computing.

Past Qst

Q1 * OS definition. * * * *

Q2 * Fundamental diff bet^n monolithic & layered structure of OS.

Q3 * advantage for an OS designer for using virtual machine architecture. main advantage for user.

Q4 * * * "Operating system acts as entended machine as well as resource manager." enplain, with example, & clarify

Q5 * * What are system calls? Enplain types of system calls.

Enplain sequence of system calls for copying one file to a new file.

Q6 * Types of kernel with necessary diagrams

Q7 * main function of OS? Dissuss different stuctures of OS in brief.

Q8 * OS as a resource manager.

Q9 * Monolithic & layered structure of OS

Q10 * Major function of OS

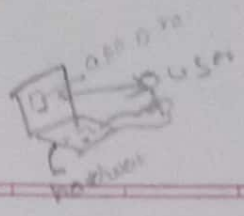
Q11 * diff. between multitasking & multiprogramming. prop. of Batch-oriented & time sharing OS

Q12 * Time sharing, parallel & real-tm-OS?

Q13 * Services by OS

Short notes
shell
OS structure

Gurukul



PAGE NO.:
DATE : / /

Q1 Operating system

Definition:-

Operating system is a program that manages the computer hardware and provide basis for application program and acts as intermediary between user and computer hardware.

Importance of operating system:-

Facilitates communication between hardware & software,
manages resources efficiently
provides a user-friendly interface

Q2 Two main function of OS

i) OS as entended machine (virtual machine) user interface
ii) OS as resource manager.

