Q:- In Deadlock prevention strategy do you think it is necessary to check either that Safe State exist or not? Give reason to support your answer?

Ans:- Deadlocks :-

Deadlocks possible when resources are shared between concurrent process.

- System can be depicted using a resource allocation graph.
- Resource-allocation graph may be used to check for deadlocks.

Deadlock ~~Ex~~ Prevention :-

Ensure that at least one of the necessary conditions does not hold

- Mutual Exclusion
  - Not required for sharable resources - (Read only files)
  - Must hold for non sharable resources.
- Cannot Prevent deadlocks by

denying mutual-exclusion condition (there are non sharable resources)

- Hold and wait
  - must gurantee that whenever a process requests a resources, it does not hold any other resources.
  - Requires process to request and be allocated all its resources before it begins execution.
  - System Calls requesting resources precede all others System Calls.
  - Allows process to request resources only when the process has none.
  - Low resource utilization, starvation possible.

Deadlock avoidance :-

Requires that the System has some additional apriori information ~~available~~ available.

- simplest and most useful model requires that each process declare the maximum number of resources of each type that it may need.

- Resource - allocation state is defined by the number of available and allocated resources, and the maximum demands of the process.

Safe state

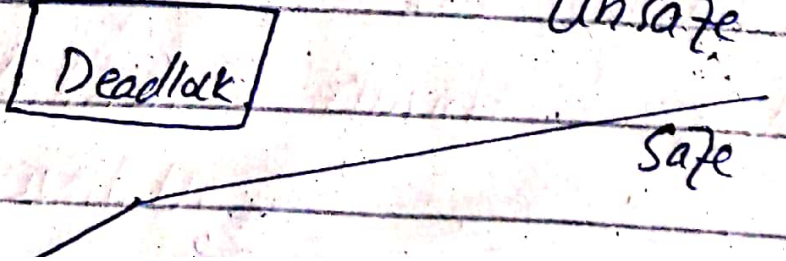
- When a process requests an unavailable resource, system must decide if ~~immediately~~ immediate allocation leaves the system in a safe state
- System is a safe state if there exists a sequence $< P_1, P_2, \dots P_n >$ of all the processes in the systems such ~~as~~ that for each $P_i$, the resources that $P_i$ can still requests can be satisfied by currently available resources + resources held by all the $P_i$
- That is
  - if $P_i$ resources needs are not immediately available, then $P_i$ can wait until all $P_i$ have finished.

- When $P_i$ is finished, $P_i$ can obtain needed resources, execute return allocated resources and terminate.
- When $P_i$ terminates $P_{i+1}$ can obtain its needed resources, and so on.

Basic Facts

- if a system is in a safe state ⇒ no deadlocks
- if a system is in unsafe state ⇒ possibility of deadlock
- Avoidance ⇒ ensure that a system will never enter an unsafe state.

Safe, unsafe, Deadlock State



Q2- Difference between dynamic loading and dynamic linking with the help of example

Ans ⇒ Dynamic loading :-

means loading the library (or any other binary for the matter) into the memory during load or run-time.

