called thrashing.The phenomenon of excessively moving pages back and forth b/w memory and secondary hasbeen called thrashing.

Working set model

Working set model algorithm uses the current memory requirements to determine the number of page frames to allocate to the process, an informal definition is “the collection of pages that a process is working with and which must be resident if the process to avoid thrashing”. The idea is to use the recent needs of a process to predict its future reader. The working set is an approximation of programs locality. Ex: given a sequence of memory reference, if the working set window size to memory references, then working set at time t1 is {1,2,5,6,7} and at t2 is changed to {3,4}

8b



Logical address space = 2^{32} Bytes
Page size = 4KB = 2^{12} Bytes
Page Table entry = 4Bytes

(i) Total No. of pages = $\frac{2^{32}}{2^{12}} = 2^{20}$ pages

∴ Page table size = $2^{20} \times 4$ Bytes
= 2MB

(ii)