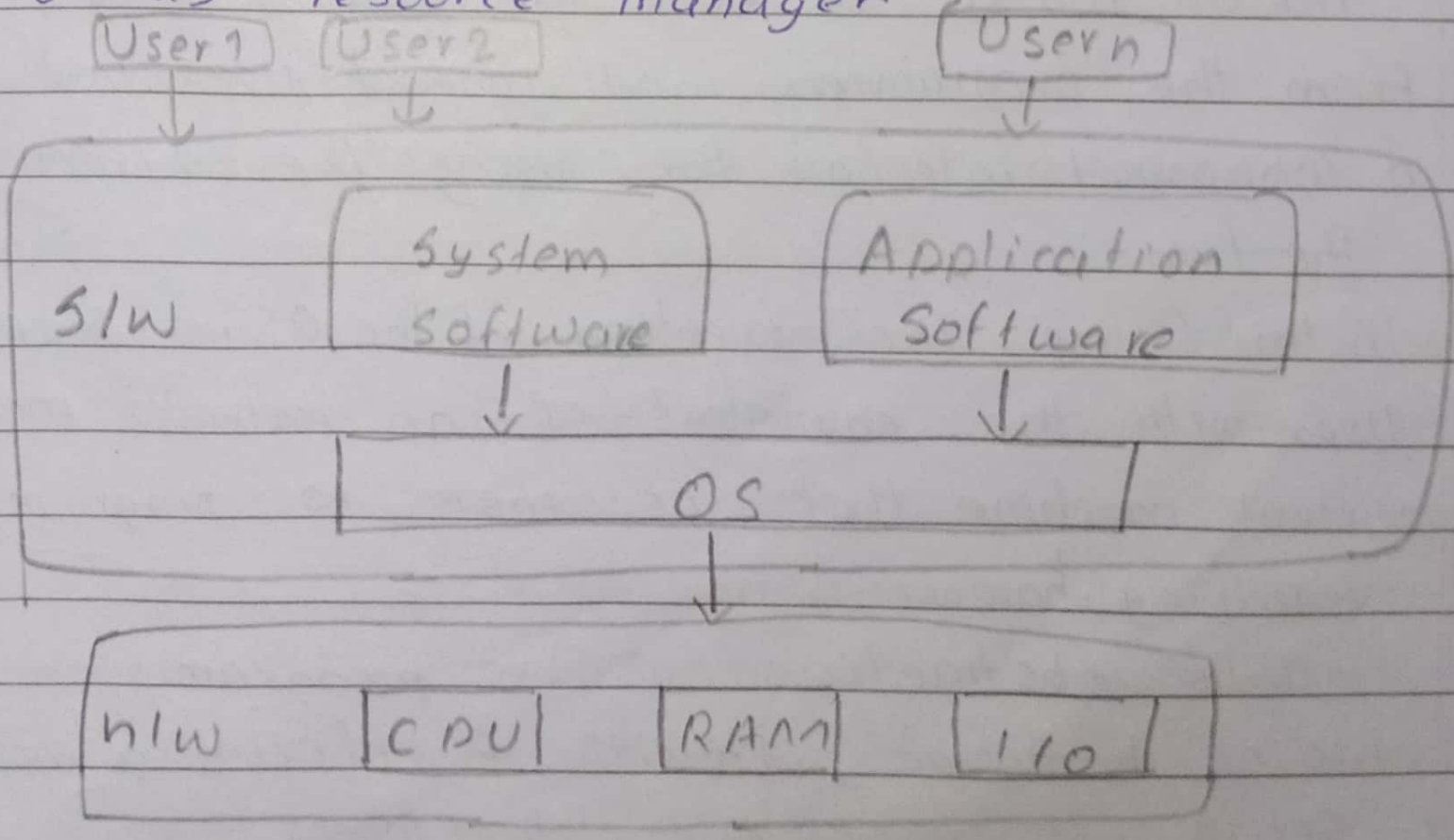


Fig: Operating System

PAGE NO.:
DATE : / /

Q4 OS as a Resource manager as well as entended machine

Write OS defn:-

OS as a Resource manager :-
The OS efficiently manages hardware and software resource such as CPU time, memory, storage & I/O devices. It uses scheduling algorithm for CPU allocation, memory management techniques like virtual memory & organizes file systems for storage control.
By effectively managing these resources, the OS optimizes system performance, enhances user enperience, and enable multitasking.

Os as an entended machine.
The OS masks or hides the details of the Hardware from the programmers and general users and provides a convinient interface for using the system.
~~The func~~
In this view the function of the OS is to present the User with the equivalent of an entended machine or virtual machine that is easire to program than underlying hardware.
This abstraction makes programming easier by hiding hardware intricacies & offering a more convinent set of instruction for developers

100+18
100

PAGE NO.:
DATE: / /

By fulfilling both roles, the OS acts as a resource manager, efficiently allocating resources while also serves as an entended machine, offering a simplified & user-friendly interface for users & programmers.

Mgjor function of OSi

Memory management function
processor management function
I/O device management function
File management function.

System call
A mechanism that allows a program to request seruiees from OS kernel.
System call types:-
It can be divide categorize into five major categorise.

Process control
load,
enecute,
get set. Process
create process
terminate process
allocate, free memory

2) File management
create, delete
open, close
read, write, reposition
get set file attribute.

3. Device manage
request, release,
read, write,
reposition
logically attach
or detach devices.

Gurukul

PAGE NO.:
DATE : / /

4) Information maintenance
get/set time, date
get/set systemdata
get/set file, device, process attributes

5) Communication.
create, delete communication connection
send, receive message
transfer status info
attach or detach remote device.

Seqvence of system calls for copying one file to new file

Open() : source file
open() : destination file.
read() : read from source file
write() :- written to destination file.
Close() :- Both are close.

Shell

Shell is the program that provides ~~provi~~ traditional text only user interface for Linux & Unix other Unix OS.

It's primary function is to read command typed into a consoler or terminal & enecute it.

The term shell derives it's name from the fact that is an outer layer of the OS.

The most fundamental way that user can interact with the system & the shell hides the detail of underlying system from user.

Eg :-

Bourne shell, Bash shell, korn shell, cshell

PAGE NO.:
DATE: / /

q7
q9
q3

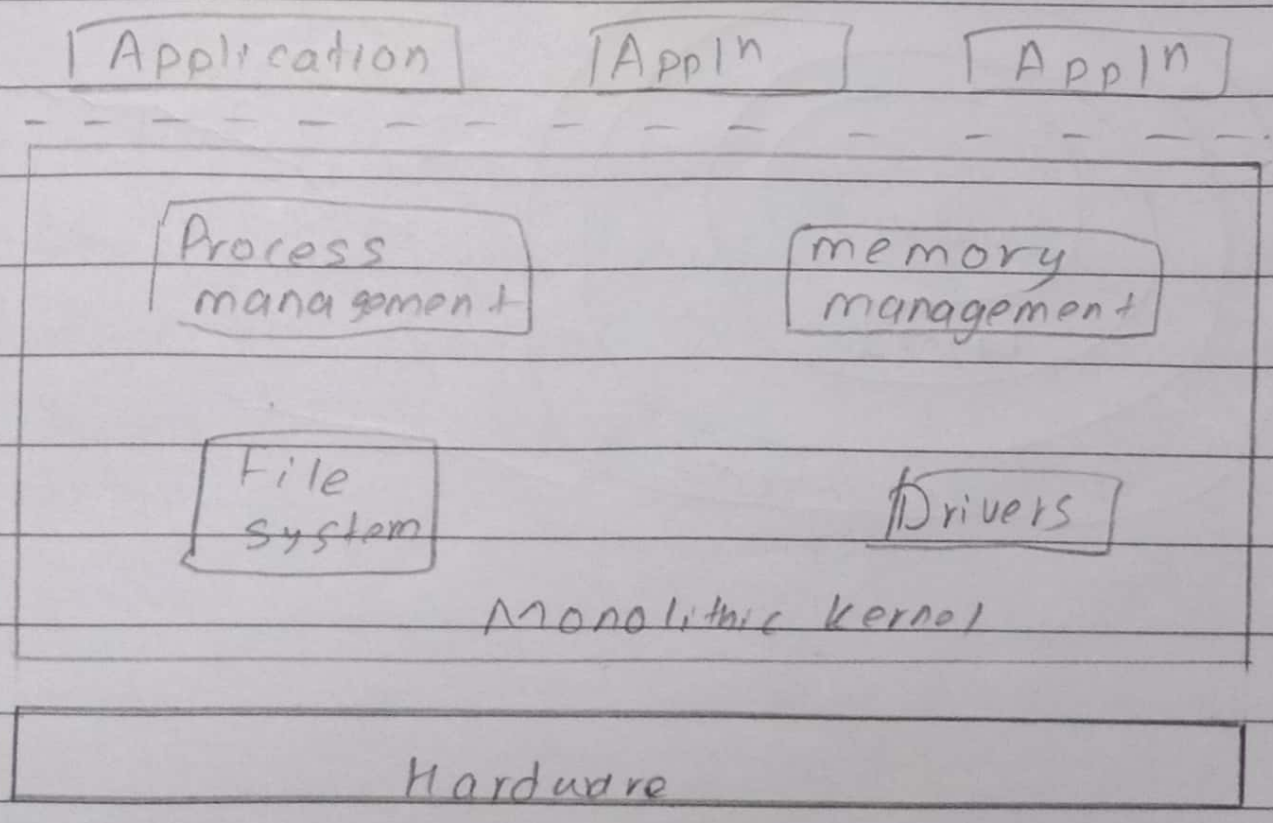
OS structure.

