OS Assignment

1) What are Various functions of kernel of Unix?

A) Function of kernel system in OS are

a) Process Management:

i) creation, execution of process takes place

ii) The process contain all info that need to be done

iii) To execute a task, a process is created inside system there are many process available to perform task system

iv) The management of process is very important to avoid deadlocks of for proper functionalities of system.

v) The management of process is very imp to avoid deadlocks of proper functionality of system.

b) Memory Mangement:

i) Whenever the process is created and executed it occupies memory and whenever it get terminated it can be reused again

ii) But memory should be handled by someone so that the realesed memory can be assigned to new process

iii) The kernel keeps track about which part of memory is currently allocated of which part is available other process.

c) Device Management: The kernel also manages all the different devices which are connected to sys like i/o devices

d) Interrupt handling:

i) While executing the process, there are conditions where task with more priority need to be handled first

ii) In these cases, the kernel has to be interrupted in between to execution of current process of handle task with more priority which has arrived in between.

e) I/O communication:- All the information that the system receives from the user of all the o/p that the user is provided via diff application is handled by kernel.

2) Explain the services provided by a kernel I/O subsystem?

A) The service provided by kernel I/O systems are:

i) I/O scheduling: To schedule a set of I/O request means to determine a good order in which to execute them.

* The order in which application issues the system call
* It can improve overall performance of system can share device access permission fairly to all process reduce average wait time, response time for i/o to complete

ii) Buffering:

* A buffer is a memory area that stores data being transfered bet between 2 device or device and application
* hold existing copy of data then
* Buffering is done for:
  i) support copy semantics for application I/O
  ii) provide adaptation for data have diff data-transfer sizes

ii) ~~Buffering~~: caching: i) To hold the copy of data.
  2) Access to cached copy easier than orginal copy
  3) for instance, the instruction of copy currently running process stored on disk, cached in physical memory,

v) Spooling of device Reservation:

* A spool is a buffer that holds the o/p of device such as printer that cannot accept inte data streams
* The o/p of all application is spooled in a seprate disk file. When an application finishes printing then the spooling system queues the corresponding for o/p to printer.

v) Error handling: An os that uses protected memory can go against many kind of hardware of application error so, that complete system failure is not that uses.

vi) I/O Protection:

* A user process may attempt to issue illegal I/O instructions to the privileged instructions where user can issue I/O instruction directly. It may issue illegal I/O instruction to disrupt normal function of system.

3) Describe various multi-threading models in details.

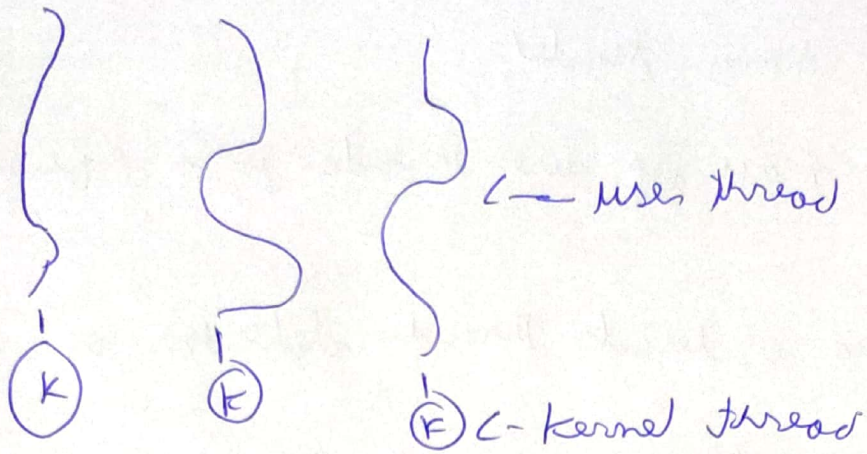
Ans) Multi-threading allows execution of multiple parts of programms at same time. These parts are known as threads and are light weight process available with process.

Modes of Multithreading are:-

1. One to one Model:-

i) Maps each of user threads to a kernel thread. This means that many threads can run in parallel on multiprocessor and other threads can run when one thread make blocking system call.

disadvantage: creation of user thread requires a corresponding kernel thread. Since, a lot of kernel threads burden the system restriction is on no of threads the system.



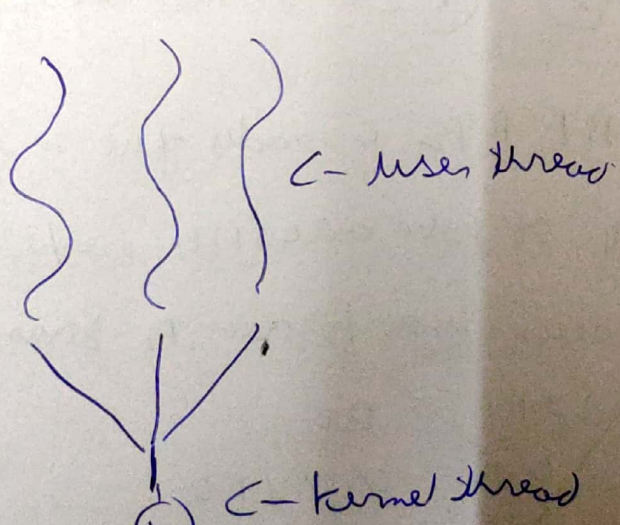
c) Many - to one model:

i) Maps many of user thread to a single thread

ii) quite efficient as user space manages the thread management

iii) Disadvantage: thread blocking system call blocks entire process

Multiple thread cannot run in parallel as only 1 thread can access kernel at time.

