

WJMK
30
१९८२

WJMK
30
१९८२

108 | 2023
Tuesday

OS

Tut : 1

कालतंत्र कार्सेवरिंग इन्स्टर
मरियादा,
निरमुक्तीर कालैवर
अभियान ॥३॥

① Commands in UNIX :-

- (1) mkdir → will create directory with name of OS lab,
- (2) cd → change the directory

cd. OS lab > current

"will change directory to
directory of OS lab" named

- (3) ls → content present in that directory

- (4) cat → display content of file

- (5) cat file → (to create file)
↳ e.g. cat file1.txt

- (6) cat file1 file2 > file3

"in files ka content, file 3 mein copy pasted hogaayega"

- (7) ls *.txt → open all the text files
ls *.c → _____ 'c' "

- (8) pwd -

"present working directory"
↳ shows current directory

- (9) cp - copy the content

- (10) mv - move the content

WEEK 3 COMMANDS

(11) head

↳ will print 1st 10 lines of file1.txt file.
(from the content)

(12) tail → " " last " " "

" " " " " " "
(from the content)

(13) tac → (reverse of cat)

(14) clear → It clears the screen.

Suppose a file

→ file.txt

Line - 1

Line - 2

Line - 3

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

Star lot's

say star

1st 30 line dikhi

ki hain, toh

baaki ki dekhne ke liye → '\$more' use
karte hain!

\$ Vi

(15) \$ Vi file1.txt

↳ to open file → file1.txt in vi editor

(16) grep

WJMK $\frac{C}{3}$ m.

file 2.txt

I am Dip
I am student

↪ want to search 'am' har in file2.txt.

toh ukho →
grep am file2.txt

↪ toh sare 'am' highlight ho jaayenge!

17) diff :-

common hoga jo
files mein ~~wo~~ wo
erase ho jaayega, aur baaki kar behega!

file 3.txt

I am Dip
I am writing

diff file2.txt

file 3.txt

(iska jo bchega,
last file ka;
wo print hoga).

↪ output:-

Now I am writing.

18) Ring :-

19) history :-

hostname :- hostname display kega of system.

hostname - i :- will show IP address of system, we are using.

20)

W5MK $\frac{9}{30}$
22.

22

chmod :-

cat file1.txt



read | write | executable
mode .

cat hello.c

A diagram enclosed in a rounded rectangle containing the C code `printf("Hello World");` with curly braces {} indicating the function body. An arrow points from this diagram to the text "so not executable file as 'bat' likha!".

3 modes hote
ham file ke
basically

so not executable file as
'bat' likha!
chmod' on karte - to make
executable

chmod +x file1.txt

abhi 'file1.txt' executable ho gya!

\$ nl file1.txt

then esse ho jaayegi file
in terminal :-

A diagram enclosed in a rounded rectangle containing the text "1 Line-1", "2 Line-2", "3 Line-3", and a final separator symbol ":-".

23

nl :- file.txt

A diagram enclosed in a rounded rectangle containing the text "Line-1", "Line-2", and "Line-3", each preceded by a number 1, 2, and 3 respectively, followed by a final separator symbol ":-".

$$WOMK = \frac{1}{32}$$

(24) wc :- kisi file mein jisme words hain / characters hain, sbki details degi !!

\$ wc file1.txt → file1.txt

{ l , w , c }
3 , 3 , 15

{ line 1
line 2
line 3 }

(25)

file1.txt

{ line 1
line 2
line 3
line 2
line 2 }
continuous
duplicate
tha,
tob
erase
redigat

Uniq file1.txt → srf unique
cheesein
part
hangi!

{ line 1
line 2
line 3
line 2 }

agar files
hoy iske
andar, tob
remove
kroga

(26)

desktop par

rmdir

folder
hao say with name '(Dip)'

: rmdir Dip

→ remove krega
desktop se Dip ko
only if wo
empty hot, o/w
nahi

(27)

rm cd Dip
rm file1.txt

Dip folder
par
chle jaayenge
on desktop & use andar ki
file → file1.txt will dlt;
remove ho jaayegi