OS are broadly classified into following categories, based on their structuring mechanism as follows:-

Monolithic system
layered system
Virtual machine
Exokernels
client-server model

Monolithic system.

In monolithic architecture, the entire OS runs as a single program in kernel space.
Services includes: process management, memory management, file system, device drivers.
Example:- Linux & older versions of UNIX.



Gurukul

PAGE NO.:
DATE : / /

Advantage:-

High performance due to direct access to hardware.

Disadvantages:-

Lack of modularity, difficult to maintain & entend.

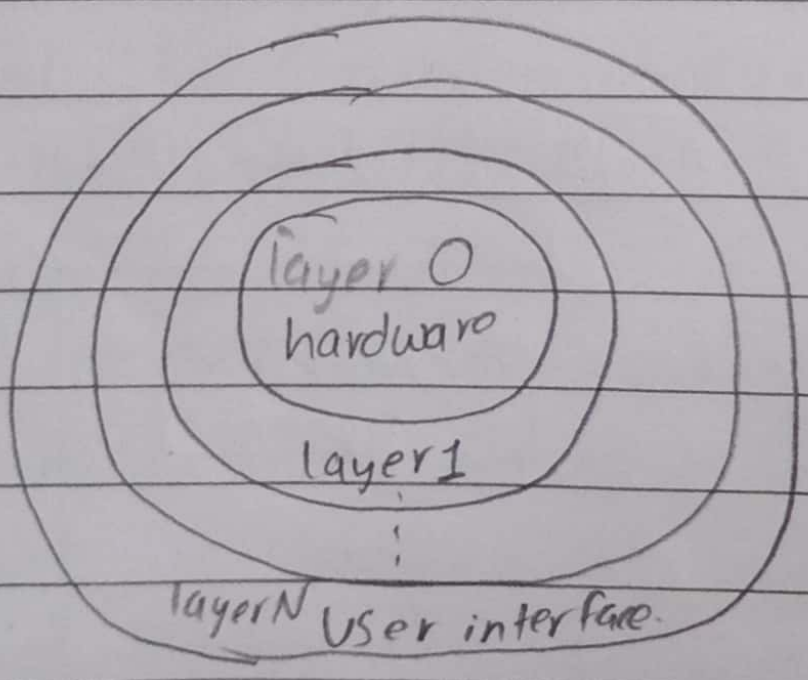
Any services failed entire system failed.

