(P-1)

Jagatbondhu Mondal.
ID:- 2121210051
CSI - 231 (Operating System Concept)
Final Examination - 2022

Ans to the Ques NO:- 1(A).

* Valid bit:- A bit of information that indicates whether data in a block is valid(1) or not(0) valid indeto that the associate page is in the logical address space.

* Invalid bit:- Invalid indicato that the associate page is not in logical address space.

Ans to the Ques NO:- 1(c)

physical Address:- physical Address is the actual address of the data inside the memory. The logical address is a virtual memory for its execution, the we never deals with the physical address.

1-b

Ans to the Qus. No – 1 (b)

Write down the architecture Segmentation:-

Segmentation architecture:

* logical address consists of a two tuple <segment-number, offset>.

* Segment table:- maps two-dimensional physical address, each table entry has:

⊛ base – Contains the starting physical address where the segment reside in memory.

⊛ limit :- Specifies the length of the segment

* Segment-table base register (STBR): point to the segment table's location in memory

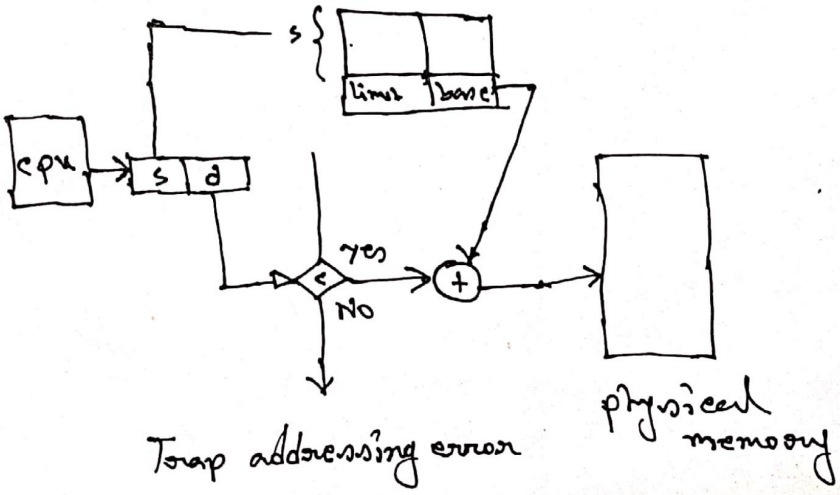
* Segment-table length register (STLR) point indicates number of segment used by a program.

Segment number S is legal if S<STLR

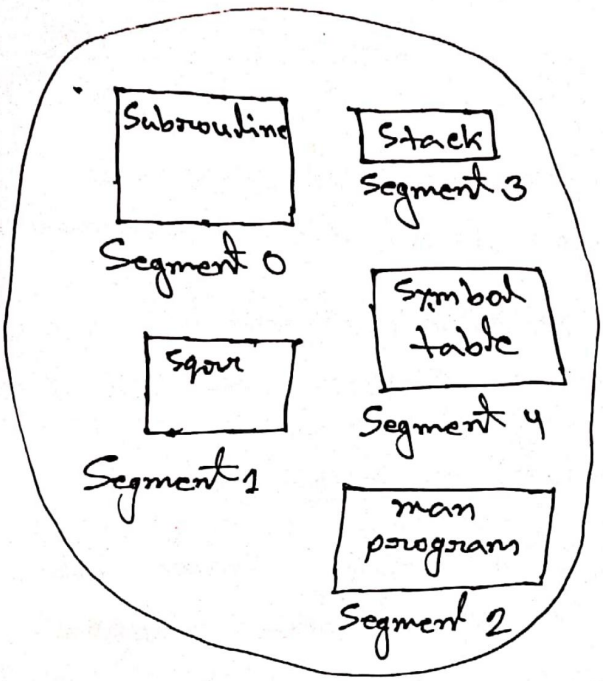
* Segmentation Architecture (Cont.)

- protection :- with each entry in Segment table associate :-
  - → validation bit = 0 ⇒ illegal Segment
  - → read / write / execute privileges
- protection bit associated with Segment: Code sharing occurs at Segment level
- Since Segment vary in length, memory allocation is a dinamic storage -allocation program.
- A Segmentation example is shown in the following diagram.

Segmentation Hardware

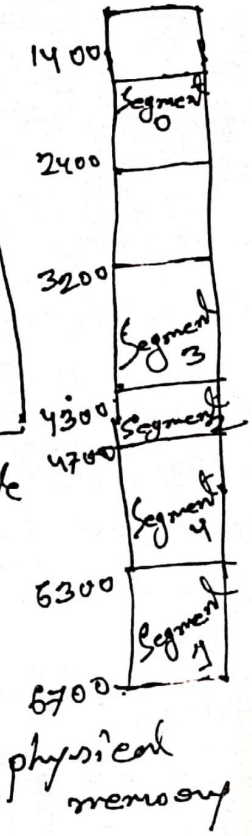


Example of Segmentation :-



	limit	base
0	1000	1400
1	400	6300
2	400	4300
3	1100	3200
4	1000	4700

Segment table



physical memory

3.a

Ans to the Qus No. 3 (a)

Characterization of Deadlock :-

Deadlock can arise if four Condition hold simultaneously.

* Mutual exclusion :- only one process at a time can use a resource.

* Hold and wait :- a process holding at least one resource is waiting to acquire additional resource held by other processes.

* No preemption : a resource can be released only voluntarily by the processes holding it. after that process has completed its task.