W3MK

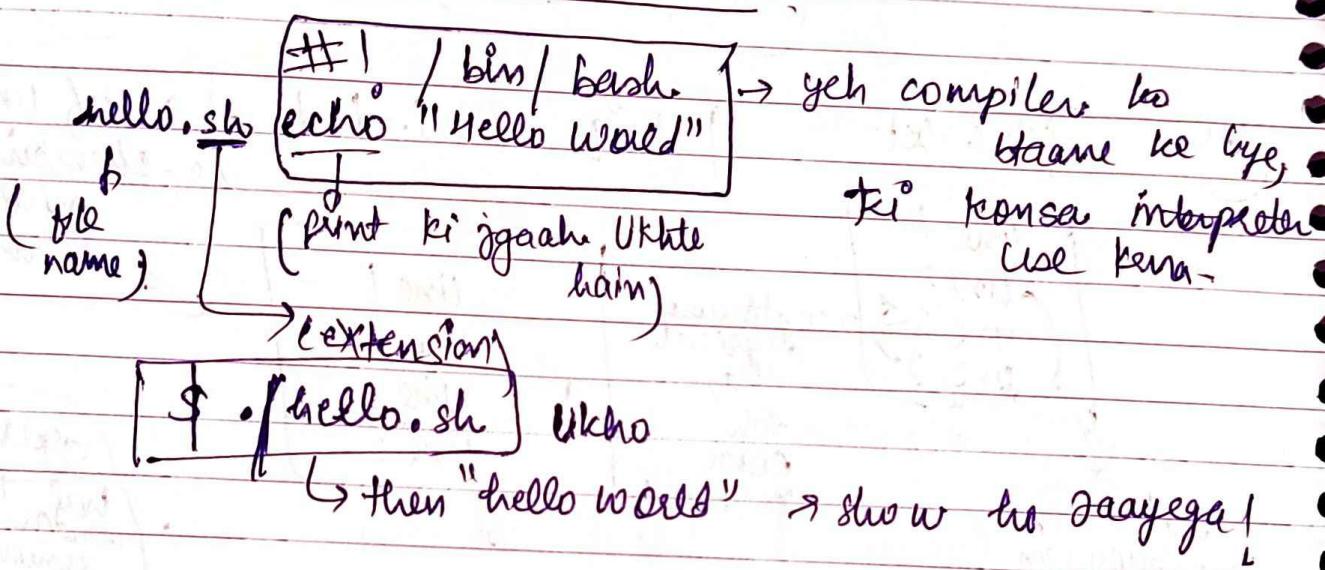
shell
programming

is 'Interpreter'

language, not executable!

means
executable file banane ki zaroorat
nhi, directly execute keta!

Basic syntax to print "Hello world": -



system variable

→ mostly in capital letters

#!/bin/bash
echo "Hello world"
echo \$ PWD

/home/dip/Desktop

variable



system
(capital)

WORK IN PROGRESS.

name = Dip

echo \$name

echo " I am \$name"

I am Dip

For user input :- we use "read".

#!/bin/bash
echo "Hello world"

yeh line toh
common hai, abhi
starting mein likho

read name → variable jismein
echo \$name → input lena chahiye
no.

to read several variables from User :-

read name1 name2
echo \$name1 \$name2 → terminal
mein result wali show.

To input an array :-

echo "enter name: "
read -a names
echo "\${names[@]},"

↑ sare input 'array' (names) mil
rone ko saktega

 \${names[1]} "

jab execute kरnge → " ./hello.sh" likh ke

WORK C
30 min.

- Enter name
- shell programming
- "

while loop :- (2ways)

M-1

while ((condition))

do

—
—
—

[OR] OR
c=1
while ((i <= 10))
do

done

less than equal to
is can also write

<=

echo " \${Names[i]} "

i=\$((i+1)) \${((i+1))}

to get value stored in i

done

M-2
while [condition]

do

while [i -le 10]

do

echo " \${Names[i]} "
[i=\$((i+1))]

done

Can
write
only
in
[]

bracket
and
only
in ()

For loop :-

(Initialization) OR $i \leq 100$ WJMK = $\frac{e}{30} = m.m.$
for ($i=1$; $i \leq 100$; $i=(\$i+1)$) OR

do for ($i=1$; $i \leq 100$; $i++$)
echo "\$names[i]"
done

① for variable in 1 2 3 4 5
do

done

→ for i in 1 2 3 4 5] → output:-
do
echo "\$i"
done

→ for i in {1..100}] → output:-
do
echo "\$i"
done

by default 1 se increment
hota gaya eg,

lekin agar 5 se kera :-

for i in {1..100..5} → 5 se increment
hoga :-

Print 1 to 10 numbers using shell programming !!

A: $i=1$
while (($i \leq 10$))
do
echo "\$i"
 $i=\$((i+1))$
done

WJMK 8
= 30. 30.

WJMK 8
= 30. 30.

WJMK 8
= 30. 30.

NE sem. test
(Esc, Shift, Esc) → to
exit (Esc) → to write
msg!
enter then enter
(Esc) → to write
msg!

- Q/W
- 1) Read a file using C programming.
 - 2) Write .. " " , "

shell program:-

- 3) ~~Create~~ Read the file
Create an array which contain your
all the current semester subject names
- 4) Print the ~~array~~ using while and for
loop.

- A:-
- 1) fgetc() → to read single character from file
 - 2) fgets() → to read string from file
 - 3) fscanf() → to read formatted input from file
 - 4) fread() → to read block of raw bytes from
file, used to read binary files

```
int main()
{
    FILE *ptr;
    char ch;
    ptr = fopen("hello.txt", "r");
    if (ptr == NULL) {
        printf("Error\n");
        printf("Content of file is\n");
        do {
            ch = fgetc(ptr);
            printf("%c", ch);
        } while (ch != EOF);
        fclose(ptr);
        return 0;
    }
}
```

3) \$ cat > hello.txt

```
echo "OS"
echo "TOC"
echo "IA"
echo "IM"
echo "PS"
echo "OOM"
```

is to display prompt "Enter file name:"
to user before reading input & then
store input in variable filename

read (-p) "Enter file name : " filename

4) while read line

```
do
    echo $line
done < $filename
```

2) FILE *ptr;
ptr = fopen ("hello.txt", "w");
fprintf(ptr,
"%s", "Hello");
fclose(ptr)

W JMK C
5
= 3.

