ID: 14252

Name: Khalid Malik.

QNO 1.

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 to Create, Schedule, and terminates the processes which is used by CPU The other processes created by the main process are called child process.

A process operations can be easily Controlled with the help of PCB (Process Control Block).

you can consider it as the brain of the process, which cantains all the Crucial information related to processing like process id, priority state, and content cpu register e.t.c.

Thread:

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

Multiple threads share information like data code, files e.t.c.

We can implement threads in three different ways.

1. Kernel-level-threads
2. User-level-threads
3. Hybrid-threads.

Parameter	Definition
Process	Process means a program is in execution
Thread	Thread means a Segment of a process.
Process	The process is not Lightweight.
Thread	Thread are Lightweight
Process	The process takes more time to termihate.
Thread	The thread takes less time to terminate.

NO 2. Threads and its types.

Thread is a single sequence stream within a process. Threads have same properties as of the process so they are called as light weight processes. Threads are executed one after another but gives the illusion as if they are executing in parallel. Each thread has different states Each thread has

① A program Counter
② A register set
③ A stack space

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 does not need to call OS and to Cause interupt to kernal. Kernal do'esnt know about the user level thread and manages

Them as if they were single-threaded processes.

Advantages of ULT

* Can be implemented on an os that doesn't support multithreading
* Simple representation since thread has only program Counter, register set, stack space.
* Simple to Create since no intervention of kernal.

Disadvantages of ULT

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

② Kernel Level thread (KLT)
kernel knows and manages the threads. Instead of thread table

table in each processes, the kernel itself has thread table (a master one) that keeps track of all the threads in the System. In addition kernel also maintains the traditional processes os kernel provides system call to create and manage threads.

Advantages of KLT

* Since kernel has full knowlede about the threads in the system Scheduler may decide to give more time to processes having large number of threads.
* Good for application that frequently block

Distoantage:

* Slow and inefficient
* It requires threed Control block so it is an overhead.

13: Deadlock.

A deadlock is a state in which each member of a group is waiting for another member, including itself to take action action, such as sending a message or more commonly releasing a lock. Deadlock is a common problem in multiprocessing systems, parallel computing, and ~~distrube~~ distributed systems, where software and hardware locks are used to arbitrate shared resources and implement process synchronization.

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 resource held by another its state indefinitely because the ~~waiting process then~~ resources requested by it are being used by another waiting

waiting process. then the system is said to be in a deadlock

In a Communications system, deadlock occur mainly due to lost or corrupt signals rather than resource Contention.

04 CRITICAL SOLUTION

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

SOLUTION TO CRITICAL PROCESES

* A solution to the critical section problem must stastfy the following three condation.

* MUTUAL Exclusion

out of the group operating process only

one processes can be in a critical Section at a given point time.

(2) PROGRESS

If no proces. in a critical Section. If one or more threads wants to execute their critical action that any of the thread must be allowed to get its a critical Section.

(3) BOUNDED WAITING

After a proceses makes a request for a getting into a critical Section. There is a limit for how many other proceses can be into their critical Section. befor this proces request is granted. So after this limit is reached System must be granted the proceses permission to get into its a critical Sectin.

Q NO 5

The OS can be have the ability to do

(1) Dynamic loading modules it uses.

(2) Load time linking of modules needed by an application.

(3) Run time dynamics linking of modules an Application needs if and when its actually need to them

(4) Dynamic loading of modules it uses.

* The OS might load some i/o drivers and other services modules of the OS starts up based on a list. might also load other modules based on need such as never a particular device is plugged.

(2) LOAD TIME LINKING OF MODULES NEEDED BY AN APPLICATION.

* Some operating System allow to user to invoke and object code main main modules even if his not yet been link-edited with the other modules the complete executable. The linking loader load the intial modules inspects it to find the modules it needed load them into memory and fixes upto function to all addreses is needed to work to together. It continue that for modules that other call modules as needed. All this is memory the result is not saved to disk for subsequent reuse is not saved to disk for the subsequent reuse as a whole.

(3) Run time dynamics linking of modules of Application needed if and when its actually need then

PHIS

BASIS FOR COMPARASION	LOGICAL ADRES	PHYSICAL ADDRESES
Basic	It is the virtual Addreses Generated by CPU	The physical Addreses is a location in a memory unit.
Addreses Space	Set of the logical Addresses to generated by CPU in reference by a program is refered as a logical Addres Space.	Set of all physical Addreses mapped to the corresponding Logical Addreses is refered as a physical Addres.
Vissibility	The user can view the logical Addchess of the program.	The user can never view physical Address of program.
Acceses	The user uses the logical Addres the acceses the physical Addres.	The user cannot directly Acceses physical Addresss.
GENERATION	The logical Addchess is generated by the CPU.	physical Addreses is Computed by MMU.