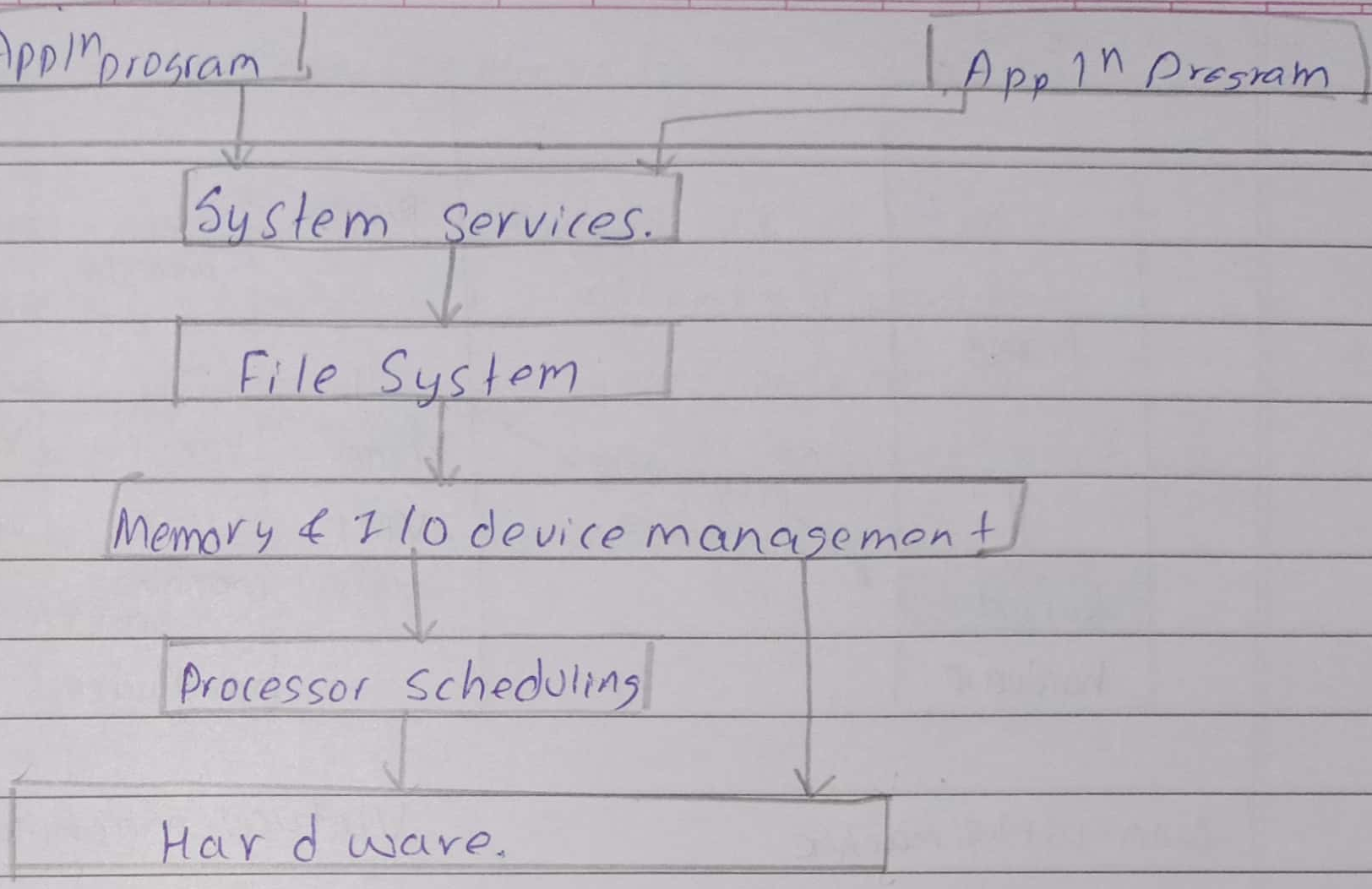
layered operating system

In layered system,

the operating system is divided into hierarchy of layers, with each layer providing specific functionalitres.

Each layer interact only with adjacent layer, & communication between non-adjacent layer is not allowed.