25/07/23
Tuesday

कालेंडर कारेचरिल एवर मर्यादा,
विरुद्धपत्रे कालेतम अपरिष्कृत॥७॥

[lec-1]
OS

- 1. Intro to OS
- 2. Process management
- 3. " Scheduling
- 4. " Synchronization
- 5. Deadlock
- 6. Memory management

[Units to Cover]

OS → ^{like} govt. / administration

has multiple departments (to do diff tasks)

like
interface b/w
hardware and ~~human~~ software

Process :- whatever we are doing eg: web browser, file system etc.

How to manage it ??

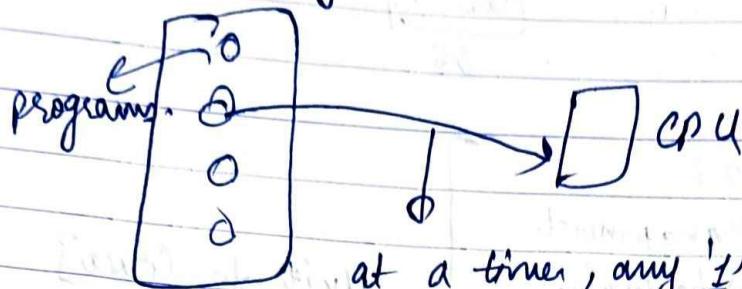
→ we'll study OS with '1' CPU, not that multiple OS parallelly!

→ CPU se jaldi run kene chalte ho, that thing must be present in RAM (Main memory).

in order to get executed, program requires CPU.

WJMK = $\frac{3}{32}$

- MM has collection of programs.



at a time, any '1' is
picked and executed;
then another !!

ya agar isko jyada time lg rha
hai, isko chod do ~~and~~, wait
kne do, and jisko km time lg
execute kro isko !

→ agar maanso ek program ke beech mein
sok ke, dusra chla diya ! → Then want some
management (OS Beta) → ki kon kab ~~k~~ jaaayega,
kab aayega [context switching].

→ then want something to save, jitna
kaam ho gya, so that baad mein can resume
from there !

CPU → should not be involved in management

↳ should be busy in doing meaningful
work !

↳ 'iski utilization
Synchronization
should be as
high as possible,

↳ iska dhyam os rikta.

WJMK $\frac{5}{5}$ 05.

→ 'as' do management!

• Process scheduling

↳

Job bhit sare processes ready for CPU,
then which one to pick ??

→ If want synchronize → then can't predict output!

f

eg:

a++;

a--;

a*a;

a/a;

d

agar

sequence

new pta,

tab kyun kaam kega, no control → can't predict

output.

maaro '2' ke baad.

(a-) me interrupt hdiya via OS,

then $a=1$, as $a=2$ save thodi hoga, tohi reflect hoga!

↳ (a--) \Rightarrow load a

Decr a

Save a → agar yhaan a++ ne first

interrupt hoga,
 $a=2$ save hoga.

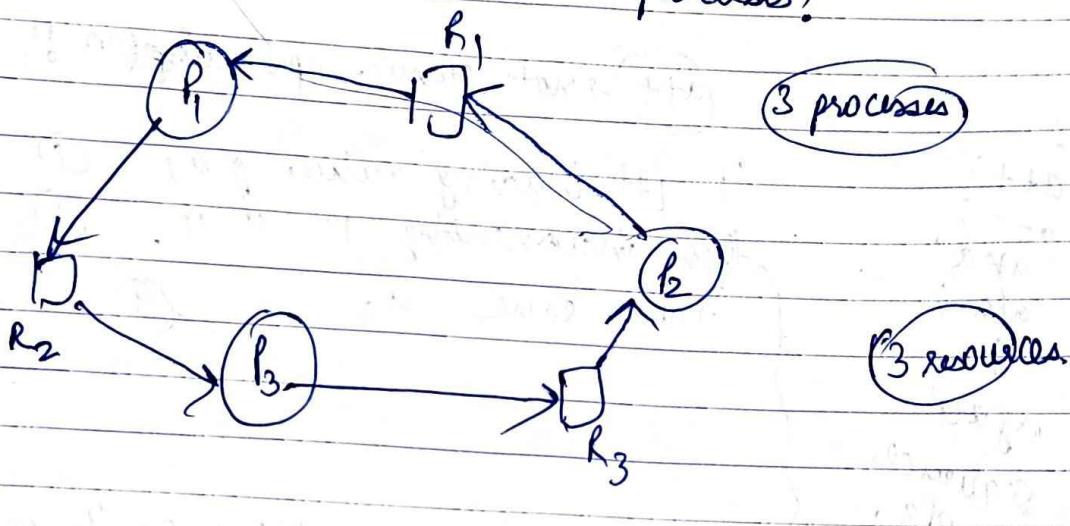
$a=0$, save!

then yeh aaya,

WJMk = $\frac{3}{5}$
 So esse bace chlta hai !
 a=0 same hoga yaa a=2,
 depend keta , last mein kaun sa chla,
 'at+' ab 'a--' ;

• Dead lock

Unlocked no. of resources & set of processes!



