NAME:

Muhammad Ali KHAN

Reg No:

16550

FINAL PAPER Operating System

Q1: Differentiate Between a process and thread with example.

PROCESS:- A process is the execution of a program that allows you to perform the appropriate actions specified in a program. It can be defined as an execution unit where a program runs. The OS helps you to create, schedule, and terminates the processes which is used by CPU. The other processes created by the main processes are called child process.
Example: If you open up two browser windows then you have two processes, even though they are runing the same program.

THREAD:- Thread is an execution unit that is part of process. A process can have multiple threads, all executing at the same time. It is a unit of execution in concurrent programming. A thread is lightweight and can be managed indepently by a scheduler. It helps you to improve the application performance using parallelism.

Muhammad Ali Khan
Reg # 16550

Parameter	Process	Thread
1) Definition	Process means a Programs is in execution	Thread means a segment of a process.
2) Lightweight	The process is not Light weight.	Threads are Lightweight
3) Creation time	It takes mor time for creation.	It takes less time for creation.
4) Resource	Process consume mor resource.	Thread consume fewer resource.
5) Memory	The process is mostly isolated.	Threads share memory.
6) Termination Time	The process takes more time to terminat	Threads take less time to terminat.

②

Q2: List and discuss few types of thread?

1: USER LEVEL THREAD (ULT):- Is implemented in the user level library, they are not created using the system calls, Thread switching doesn't need to call OS and to cause interrupt to kernel. Kernel does not know about the user level thread and manages them as if they were single threaded process.

ADVANTAGES Of (ULT):-

- ★ Can be implemented on an OS that doesn't support multithread.
- ★ Simple representation since thread has only program counter, registeo set, stack space.
- ★ Simple to create since no intervention of kernel.

DISADVATAGES of ULT:-

- No or less co-ordination among the threads and kernel.
- If one thread causes a page fault the entire process blocks.

Muhammad Ali khan.
Reg # 6550.

2: KERNEL LEVEL THREAD (KLT):-

Kernel knows and manages the threads. Instead of thread table in each proce a kernel itself has a thread table (a master one) that keeps tracks of all threads in the system. In adition kernel also maintains the traditional process table to keep the track processes. OS kernel poovides system call to create and manage threads.

* ADVANTAGES OF (KLT):-
* Since kernel has full knowledge about the threads in the system, scheduler may decide to give more time to processes having large number of threads.
* Good for applications that frequently block.

- DISADVANTAGES OF (KLT):-
- slow and infecient.
- It requires thread control block so it is an overthread.

(7)

Q3: What is a deadlock? In what situation it occurs in an OS.

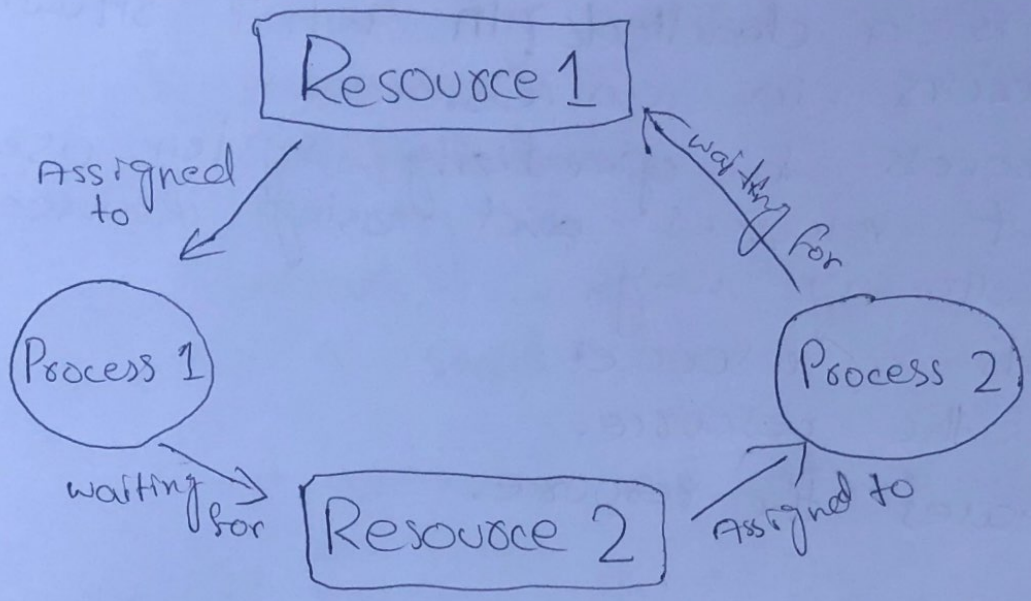
Ans: A process in operating system uses different resources and using resources in following way.

1) Requests a resource
2) Use the resource.
3) Releases the resource.

Deadlock: is a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process. Consider an example when two trains are coming toward each other on same track and there is only one track, none of the train can move once they are in front of each other. Similar situation occurs in operating systems when there are two or more processes hold some resources and wait for resources held by other(s).

For example: in the below table Process 1 is holding Resource 1 and waiting for resource 2 which is acquired by process 2 is waiting for resource 1.

Muhammad Ali Khan
Reg # 16550



Deadlock can arise if following four condition hold simultaneously (Necessary conditions).

- Mutual Exclusion: One or more than one resource are non-sharable (only one process can use at a time).
- Hold and Wait: A process is holding at least one resource and waiting for resources.
- No Preemption: A resource cannot be taken from a process unless the process releases the resource.
- Circular Wait: A set of proceses are waiting for each other in circular form.

★ In what situation it occur in OS?

In an operating system, a deadlock occurs when a process or thread enters a waiting state because a requested system resource is held by another waiting process, which in turn is waiting for another

③

resource held by another waiting process. If a process is unable to change its state indefinitely because the resources requested by it are being used by another waiting process, then the system is said to be in a deadlock.

