Name = Farhan Akhtar

ID = 15554

Subject = OPerating System

Semester = 3rd Semester

Program = Bs(S.E)

T. Name = Sir Daud.

Q1 = In deadlock prevention Strategy do you think it is necessary to check that either Safe State exists or not? Give reason.

Ans :— Yes, it is necessary to check that either Safe State exists or not in deddlock prevention because when we disallow deadlock by

Setting "safe states" it means process completion is always guaranteed.

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

Ans :- Dynamic Loading :-

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

Dynamic loading can be Imagined to be Similar to Plugin, that is an exe can actually execute before the dynamic loading happens.

Dynamic loading refers to mapping an executable or library into a process's memory after it has started.

Example:—

When we write a program in C language and we are going to execute the same program. given below example:

```
main( )
    if{          }
    move( )
    else
    run( )
```

First the loader will only load main function to the main memory. Then if Statement is true, it will call the move( ) function or when the if Statment is false it will call the run function.

As we noticed that using dynamic loading we save the Space in the main memory Instead of loading the whole program to the main memory.

Hence the dynamic loading increase the degree of multiprogramming.

⇒ Dynamic linking :—

Every dynamically linked program contains a small statically linked function that is called when the program starts.

This static function only maps the link library in to memory and runs the code that the functions contains.

The link library determines that what are all the dynamic libraries which the program requires along with names of variables and functions needed from these libraries by reading the information contained in sections of the library.

Q3 :— Which Component 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 Component.

Ans :— The most Suitable Component of an operating System that suited to ensure fair and secure use of memory is memory managment. Memory managment Includes keeping tracks of used and free memory space as well as when where and how much memory to allocate and deallocate.

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

Ans :- Symmetric Encryption :—

In Symmetric Encryption the message is ~~ency~~ encrypted by using a key and the same key is used to decrypt the message which makes it easy to use but less secure. It also requires a safe method to transfer the key from one party to another.

Examples :—

⇒ Practical example is Sending Verification to the mobile to access the data.

* 3DES * AES

⇒ Asymmetric Encryption :—

Asymmetic Encryption is based on Public and

Private key encryption technique. It uses two different key to encrypt and decrypt the message. It is more secure than Symmetric key encryption techneque but it much slower.

Examples :— * Deffie-Hellman
* Algorithums

Differences b/w Symmetric and Asymmetric Encryption

Symmetric Encryption	Asymmetric Encryption
* It only requires a single key for both encryption & decryption	* It requires two key one to encrypt and otherone to decrypt.
* The encryption process is very fast.	* It is used to transfer small amount of data.
* It is used when a large amount of data is required to transfer.	

* The size of cipher text is same or smaller than the original plan text.	* The size of cipher text is same or larger than the original plain text.
* It only provides confidentiality.	* It provides confidentiality, authenticity and Non-Repudiation.
* Examples :- 3DES, AES, DES and RC4	* Examples :- Diffie-Hellman, ECC, El Gamal, DSA and RSA.

Q5:— Describe the differences between external and internal fragmentation. Why should they be avoided.

Ans:— Differences between the external and internal fragmentation.

Internal Fragmentation	External Fragmentation
* In internal fragmentation fixed-sized memory, blocks square measure appointed to process.	* In external fragmentation variable sized memory blocks square measure to appointed to method.
* The solution of internal fragmentation is best-fit block.	* Solution of E.F is compaction, Paging and Segmentation.
* Internal fragmentation happens when the method or process is larger than memory.	* External fragmentation happens when the method or process is removed.

* Internal fragmentation occurs when memory is divided into fixed Sized Partition.	* E.F occurs when memory is divided into variable size partition based on size of Process.
* The difference b/w memory allocated and required Space or memory is called Internal fragmentation.	* The unused Spaces formed b/w non-contiguous memory fragments are too small to serve a new process is called E. fragmentation.

Q6 :- List and describe the four memory allocation algorithms covered in lectures. which two of the four are more commonly used in practice. ?

Ans :- The four memory allocation algorithms that are covered given below :—

* First-Fit.
* Next-Fit.
* Worst-Fit.
* Best-Fit.

* First-Fit :—

First-Fit allocates in to the first available gap found of adequate size, starting from the beginning of memory. In the linked list of availible memory address, we place the data in the first entry that will fits its data.

Its aim to minimise the amount of searching, but leads to external fragmentation later on.

* Next-Fit :—

It allocates in to the first availible gap from resuming to search from the last allocation. Scan memory region list from print of last allocation to next fit breaks up large block at the end of memory.

* Worst-Fit :—

Traverse the memory and gives the partitions as large spaces as possible to leave usable fragments left over. Needs to search the complete list and such is a poor performer.

* Best-Fit :—

pick up the closest free region in the

entire list leaves Small unusable regions and Slower due to Searching of entire list.

⇒ Best-Fit and Next-Fit are most commonly used in practice.

Q7 :-

Ans :- User-level threads Implement in user-level libraties, rather than via System calls, So thread Swithing does not need to call the operating System and to Cause Interruption to the ~~k~~kernel. In fact the kernel know nothing about user-level threads and manges them as If they were Single threaded process.

The most abvious advantage of this techmequie is that a user level threads package can be Implemenfeel on an OS. that does not Suppost thread.

1 :- user level threads do not require modification to OS.

2 :- Simple representation

3 :- Simple managment

4 :- Fast and efficient : Thread Swithing is not expensive than procedure call.