Advantages:-

Modularity — Easy debugging

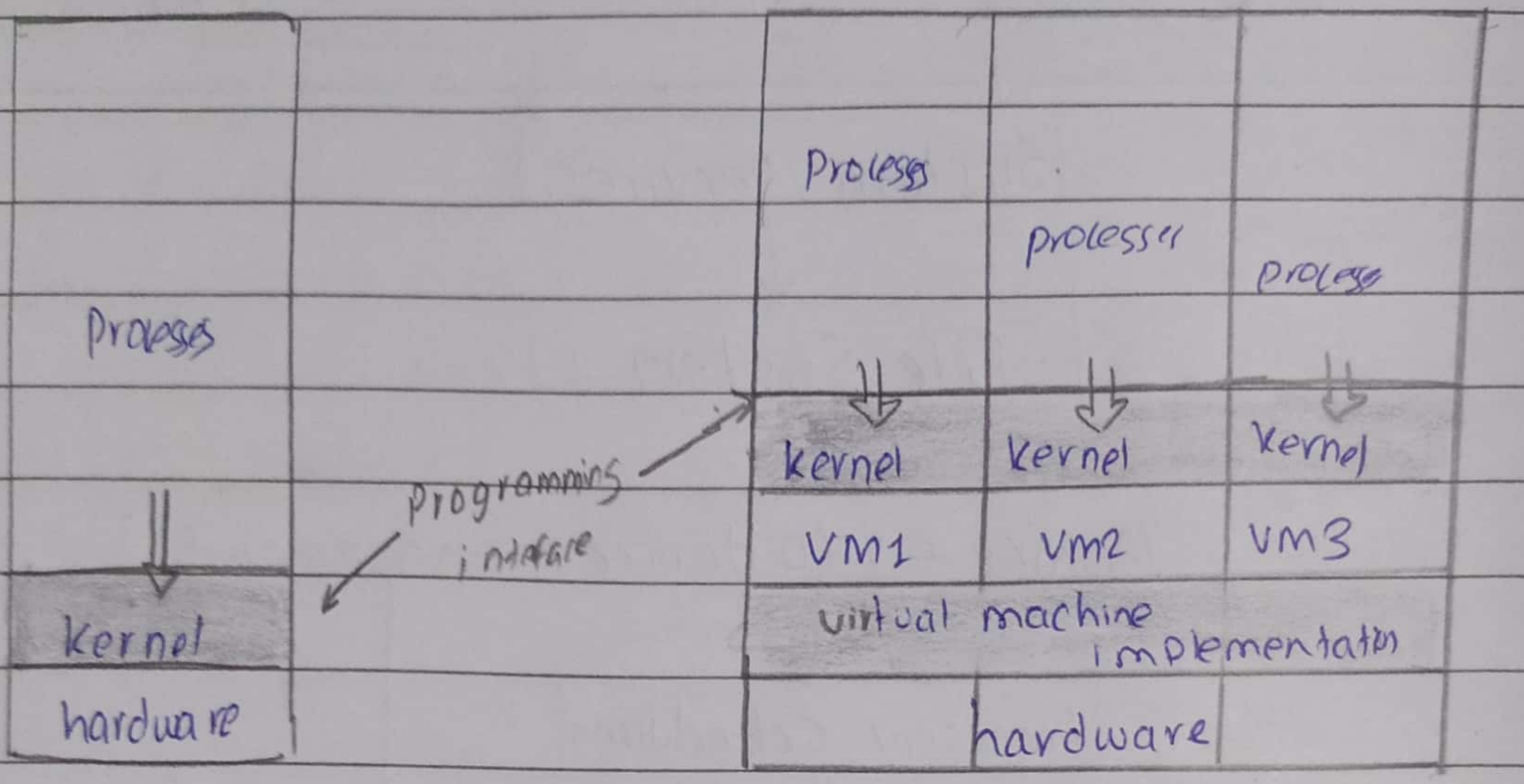
Easy update — Abstraction

Disadvantage.

Complen , Slowor.

Virtual Machines.

In virtual machines architecture, hypervisor creates virtualized environment for multiple OS instances.



Non-virtual machine

Virtual machine

Benifits of Vms.

Due to their ability to create isolated & independent OS environments within a single physical machine., VM's are beneficial.

multipe OS environment enists simultaneously on same machine, which is isolated from each other.

Easy maintainance, application provisioning

PAGE NO.:
DATE: / /

Client-server or microkernel.

In client-server model, the OS is divided into two distinct component : clients & servers. Clients are responsible for user interaction & appln logic, while servers provide services such as file storage, printing & database access

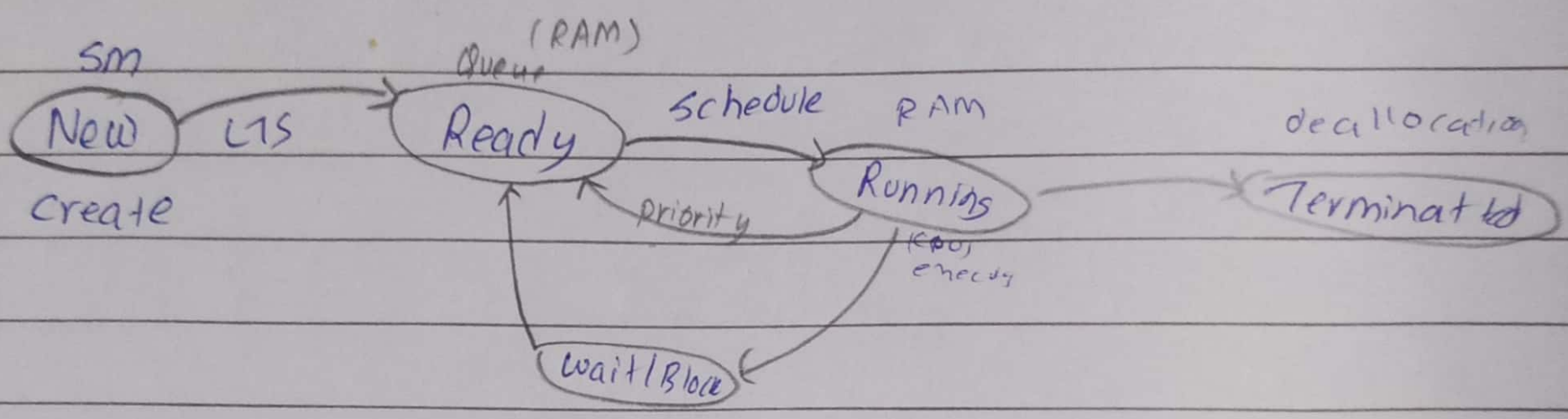
=

Batch OS

PAGE NO.:
DATE : / /

Define PCB. How do process transition betn diff states & what triggers these transitions?