P₁ holding resource R₁ and P₁ ~~is~~ requested
 for R₂ which is held by P₃, P₃
 " " R₃ ————— P₂, P₂"

① releases P₃ Resource only when task is done,
 as P₁, R₂ nktta hi task kne ke liye
 hai !!

P₁ → kelta jab meraa task ho gya, I'll
 releases R₁ & R₂ both, but pchle R₂
 chalne !

→ so there is cycle of waiting, esse scenario ~~called~~
 called 'dead-lock'.

WJMK $\frac{c}{3}$

- although P available, but CPU kuch nhi krega!
all I/O devices jo use ho na hain, ~~but~~
will be in waiting queue.
- Concept of memory management apply hota, when
memory of 4 GB, but application u want
to get executed is of 10 GB.

wo ho bhi jaab' hai execute.

↳ DMM (Dynamic memory management)

→ as user ko nahi pta, but program partially
load hota, pehle load kya tha
part and then chda, dusra part load
kya sun kya!

→ User ko gta 'so' memory hai!

Book

→ galvin + gagne, silberschatz
(os concepts)
(9th edn)

WJMKG
= ₹
= ₹ 60.

26/07/23
wed:

[lec-2]

काले तंत्र अर्थात् स्कर
मरियू,
निरसुक्ति कालेत्वा अपरिणाम.

• What is OS?

→ program that acts as a intermediary /
interface b/w a user of a computer
and computer hardware.
goals?

(1) execute user programs & make solving user problems easier

(2) make computer system convenient to use

• Computer System (CS) can be divided into 4 components:

(1) Hardware - provides basic computing resources
• CPU, memory, I/O devices

(2) OS - Controls & coordinates use of hardware among various applications and users

(3) Application programs - define ways in which system resources are used to solve the computing problems of users

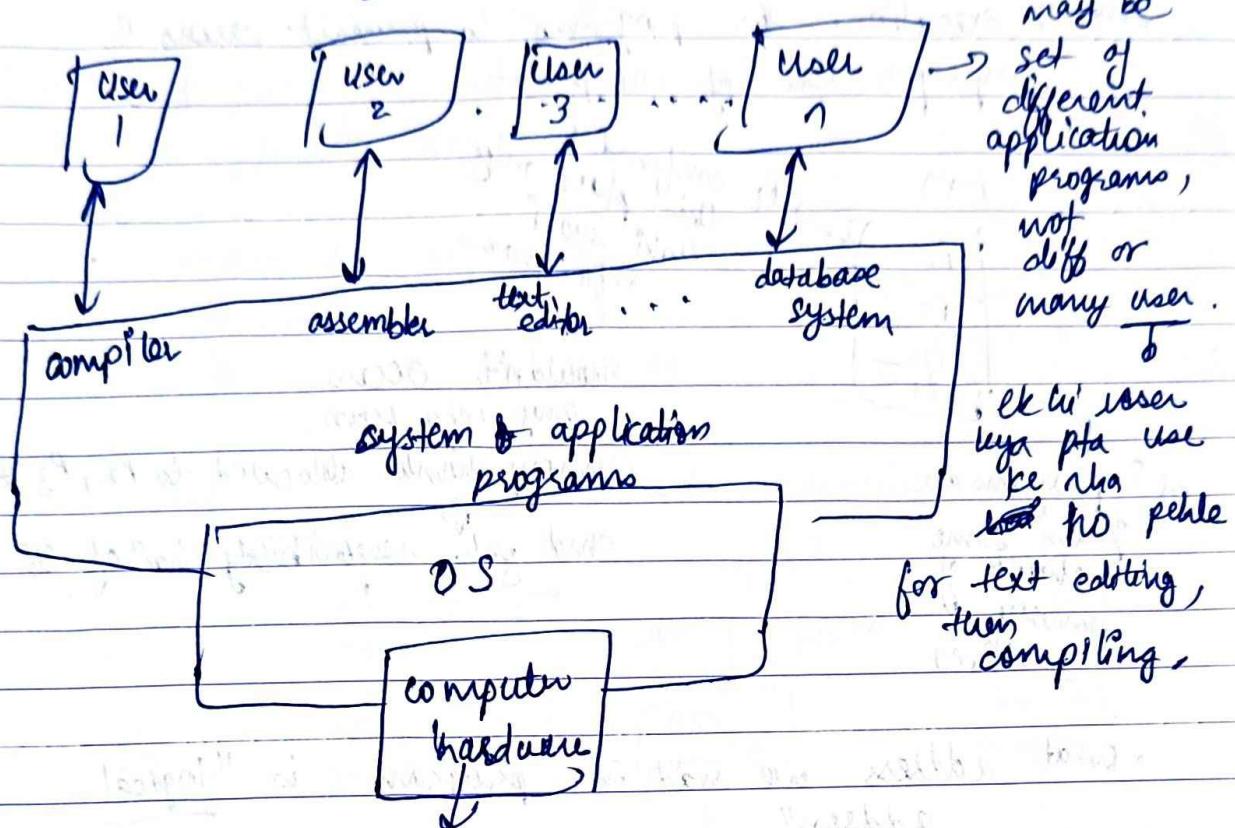
- word processors, compilers, web browsers, database systems, video games.