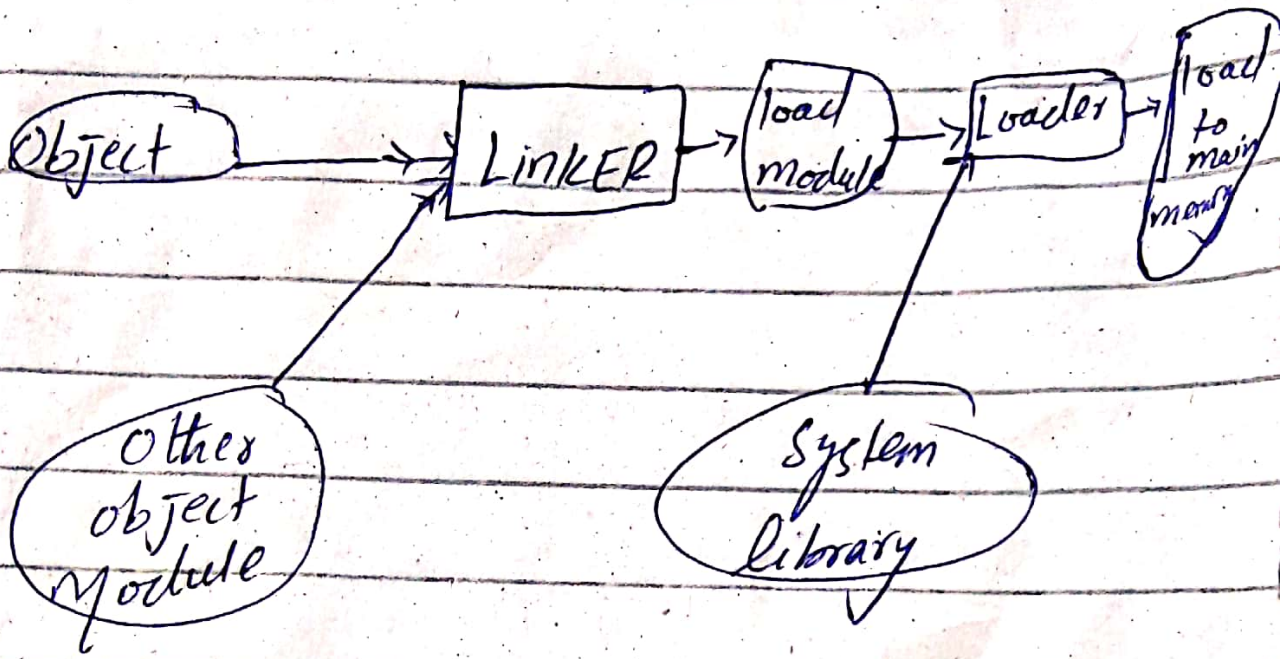
Dynamic loading can be imagined to be similar to plugins, that is an exe can actually execute before the dynamic loading happens (The dynamic loading for example can be created using loadlibrary call in C or C++)

Dynamic linking :-

refers to the linking that is done during load or run-time and not when the exe is created.

In case of dynamic linking the linker while creating the exe does minimal work. For the

dynamic linker to work it actually has to load the libraries too. Hence it's also called linking loader.



Q3: Which component of an operating System is best Suited to ensure fair, orderly and effiecent use of memory? Also identify more task managed by that component?

Ans:- The most Suitable Component of an operating System that Suited to ensure fair, Secure, orderly and efficient use of memory is memory managment System.

The task of memory managment includes keeping trade of used and free memory space as well as when where and how much memory to allocate and deallocate. it is also responsible for Swapping process in and out of main memory mangment is to ensure fair secure orderly and efficient use of a memory.

Q:4:- Difference between Symmetric and Asymmetric Encryption with the Help of Example?

Ans:- Symmetric encryption uses a single key that needs to be shared among the people who need to recieve the message while Asymmetrical encryption uses apair of public key and aprivate key to the decrypt messages when communicating.

- Symmetric encryption is an old technique while asymmetric encryption is relatively new.
- Asymmetric encryption was introduced to complement to the inherent problem of the need to share the key in Symmetrical encryption model eliminating the need to share they key by using apair of public-private keys.
- Asymmetric encryption takes relatively more than the Symmetric encryption.

Difference b/w Symmetric and Asymmetric

Symmetric Encryption

- Key for both encryption and Decryption.
- Symmetric encryption is fast in execution.
- DES, 3DES, AES, and RC4

The symmetric encryption is used for bulk data transmission.

Asymmetric Encryption.

- Key for encryption and decryption
- Asymmetric Encryption is slow in execution due to the high computational burden.
- Diffie-Hellman, RSA.

The symmetric encryption is often used for securing exchanging secret keys.