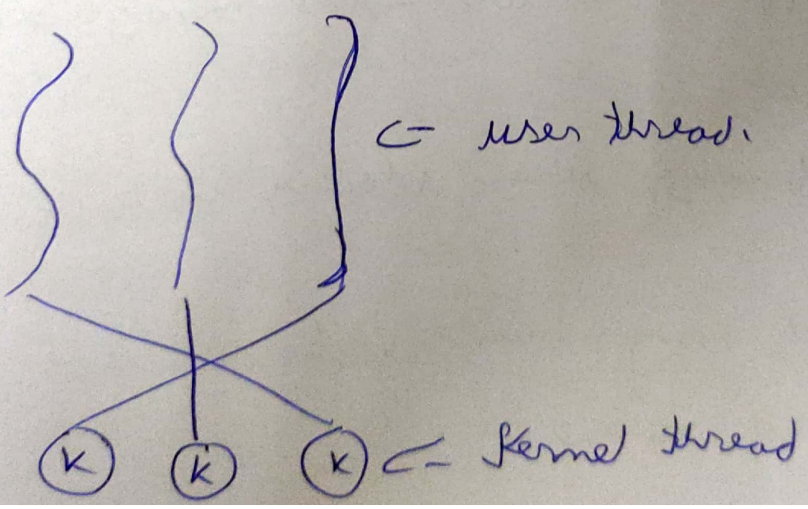


3) Many to Many Model:

i) Maps many of user threads to a equal no of less kernel threads

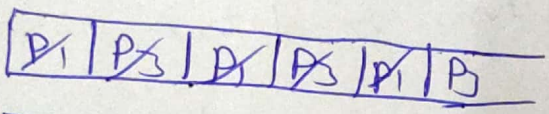
ii) The no of kernels threads depends on application or machine

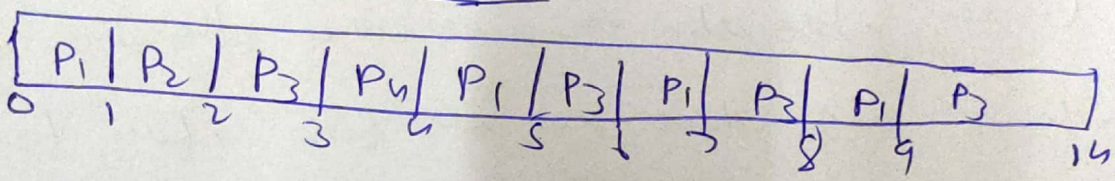
iii) There can be as many user threads as required and there ~~threading~~ corresponding kernel threads can run in Parallel on multiprocessors.



4)

Process	AT	BT
P_1	0	4 2 1 0
P_2	0	1 0
P_3	0	8 1 1 5
P_4	0	1 0

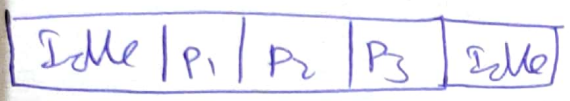
Ready que 

Gant chart 

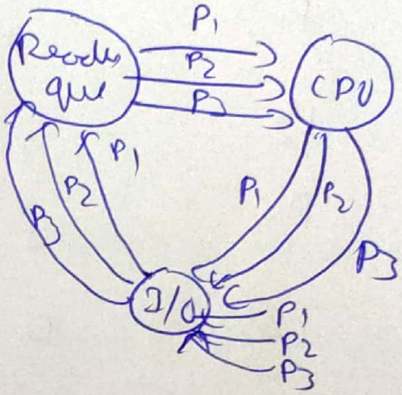
C.T of Process $P_1 = 9$

5)

Process	AT	BT	I/O time	CPU time	I/O time
P_1	0	10	$(10*20/100)=2$	$(10*70/100)=7$	$(10*10/100)=1$
P_2	0	20	$(20*20/100)=4$	$(20*70/100)=14$	$(20*10/100)=2$
P_3	0	30	$(30*20/100)=6$	$(30*70/100)=21$	$(30*10/100)=3$



total time = 47



CPU idle time = $\frac{2+3}{47} \times 100 = 10.63\ \%$

6) Discuss Banker Algorithms in details

A) A banker algorithm is a resource allocation and deadlock avoidance algorithms that test for safety for simulating the allocation for ~~bef~~ predefined maximum possible amount of all resources, then makes an "S-state" check to test for possible activities before deciding weather allocation should be allowed to continue.

Problem statement: It can be used in banking system to check weather loan can be sanctioned to a person or not. Input data

Input data: suppose there are n number of account holders in bank of total sum of their money in $.

→ If a person applies for loan then the bank first subtract the loan amount from total money that bank has of if the remaining amount is greater than $ the only loan is sanctioned.

It is only done because if all account holders comes to withdraw their money from bank easily do it. other words the bank would never allocate its money no longer satisfy customers.

The bank would try to be in safe state always.

Safe sequence Algorithm:

1) Let work of finish vectors of length 'm' of n respectively.

Initially work = Available

Finish[i] = False for i = 1, 2, 3, 4 ... n

2) Find an i such that both

a) Finish[i] = false

b) Need C ≤ Work

3) If no such i exist goto step 4

work = work + Allocation[i]

Finish[i] = true. goto step(2)

4) If finish[i] = true for all i

the system in safe state.