(4) Users - people, machines, other computers

• Aircraft also has OS at autoplane mode.
??

WJMK = $\frac{C}{M}$

Hardware - very difficult to manage by common user,
that's why, we've OS.



To use this, we design OS

also not easy to manage by
common user

so have programs

For OS, every application program is user, as will take
help of OS.

we have limited resources
connected to PC

• What OS do?

→ depends on point of view

→ User wants convenience

• OS is Resource allocator, manages all resources, decides
b/w conflicting requests for efficient & fair resource use.

WJMK

- OS is a control program.

control execution of programs to prevent errors & improper use of computer.



P₁ confined to
this place only
use ki saare
info share

b
shouldn't access
any info from
memory chunk allocated to P₂, P₃ & P₄
and yet responsibility lies of OS.

all programs are
given some
chunk of
memory
locations

- What address we read in program is "logical address".

- OS do mapping b/w physical and logical address.

a big system application written by some professional vendor or organisation to manage computer system.

- Kernel :- one program running at all times on the computer.

• "Everything" a vendor ships when u order an OS is good approx.

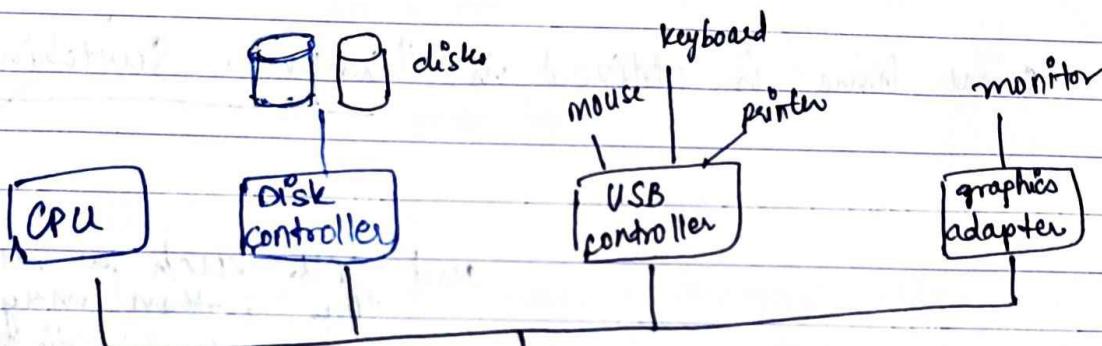
• But varies widely

• Everything else is either :-

- a system program (ships with OS), or
- an application program

- do in MM → do in MM → non-volatile, when power-off, it will go off.
 OS → not in MM → WJMK = $\frac{1}{3} \text{ sec}$, WJMK = $\frac{1}{3} \text{ sec}$.
 (needs in secondary memory) → go off.
 * we give a fixed slot to OS of main memory, other slots to other users; ↓
 (usually initial slot) secondary
- ① bootstrap program - loaded at power-up or reboot
 - Typically stored in ROM or EPROM, generally known as "firmware".
 - ~~Initially~~
 - initializes all aspects of system
 - loads OS kernel & starts execution.

