NAME : JANJAID KHALIQ
ID : 15386
PAPER : OPERATING SYSTEMS
TEACHER : SIR DAUD
EXAM : FINAL TERM
DATE : 22/06/20

Q2 :- Differentiate between Dynamic Loading and Dynamic Linking with the help of examples.

Ans :-

Dynamic Loading :-

- With dynamic loading, a routine is not loaded until it is called.
- All routines are kept on disk in a relocatable load format.
- The main program is loaded into memory and is executed.
- When a routine needs to call another one, the calling routine first checks to see whether the other routine has been loaded.
- If not, the relocatable linking loader is called to load the desired routine into memory.
- Then, control is passed to the newly loaded routine.

Example :-

Suppose our Program that is to be executed consist of various modules of course its not wise to load all the modules into main memory together at once (in some cases it might not be even possible because of limited main memory). So basically what we do here is we load the main module first and then during execution we load some other module only when its required and the execution cannot proceed further without loading it.

Dynamic Linking :-

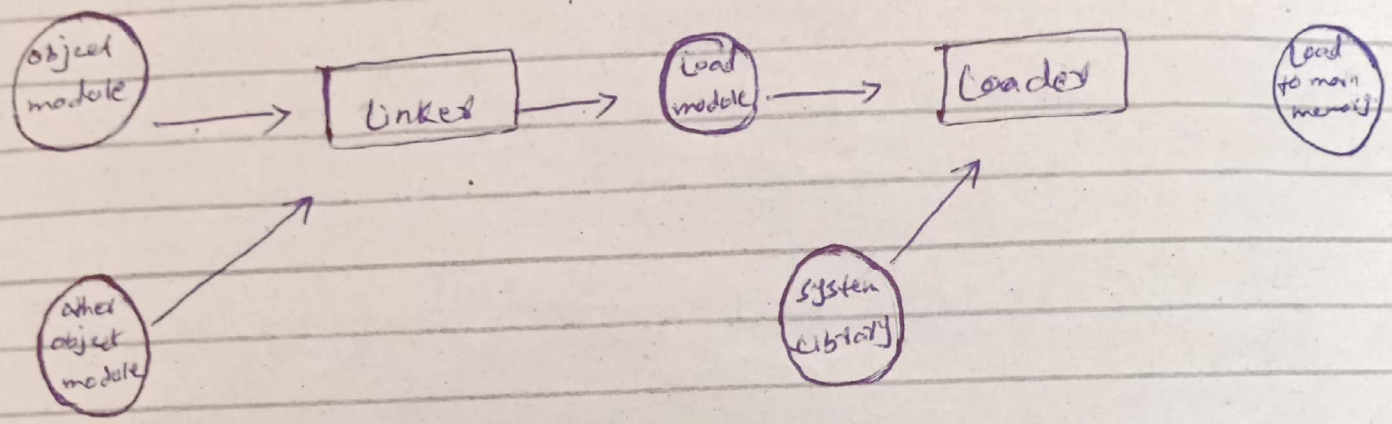
Establishing the linking between all the modules or all the functions of the Program in order to continue the Program execution is called linking.

Example :-

Suppose our Program has some functions whose defination is present in some system library we do know the header file only consist of declaration of function and not definitions. So during execution when the function

gets called we load that system library into main memory and link the function call inside our program with function defination inside system library.

Diagram :-



Q4 :- Differentiate between Symmetric and A-Symmetric encryption with the help of example.

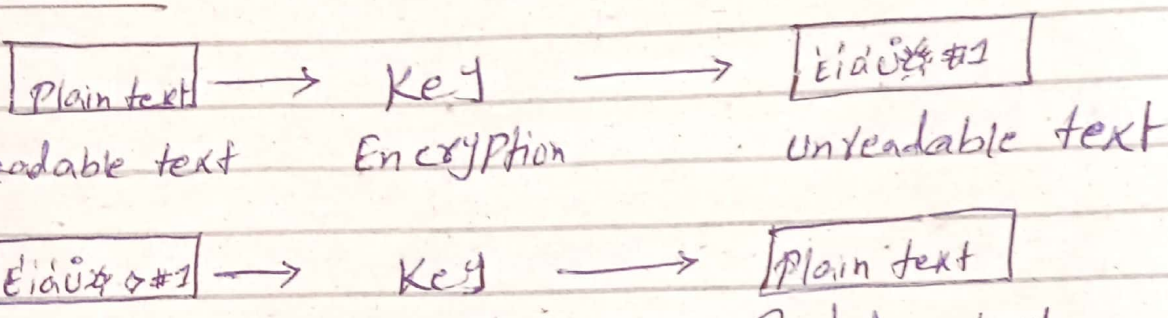
ANS :-

Symmetric encryption :-

Symmetric encryption is the Process of converting readable data unreadable format and converting it back to readable format using same key. Symmetric encryption algorithms use the same key for encryption and decryption. The key must be exchanged so that both the data sender and the data recipient can access the plaintext data. The Plain text (Readable Text) is converted

to ciphertext (unreadable text) using a key and at the receiving side the same key is used to convert back the ciphertext (unreadable text) to plain text (Readable text).