* Circular wait : there exisits a set $\{P_1 \ldots P_0\}$ of waiting processes such that P_0 is waiting for a resource that is held by P_1. P_1 is waiting for resource that is held by $P_2 \ldots P_{n-1}$ is waiting for a resource that is held by P_n and P_0 is waiting for resource that is held by P_0.

P-6

3-b

Ans to the Qus NO – 3 (b)

Re Deadlock prevention

Restrain the ways. request can be made:–

* Mutual Exclusion :– not required for sharable resources; must hold for non sharable resources.;

* Hold and wait :– must guarantee that whenever a process request a resource, it does not hold any other resources.

→ Requir process to request and be allocation all its resource before it begins execution. or allow process to request resources only when the process has non.

→ low resources utilization; Starvation possible.

* NO preemption :–

→ If a process that is holding some resources requests another resource that cannot be immediately allocated to it.

then all resources currently being held are released.

→ preempted resources are added to the list of resources for which the process is waiting

→ process will be restarted only when it can regain its old resources, as well as the new ones that it is requesting

Circular wait :- impose a total ordering of all resources types, and require that each process request resources in an increasing order of enumeration.

Ans to the Qus NO – 3 (c)

Safe State :- when a process requests an available resource. System must decide if immediately allocation leaves the System in a Safe state.

System is an Safe state if there exists a Safe sequence of all process

Sequence $<P_1, P_2 \cdots P_n>$ is safe state if for each P_i the resources that P_i can still request can be satisfied by currently available resources + resources held by all the P_j with $j<i$

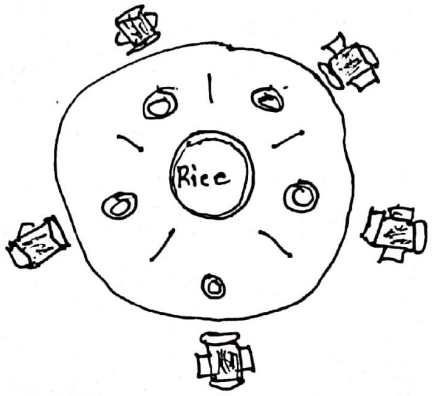
→ if P resources need are not immediately available, then P_i can wait until all P_j have finished.

→ when P_j is finished P_i can obtain needed resources, execute, return allocated resources, and terminate.

→ when P_i terminate P_{i+1} can obtain its needed resources, and so on.

Ans to the Qus NO - 4 (a)

Dining - philosopher problem



Shared data

* Bowl of rice (data set)
* Semaphore chopstick[5] initialized to 1

The structure of philosopher i:

```
do {
    wait (chopstick[i]);
    wait (chopStick[(i+1) % 5]);
        // eat
    Signal (chopstick[i]);
    Signal (chopStick[(i+1) % 5]);

        // Think

} while (True);
```

p-10

Ans to the Qus NO- 4 (b)

4-b

Bounded - Buffer problem.

* N buffers. each can hold one item
* Semaphore mutex initialized to the value 0
* Semaphore empty initialized to the value N
* The Structure of the producer process

```
do {
        // produce an item in nextp
    wait (empty);
    wait (mutex);
        // add the item to the buffer
    Signal (mutex);
    Signal (Full);
} while (True);
```

The Structure of the Consumer process

```
do {
```

wait (Full);
wait (mutex);

// remove an item from buffer to nextc.

Signal (mutex);
Signal (empty);

// Consumer the item in nextc

} while (True).

Ans to the Que NO - 4 (c)

4-c

Starvation :- Starvation is a severe deficiency in calorie energy intake, below the level needed to maintain an organism's life. It is the most extreem form of malnutrition, in humans. prolonged Starvation can cause permanent organ damage and eventually death.

The term manition refers to the Symptoms and effects of Starvation

5-a

Ans to the Qus No – 5 (a)

Multithreading issues: below we have mentioned a few issues related Multithreading well its an old saying. ALL good things come at a price.

* Thread cancellation :– Thread Cancellation means terminating a thread before it has finished working, there can be two approaches for this. one is Asynchronous Cancellation, which terminates the target thread immidiately, the other is Deffered Cancellation allows. the target thread to periodically Check if it should be cancelled.

* Signal Handling :– Signal are use in Unix System to notify. a process that a particular event has occerred now in which a multithreaded process receives a signal. to which thread it must be delivered ? It can be delivered to all, or a Signal Thread.

* fork() System call :- fork() is a systems call executed in the kernel through which a process creates a copy of itself. Now the problem in multithreading process is, if one thread forks, will the entire process be copied or not.?

* Security issues :- yes there can be Security issues because of extensive sharing of resources between multiple threads.

There are many other issues that you might face in a multithreaded process, but there are appropriate Solution available for them. pointing out some issues here was just to study both Sides of the coin.

5-b

Ans to the Qus NO - 5 (b)

Semaphore :- The classical definations of wait and Signal are:

* **wait :-** Decrements the value of its argument S. as soon as it would become non-negative (greater then or equal to 1)

* **Signal :-** Increments the value of its arguments S. as there is no more process blocked on the queue.

properties of Semaphore :-

* its sympte and always have a non-negative integer value.
* works with many processes
* Can have many different coritical Section with different Semaphores
* Each coritical Section has unique access Semaphores.
* Can permit multiple process in to the coritical Section at ones if desirable.

Ans to the Qus No - 5 (c)

⊞ worst fit :- worst fit allocates a process to the partition which is largest sufficient among the freely available in the main memory. if a large process comes at a latter stage, then memory will not have space to accommodate it.