

Test-1

12/01/2024

3a) Interprocess communication involves information sharing between processes.
It can be two ways:

- * Independent: The execution of processes is not affected by other processes execution.
- * Co-operative: The execution of process is affected by other processes execution.

Co-operative interprocess communication is crucial in an interactive appliance. Here are few (applications) ^{advantages} of interprocess communication:

- Information sharing: It involves exchange of services / resources between different processes.
- Modularity: It increases modularity by splitting into sub-processes thereby increasing speed.
- Convenience: It makes it convenient to execute different processes.

Co-operative IPC takes place in two ways

- * Message passing
- * Shared memory

→ Message passing: This method is slow

- This method is used when the data is small.



- In this method each block is created.

- In this method system call has to be called whenever it has to be read/written.

→ Shared memory:

- A common pool of region is created in this method.



- System call is called only once

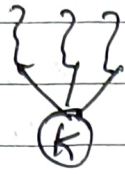
- It is faster

- It is used when data is large.

3b) Multi-threading models can be understood as a relationship between user thread & kernel thread.

There are mainly 3 multi-threading models.

→ Many to one: Here many user threads are connected to one kernel thread.

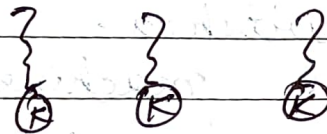


- Here various libraries can be used from single kernel thread.
- But whenever a block system call is initiated, ^{is one user thread.} whole ~~that~~ set-up is affected.

Eg: Solaris

→ One to one model:

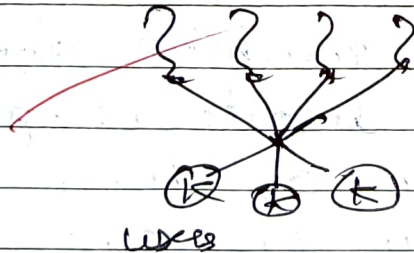
Here one user thread is linked to one kernel thread.



- Here parallelism is introduced and block system call ~~is~~ in one thread does not affect the other.

Eg: Windows

→ Many to many model:



Here several threads is connected to several kernel threads

— There is high parallelism

Eg: HP36

- 2a) • Virtual machine is an abstract of various internal hardware, so that it creates an illusion that there is separate processor for each.
- virtual machine involves a virtualisation layer.
 - Here in a virtual machine one host O.S. is used & many guest O.S. can be used.
 - So, therefore in a single system, that is with the same CPU, other internal hardware various environments can be switched.
 - This virtual machine helps the programmers to comprehend how their code works in different environments (i.e. different operating system).
 - This increases the efficiency of the system & makes information sharing easier.
 - In one virtual machine different environment despite being present, each of them is protected from one another.

2b) • System calls are the means by which we can access the services of the operating system.

• It is necessary ~~to~~ for the interaction between the user and O.S.

• In message passing, method

• system calls is used every time a block is executed.

• There are various categories of system calls

→ Process control

→ Device management

→ File management

→ Information management

→ Communication

→ Protection

• System calls differ for every operating system

→ Process control

Process control system calls are used to create a process, wait for a process or terminate a process.

→ Device management

Device management system call is used to release/receive ^{devices} data ~~files~~, attach or detach ^{devices} ~~data files~~, reposition of ~~data files~~ devices, get/set devices.

→ File management

File management system calls are related to create a file, delete a file, read a file, write a file ~~etc.~~, get/set files.

→ Information management

Information management system calls are related to sleep, alarm, getpid, which manages the information provided.

→ Communication

Communication system files are related to a communication link between services of operating system.

→ Protection

Protection system calls are related to security, to determine access, modify access to certain services of operating system.

Examples for system calls:

→ Process control:
 CreateProcess ()
 ExecuteProcess ()
 WaitProcess ()

} - Windows

fork ()

exec ()

wait ()

Exit ()

} - UNIX

→ File management: create ()

read ()

write ()

getFile ()

setFile ()

→ Device management: read ()

write ()

→ Information management: getpid ()

sleep ()

alarm ()

→ Communication: shmget ()

→ Protection: unmask ()

chmod ()

Quiz

1. c) new ✓
2. (a) when process is scheduled to run after some execution
3. (b) communication b/w two processes.
4. (b) Program counter ✓
5. b) 5 ✓ ~~ready~~ ~~run~~ ~~wait~~ terminate