Example :-



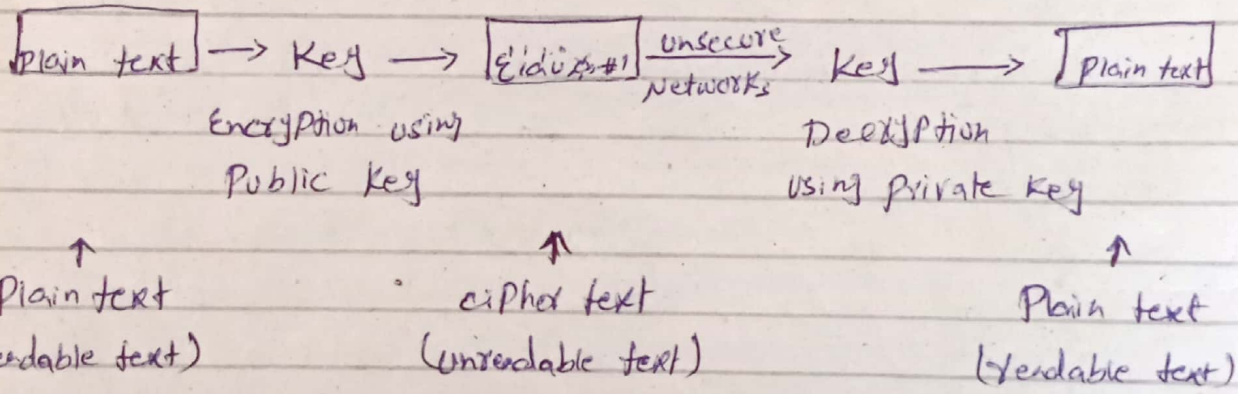
A-Symmetric Encryption :

A-Symmetric encryption increases the security of the encryption process utilizing two seperate but mathematically related keys known as a public key and a Private key. A-Symmetric encryption algorithms use a key mathematically related key pair for encryption and decryption. one key of the key pair is known as the public key and other one is private key.

The Private key is possessed only by the user or computer that generates the key pair. The public key can be distributed to any person who wishes to send encrypted data to the Private key holder. it is impossible to compute

the Private key if you known the Public key. Hence it is safe to Publish the Public key.
if the Public key is used for Encryption the associated Private key is used for decryption.
If the Private key is used for Encryption the associated Public key is used for decryption.

Example :-



Q1 :- Describe the difference between external and Internal fragmentation. why should they be avoided?

ANS :-

External fragmentation :-

External fragmentation exists when total free memory is enough for the new process but it's not contiguous and can't satisfy the request. Storage is fragmented into small holes.

Internal fragmentation :-

Internal fragmentation is the area occupied by a process but cannot be used by the process. This space is unusable by the system until the process release the space.

why should they avoided ?

Internal fragmentation (memory in a partition not used by its own process - but not available to other process, also external fragmentation. two small partitions lefts, but one big job partition 1, 3 used and does't fit in 2 or 4).
Partition size fixed at bootstrap.
Program loaded into partition that fits best. when context switch load base and limit registers into partitions.

Q1 : In deadlock Prevention Strategy do you think it is necessary to check that either safe state exists or not ? Give reason to support your Answer.

Answer : YES, it is necessary to check that either safe state exists or not in deadlock Prevention strategy because A state is safe if there one can find an ordering of the processes

Such that: if the processes are run in this order, they will terminate (assuming none exceeds its claim).
Give an example of all four possibilites.
A State that is

- Safe and deadlock - not possible
- Safe and not deadlock
- Not Safe and deadlock
- Not Safe and not deadlock - interesting.

A manager can determine if a state is safe

- Since the manager know all the claims, it can determine the maximum amount of additional resources each process can request.
- The manager knows how many units of each resource it has left.

Q6:- List and describle the four memory allocation algorithms covered in leatures. Which two of the four are more commonly used in Practice?

Answer:-

Four memory allocation algorithms are listed below.

1) First fit
2) Next fit
3) Best fit
4) Worst fit

1) First fit :-

- Search starts from the starting location of the memory.
- First available hole that is large enough to hold the process is selected for allocation.
- The hole is then broken up into two pieces one for process and another for unused memory.
- Fastest algorithm because it searches as little as possible.
- Memory loss is higher, as very large hole may be selected for small process.

2) Next fit :-

- it works in the same way as first fit, except that it keeps the track of where it is whenever it finds a suitable hole.
- The next time when it is called to find a hole, it starts searching the list from the place where it left off last time.
- Search time is smaller.
- Memory manager must have to keep track of last allotted hole to process.

3) Best fit :-

- entire memory is searched here.
- The smallest hole, which is large enough to hold the process, is selected for allocation.
- Search time is high, as it searches entire memory every time.
- memory loss is less.

4) worst fit :-

- Entire memory is searched here also. The largest hole which is largesh enough to hold the process, is selected for allocation.
- Search time is high, as it searches entire memory every time.
- This algorithm can be used only with dynamic Partitioning.

Two of the four are more commonly used in practice is Best fit and First fit.

Q7:- Why is the Context Switch overhead of a User-level as compared to the overhead for Processes? Explain.

Ans:-

Context switch:-

The Process of Saving the Context of one Process and loading the context of another Process is known as Context switching. In simple terms, it is like loading and unloading the Process from running State to ready state.

Why as compared to the overhead for Processes:-

- Context switch time is pure overhead, because the system does no useful work while switching.
- Its speed varies from machine to machine, depending on the memory speed the number of registers that must be copied, and the existence of special instructions (such as a single instructions to load or store all registers).
- Typical speeds are a few milliseconds.

Q3 :- Which Components of an Operating System is best Suited to ensure fair, secure, orderly, and efficient use of memory? Also Identify some more tasks managed by that Components :-

Ans :-

Kernal is best Suited to ensure fair, secure, orderly and efficient use of memory and

- A kernal is a core Program responsible for hardware control, memory and Process management and inter Process Communication, address Space.