Process states



LTS - long term schedular
(multiprogramming)

Fig: process states & its transition

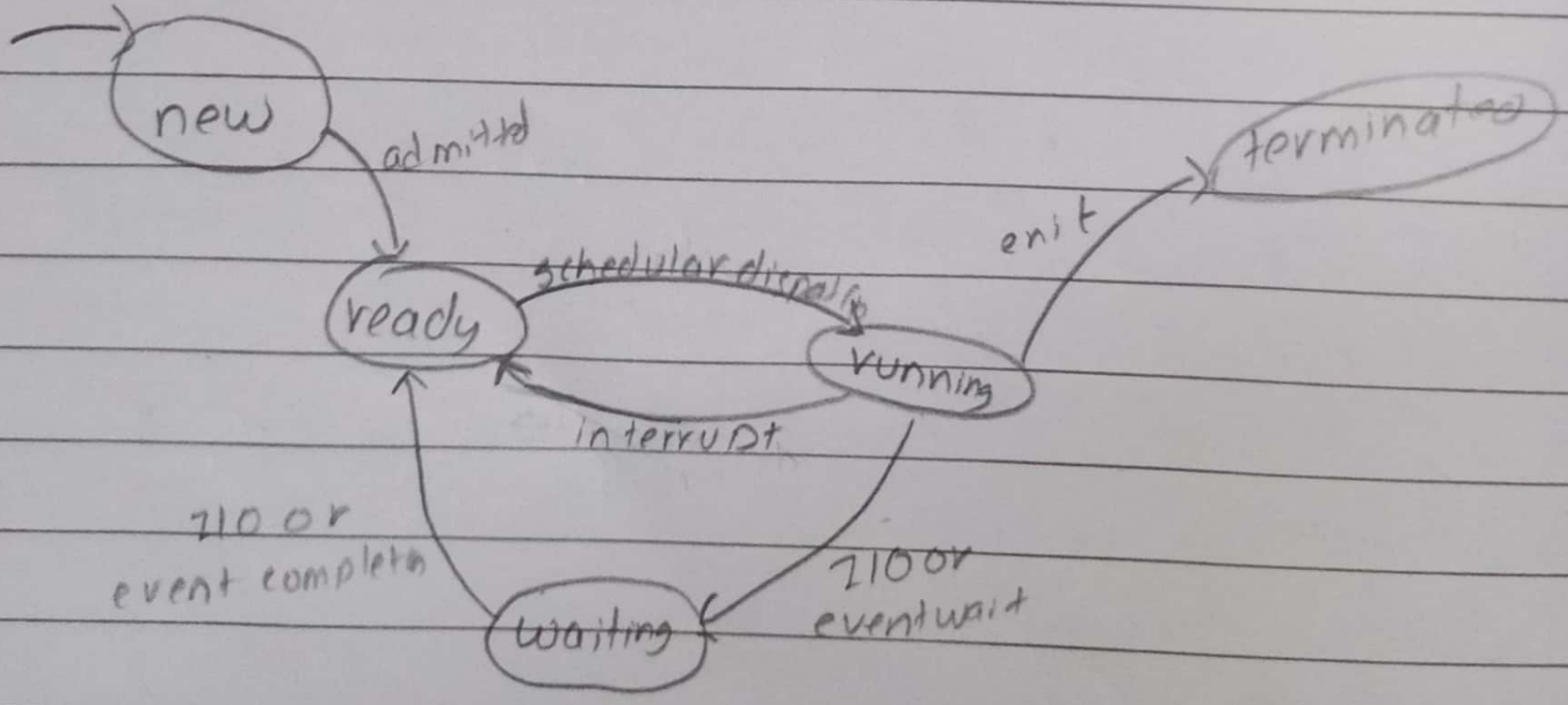


Fig: Process states & its transition

PAGE NO.:
DATE: / /

Process is an instance of a program in execution.

When a process executes, it passes through different states.

These stages may differ in different OS, as names are not standarized.

In general, there are five state.

New: The process is being created

Ready: process is waiting to be assigned to a process

Running: Instructions are being executed

Waiting: process is waiting for some event to occur.

Terminated: process has finished execution

PAGE NO.:
DATE : / /

** For what purpose semaphores are used? Give soln to producer-consumer problem using
* semaphores
** Diff threads & process.
* How multiprocessor system can be effectively utilized with threads.
**** test & set instruction? Explain producer-consumer prblm & algorithm for resolving using monitor.
* Explain field in PCB.
* Explain process state transition diagram used in multiprogramming environment.
** How can mutual enclusion affect program performance
Describe sleeping barber problem with pseudocodes
* Multithreading, Aduantage of implementing thead in user space.
* Reader-writer problem. IPC problem.
Race cond^n & mutual enclusion. how mutual enclusion be achieved using TSL.
* Sleep & awake.
User level & kernel level thread.
What is critical region? Explain Dekker's algo.

PAGE NO.:
DATE: / /

PCB (Process control Block)

Each process is represented in the OS by a Process control Block.

(PCB) - also called task control block.