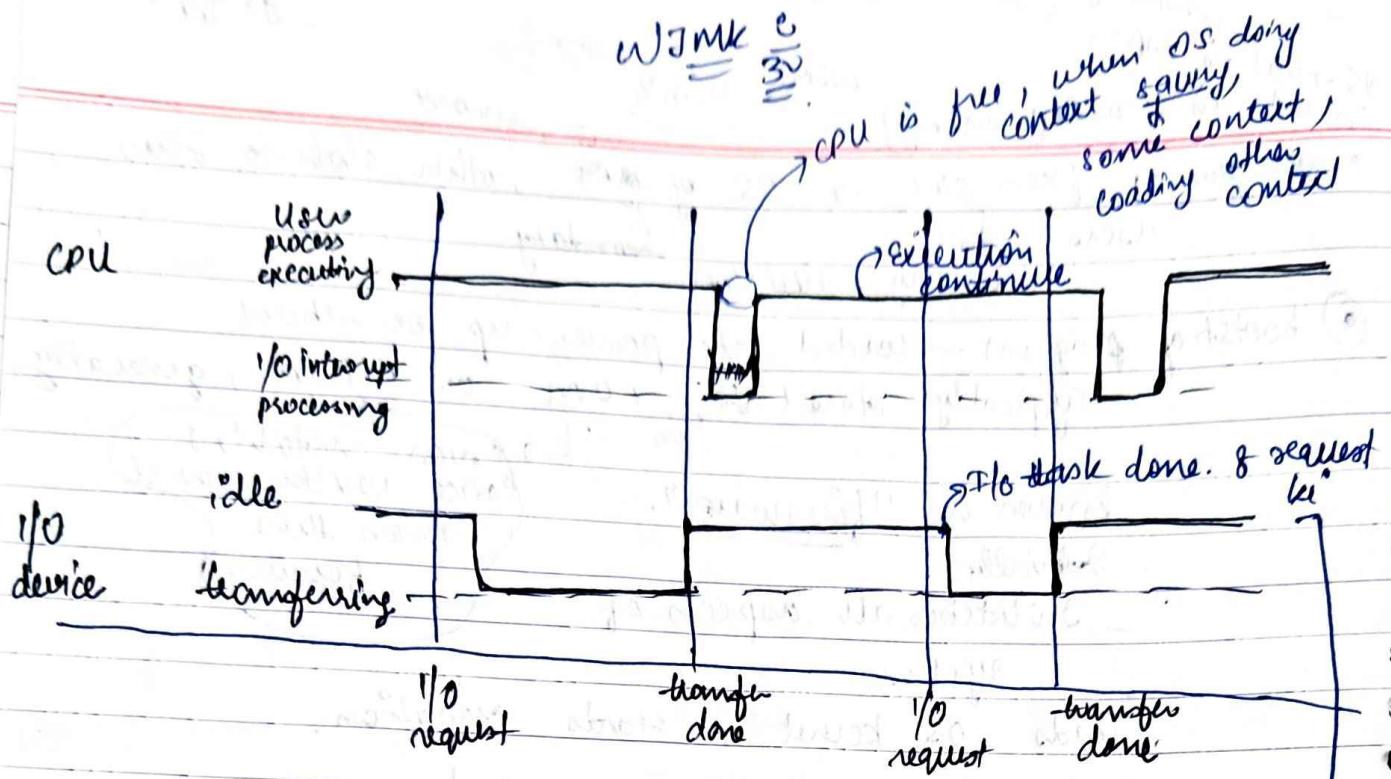
(non-volatile)
 Once written, will remain there forever



- Interrupt - means OS is stopping somebody
- I/O devices and CPU can execute concurrently.
 ↳ har device ke kya device controller hota, jo karta hai ki haan task complete ho, gyar CPU ko,

OS → can intercept any program / application, any time.

• Interrupt Timeline :-



and CPU kuch koi nahi
thaa, then may or
may not load the process
jisme request ki.

I/O Structure :-

System call :- request to OS to allow user to wait for I/O completion.

- speed $\uparrow \Rightarrow$ cost $\uparrow \Rightarrow$ volatility \uparrow (?)

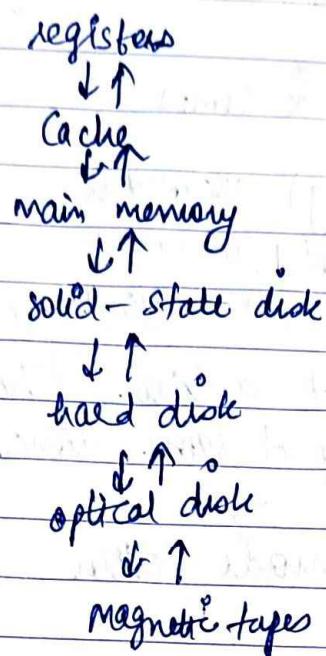
• Caching :- (tmp to speed up in CPU or system)

↳ copying info into faster storage system; MM can be viewed as cache for secondary storage.

Cache ke MM bhi state (as then cost will \uparrow but jyada)

$$WJMK = \frac{C}{3M}$$

- What should be kept in cache memory & how to manage it?