Q5:- Difference between Internal and External fragmentation? Why should they for avoided?

Ans:- Whenever a process is loaded or removed from the physical memory block, it creates a small hole in memory space which is called fragment. Due to fragmentation, the system fails in allocating the the contigous memory space to a process even thought it have the requested amount of memory but, in anon-contingous manner. The fragmentation is further classified into two categories internal and External fragmentation.

Both the internal and external classification affects data accessing speed of the system. They have a basic difference between them. i-e internal fragmentation.

Internal Fragmentation

Occurs when fixed sized memory are allocated to the process without conceming about the size of the proce and External fragmentation occurs when the processes are allocated memory dynamically.

~~Difference~~

Difference between internal and external fragmentation.

Internal fragmentation:

① Definition: When there is a difference between required memory ~~state~~ space vs alloted memory space. problem is termed as internal fragmentation.

② Memory Block size:-

Internal fragmentation occurs when alloted memory blocks are of fixed size.

③ occurance:

Internal fragmentation occurs when process needs more space than the size of alloted memory

or use less space.

④ Solution :-

Best fit Block search is the solution for internal fragmentation.

External fragmentation :-

① Definition :-

When there are small and non-contiguous memory block which cannot be assigned to any process, the problem is termed as External fragmentation.

② Memory Block size :-

External fragmentation occurs when alloted memory blocks are of varying size.

③ occurance :-

External fragmentation occurs when a process is removed from the main memory.

④ Solution :-

Compaction is the solution for external fragmentation.

Q6: List and describe four memory allocation

Ans 6

1. First Fit
2. Best Fit
3. Worst Fit
4. Next Fit

First Fit :-

In the first fit approach is to allocate the first free partition or hole large enough which can accommodate the process. it finishes after finding the first suitable free partition.

Advantages

Fastest algorithm because it searches as little as possible.

Disadvantages

The remaining unused memory areas left after allocation become waste if it is too smaller Thus request for larger memory requirements cannot be accomplished.

② Best fit:-

The best fit deals with allocating the small free partition which meets the requirements of the requesting process. This algorithm first searches the entire list of free partition and considers the smallest hole that is adequate. It then tries to find a hole which is close to actual process size needed.

③ Worst fit:-

In worst fit approach is to locate largest available free partition so that the partition left will do big enough to be useful. It is the reverse of best fit.

④ Next fit:-

Next fit is a modified

version of first fit. it begins as first fit to find a free partition. when called next time it starts searching from where it left off not from the beginning.

(1) Single continguous allocation.
(2) partitioned allocation.
(3) Paged memory allocation.
(4) segmented memory allocation.

(1) single continguous allocation.
Smallest allocation method used by MS-DOS, All memory is available to aprocess.

(2) partitioned allocation:-
Memory is divided in different blocks or partitions. Each process is allocated according to the requirment.

(3) Paged allocation:-

Memory is divided into fixed sized units called page frames used in a virtual memory environment.

(4) Segmented memory managment:-

Memory is divided into different segments. In this managment allocated memory doesn't have to be continguous. Most of operating system (for example windows and linux) are use segmentation with paging. A process is divided into segments and individual segments have pages

Q7:- Which is the context switch overhead of a user-level threading as compared to the overhead for processes? Explain.

Ans:- One might that in general process are more flexiable than threads for example processes are controlled independent by the OS, meaning that if one crashes it will not effect other processes However, processes require explict communication using either message passing or shared memory which may add ~~overhead or shared memory which~~ overhead Since it requires support from the OS kernal, using threads within a process allows them all to have the same address space simplifying communicati between threads However, threads have their own problems because they communicate through shared memory

They must run on some machine and save.

Name :- Syed Ihtisham ul Haq Bacha

ID :- 14804

Dept :- Software Engineering

Smster :- 4th

paper :- Operating System

Submitted to :-

Daud Khan Khalil