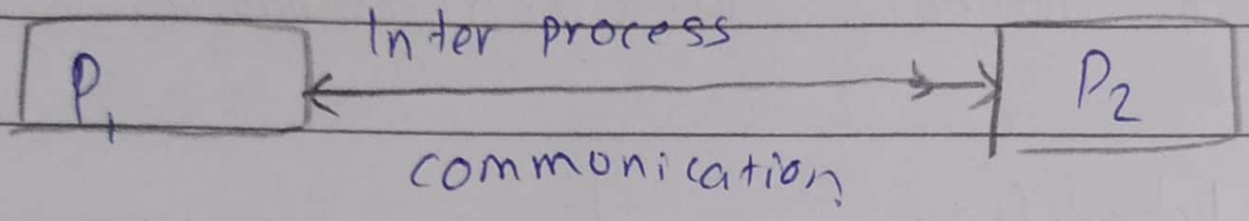
Identifier
State
prioritize
program counter
memory pointer
Content data
I/O status info
Accounting info.

identifier :- to distinguish from all other process.
State :- executing - running state.
priority :- priority level relative to other process.
program counter :- address of next instruction to be executed.
memory pointer :- pointer to program code.
conten data :- present in register.
I/O status :- I/O requests, I/O devices,
Accountinginfo :- amount of processor time, clock time, time limits.

PAGE NO.:
DATE : / /

IPC (Inter-process communication.

IPC refers to the techniques & mechanisms used by process to communicate with each other.



Race condition.

It is an unpredictable behaviour due to multiple processes or threads accessing shared resources concurrently without synchronization
Eg:- Two threads trying to update same variable simultaneously.

mutual exclusion:-

It ensures only one process or thread can access a shared resource at a time to prevent race condition
Eg: Using locks or semaphores to control access to critical sections of code.

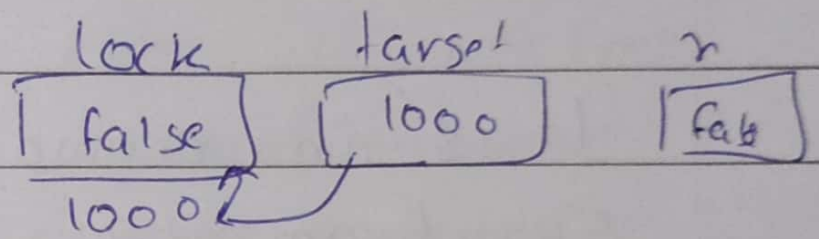
1 = true
0 = false

PAGE NO.:
DATE: / /

How mutual enclusion be achieved using TSL?.

```
while( test_and_set (&lock));
Critical Section
lock = false;

boolean  test-and-set (boolean *target)
{
boolean r = *target;
*target = True;
return r;
}
```



Sleep & awake

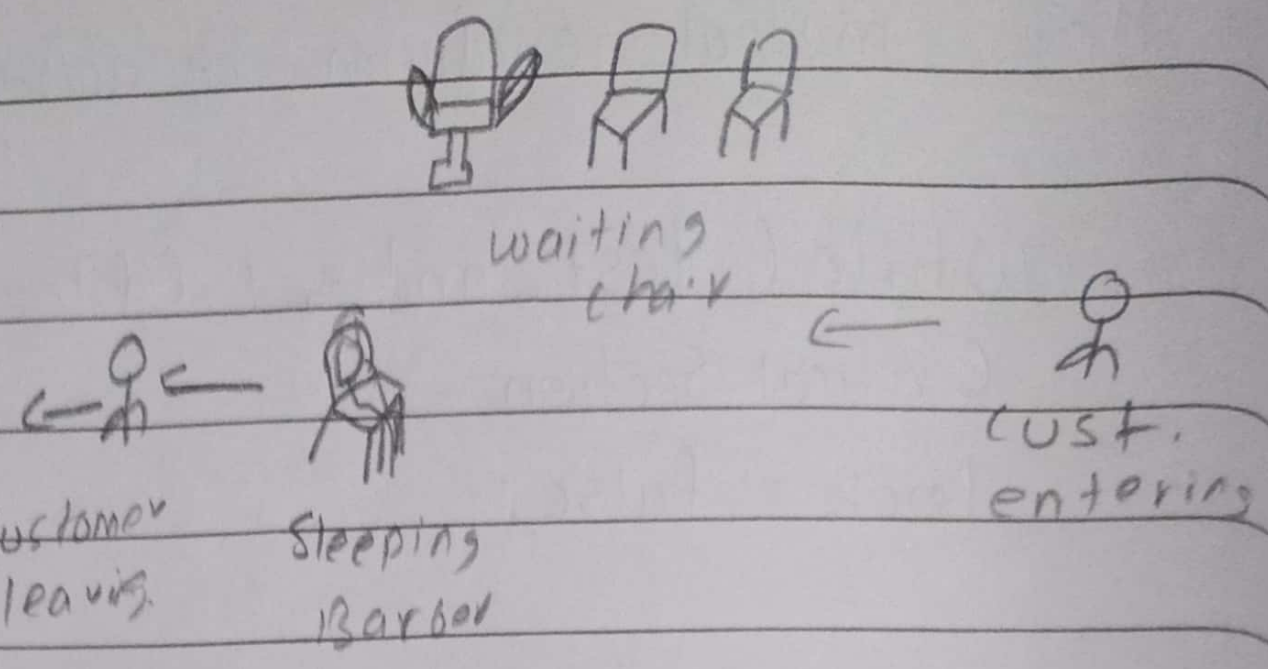
Concept of sleep & awake is very simple. If critical section is not empty then the process will go & sleep. It will be waked up by the other process which is currently executing inside the critical section so that the process can get inside the critical section.