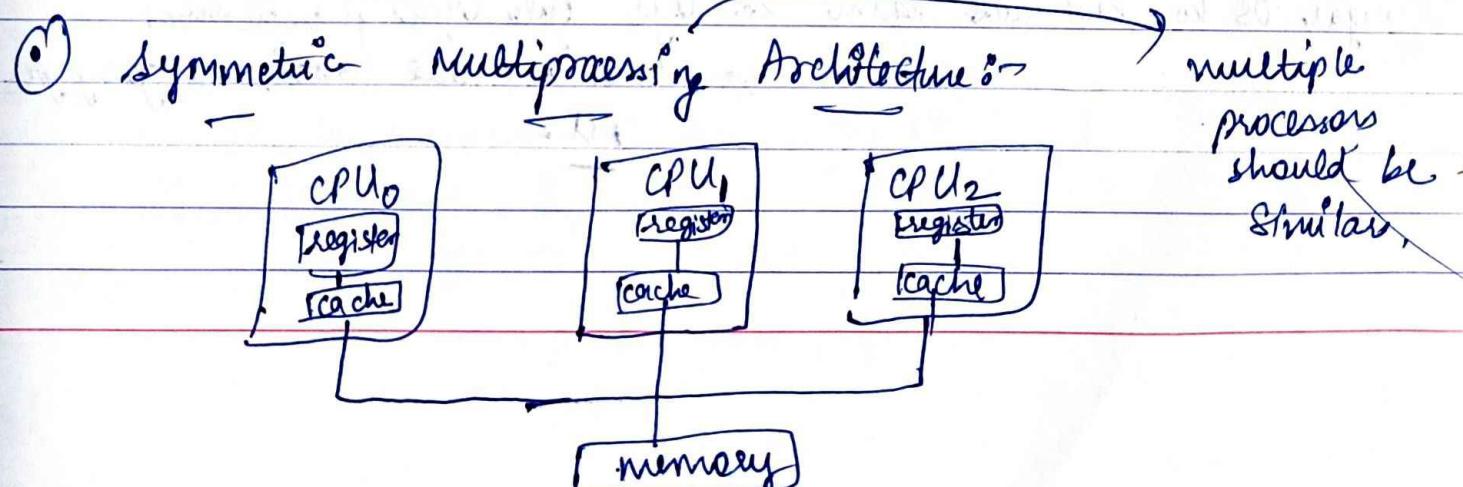


• Cache - Small in size

↳ so must have good replacement policy, as kuch nyaa dhaalna, toh kuch replace kera phega? kya? → wo mat ke dena, jo jaldi chahiye! → ~~so~~ usko nikaalo, jo jaldi mhi "

• DMA Structure (Direct memory access)

↳ one interrupt generated / block, rather than interrupt/byte.



WJMK $\frac{5}{30}$
= 17.5%

- We'll mainly consider single-process architecture.

① Multiprogramming (Batch System) \rightarrow architecture needed for efficiency

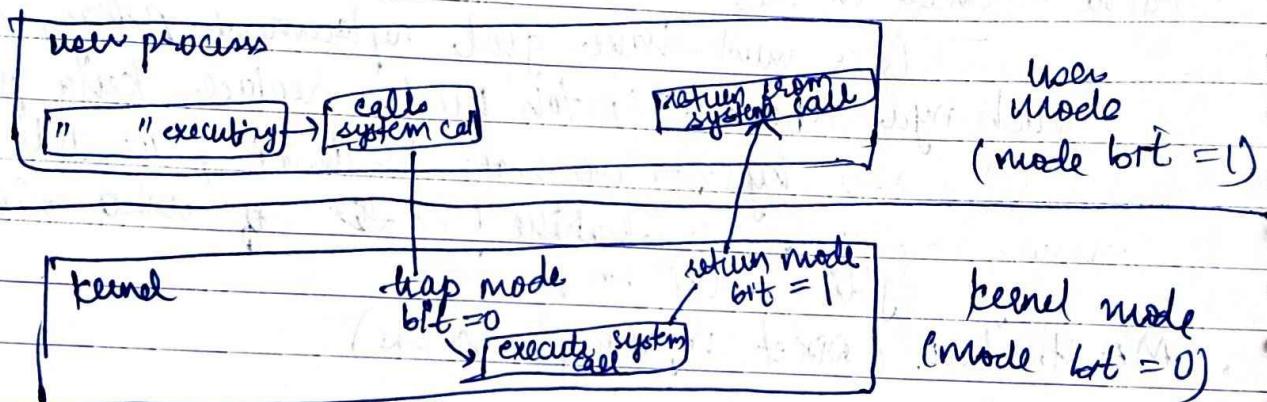
* Job scheduling * (IMP)

② Time Sharing (multi-tasking) architecture

- actually CPU switching b/w contexts, running only 1 program at a time, but we think many "running at same time."

- Computer runs in 2 mode either kernel or user mode.

↳ generally means data,



- agar OS ko kisi aur kaam ke liye bula diya, user mode se kernel mode mein aa jayega CPU.

2/8/23
wed.

lec-3 OS

ch-3

Processors

कालेंट्रो कार्यपाल द्वारा नियंत्रण,
परम्परातेर, कालेंट्रो अपरिस्थितीय.

- For us, program or write is a process to be executed / loaded on CPU.

But in actual, ese nahi hota! (Kuch aur addit. feature bhi hote hain)

Firstly we compile program and compiler generates .out file

Task of OS → not to compile file

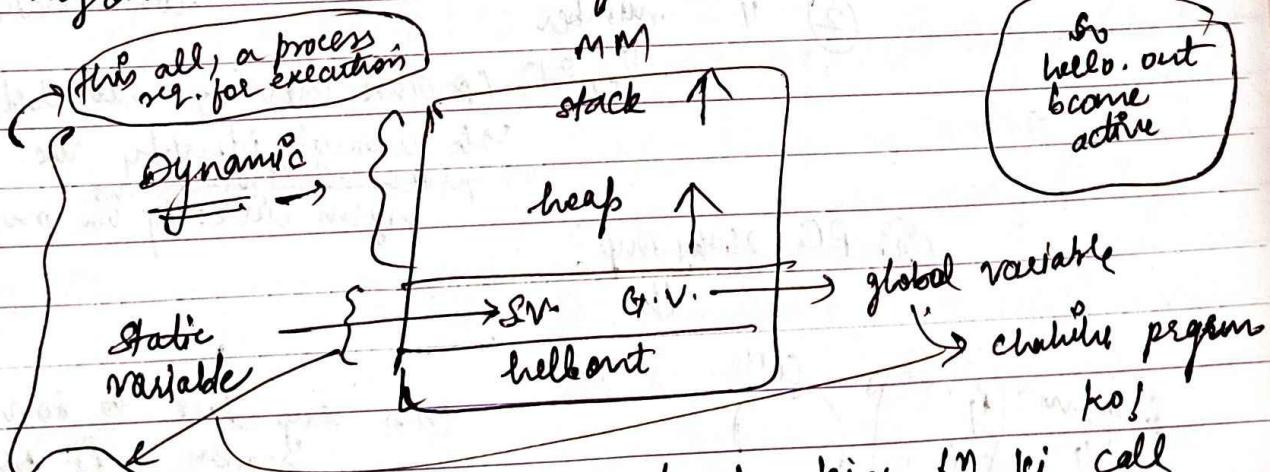
OS \leftarrow hello.c \rightarrow beside in secondary memory
hello.out \rightarrow yeh OS ko snap aata!