Muhammad Ali Khan
Reg # 16550

Q4: Discuss a solution to the critical-section problem must satisfy the three requirements.

Ans: A critical section is a code segment that accesses shared variable and has to be executed as an atomic action. It means that in a group of cooperating processes, at a given point of time, only one process must be executing its critical section. If any other process also wants to execute its critical section, it must wait until the first one finishes.

1: MUTUAL EXCLUSION: Out of a group of cooperating processes, only one process can be in its critical section at a given point of time.

2: PROGRESS: If no process is in its critical section, and if one or more threads want to execute their critical section then any one of these threads must be allowed to get into its critical section.

3: BOUNDED WAITING: After a process makes a request for getting into its critical section, there is a limit for how many other processes can get into their critical section, before

Muhammad Ali Khan
Reg#16550

this process's requests is granted. So after the limit is reached, system must grant the process permission to get into its critical section.

Q5: Differentiate between dynamic loading and dynamic linking with example.

*. Dynamic Loading: system library or other routine is loading during run-time and it is not supported by OS. Suppose our program that is to be executed consist of various modules. Of course it's 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 it required and the execution cannot proceed further without loading it.

*. Dynamic Linking: system library or other routine is linked during run-time and it is supported by OS. Suppose our program has some function whose definition is present in some system. library. We do know the header file only consists of declaration of functions 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 the function definition inside system library.

1. Both dynamic loading and linking happen at ron time, and load whatever they need into memory.
2. The key different is that dynamic loading checks if the routine was loaded by the loader while dynamic linking checks if the routine is in the memory.
3. Therefore, for dynamic linking, there is only one copy of the library code in the memory, which may be not true for dyaamic loading. That's why dynamic linking needs OS support. to check the memory of other processes. This feature is very important for language subroutine libraries, which are shared by many programs.

muhammad Ali khan
Reg # 16550

Q6. Write your understanding about logical Vs physical address space?

Ans: An address generated by the CPU is a logical address whereas address actually available on memory unit is a physical address. Logical address is also known as virtual address.

Virtual and physical address are the same in compile-time and load-time address-binding scheme.

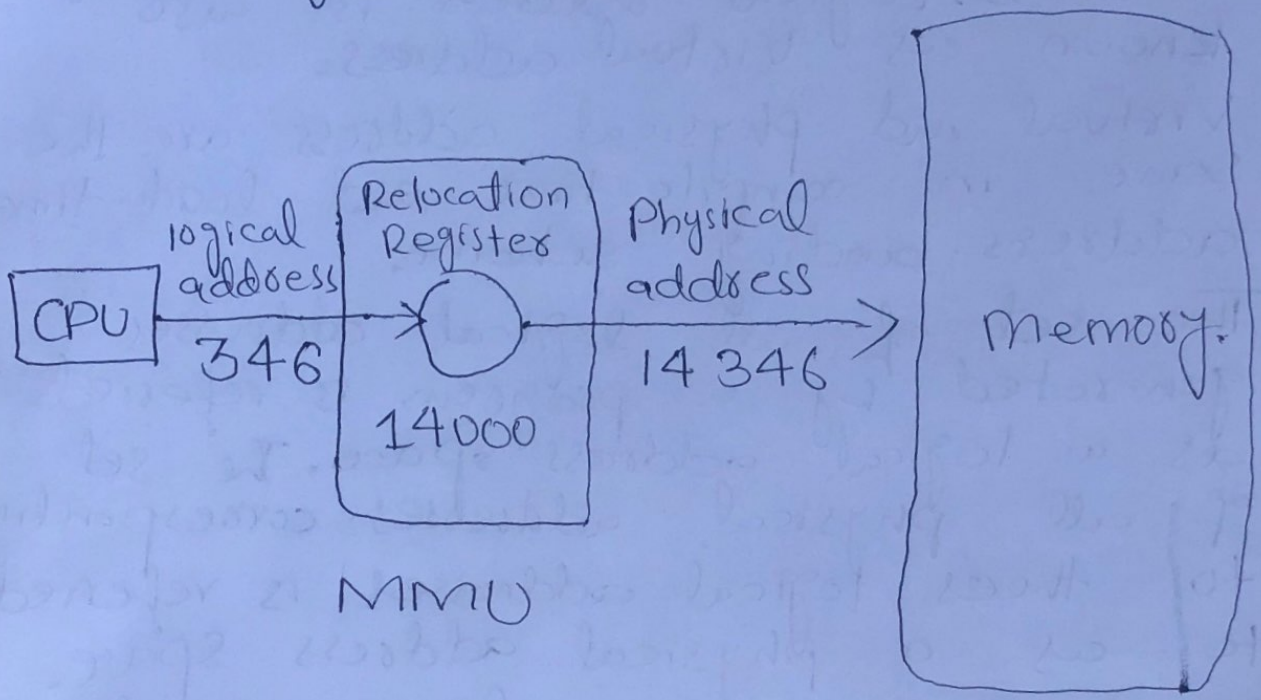
The set of all logical addresses generated by a program is referred to as a logical address space. The set of all physical addresses corresponding to thees logical addresses is referred to as a physical address space.

The run-time mapping from virtual to physical address is done by memory mangemet unit (MMU) which is hardware device. (MMU) use following mechanism to convert virtual address to physical address.

- The value in the base register is added to every address generated by a user process which is treated as offset at the time it is sent to memory. For eample, if the base register value is 10000, then an attempt by the user

to use address location 100 will be dynamically reallocated to location 10100.

- The user program deals with the virtual addresses; it never sees the real physical addresses.



Muhammad Ali Khan
Reg # 16550