~~If~~ no customer - barber sleeps in his own chair.

when customer arrives - he has to wake up the barber

many custome - barber cutting customer hair - remaining will either wait ~~of~~ have empty chair or go leave.

DATE : / /



Example to use sleep & wakeup :-

Producer & consumer's Problem.

It is an example of multi-processing synchronization.
producer produce (generate data item,
consumer consume these items.

Challenge is to ensure, that producer do not produce data item faster than consumer can consume them & vice versa.

Producer :-
Producer must be synchronized to avoid overfilling the buffer when it's full.

2). Consumer must be synchronized to prevent attempting consume from an empty buffer.

Synchronization mechanism are needed to ensure proper coordination between producer & consumers.

PAGE NO.:
DATE: / /

Problem :-

If common fixed-size buffer is given, the task is to make sure producer can't add data into buffer when it is full, & consumer can't remove data from empty buffer.

Solution :-

- Allow producer to sleep or discard data if the buffer is full.
- When consumer removes an item, it notifies the producer to start filling the buffer again.
- consumer can sleep if buffer is emptly, & producer wake up when data is available. proper coordination prevent deadlock situation.

N = man no. buffer can store,

```
#define N 100
int count = 0
void producer (void)
{
  int item;
while (True) {
item = produce_item();
if (count == N) sleep();
insert_item (item);
count = count + 1;
if (count == 1) wakeup (consumer);
} }
```

```
void consumer (void)
{
  int item;
while (True) {
if (count == 0) sleep();
item = remove_item();
count = count - 1;
if (count == N - 1)
wakeup (producer);
item consume_item (item);
```

PAGE NO.:
DATE : / /

Semaphore

They are integer variables that are used to solve critical section problem by using two atomic operation.

wait & signal that are process used for synchronization.

Dining Philosophers problem,

This involves N philosophers sitting around a circular table, each needing two forks, to eat spaghetti.

Each follow seqaunce of action

Think,
pickupt left fork.
pick up right fork,
eat
putdown right fork
put down left fork.
repeat.

This soln leads to deadlock. if all philosopher pickup left one.

It leads to starvation.
The situation, where program continues to run indefinitely without progress is called starvate.

PAGE NO.:
DATE: / /

One solution :-

* Use an array called 'state' to track whether a philosopher is eating, thinking, or hungry.
* philosopher can move into eating state if neither philosopher are eating.
defined by Macros Left & Right.

Reader's Writers problem.

It is acceptable to have multiple processes reading the database at the same time, if one process is updating (writing) the database, no other process may have access to the database, not even a reader.

DATE : / /

Thread.

Thread is a flow of control within a process.

basic unit of CPU utilization.

with its own program counter, register & stack.

Two a types:-

Single threaded process.

In this,

each process has its own address space, and a single thread of control.

doesnot provide parallelism.

Multithread praess.

Single process has moltiple thread of control.

provide parallelism, & hence improve performance

User level thread

managed in user space of main memory

kernel is unaware of these threads

Thread management is handled by user-level thread library.

Threads are faster to create and manage.

Advantages:-

easy implementation

no OS modification required.

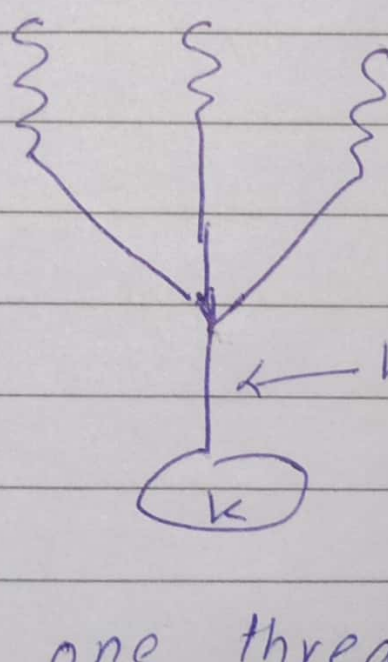
Disadvantage:- lack of OS integration.

lack of coordination between threads & OS kernel.

PAGE NO.:
DATE : / /

Kernel-level threads
Defined & managed directly by kernel of OS.
Kernel perform thread creation, scheduling, management.
Used in interna1 working of OS
Advantage:-
1) Include kernel's full knowledge of all threads allowing informed scheduling decisions.
Disadvantage:-
slower & less efficient operations compared to user-level threads.

Multithreading Model.
Many to one model



many user-level threads
are mapped to
single kernel-level thread.

Only one thread can access at a time.
Advantage:-
simple & efficient.