\rightarrow compiler yeh load kega in MM body.

Note all (programs) in MM - are process, us change, jo execution mein hoga!

like songs \rightarrow sab get body \rightarrow become active

- Program contains "set of instructions"



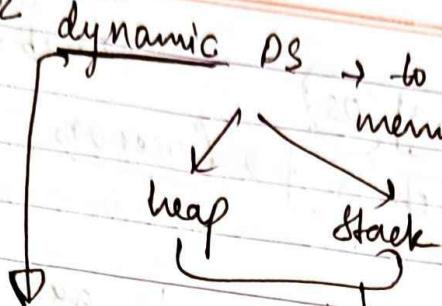
Data section \rightarrow Stack req. to keep track incs of in call

complete & ab incs ko call kiya complete, tabs chla gaya & control will go to b1()

stack (f2(),
f1(),
main())

WJMK

Q1, Q2



→ to have dynamic memory locations

stack

means it will grow! → so both will grow!

→ so difficult to set boundaries of heap or stack. Iya pta kyi voar stack tam space le, heap ↑, so, ese grow kرواے! -

stack ↓

heap ↑ where they will overlap, stop!

DD Card → chahiye have process ko, as ~~but~~ but same process

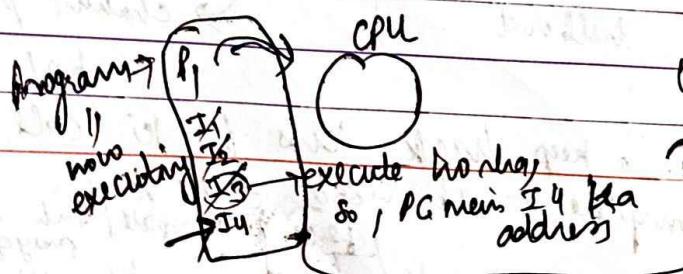
PCB (Process control block)

- ↳ contains:
- (1) Process State
 - (2) " number

to uniquely identify the process
ke use kuch toh chahiye,

"(pid)" (process identity number)
to uniquely identify the process in your program currently in process.

③ PC → Why msp?

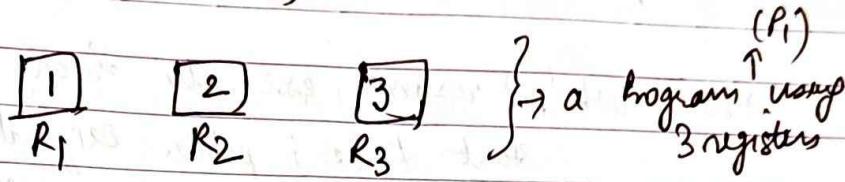


let's say due to some reason, P1 stopped & P2 load, this become msp. to save PC info some where of w, P1 kab aghi voar load hogi, u don't know

W.M.C
= 5000.

from where to start?

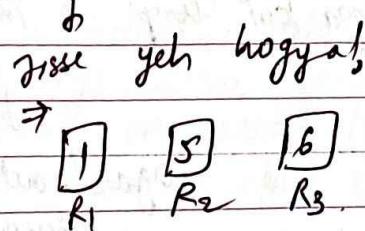
(4) registers (GPR) general purpose registers



want to execute this execution $\Rightarrow R_1 = R_2 + R_3 = (I_2)$

Just before executing this,

P₁ stopped, P₂ loaded!



$\Rightarrow P_2$ completes its execution,

then again P₁ to load kya?

PC \rightarrow helps process to resume its execution

$\hookrightarrow I_2$ se must start:

$$R_1 = R_2 + R_3$$

$$R_1 = 11$$

But

$$R_1 = 2 + 3 = 5 \text{ hona chahiye tha?}$$

\hookrightarrow so correct oprn nhi hoga, as values of R₂ and R₃ are changed by some other program!
values are inconsistent.

\Rightarrow info of registers also needs to be stored!

\hookrightarrow contains values of registers, which this process (P₁) was about to use!

W.M.K. 34.

Registers
 $R_1=1, R_2=2, R_3=3$

→ So if job (P_1) resumes, esse nlr directly execution
start kede; pchle original values ko
copy krega!

(5) memory limits! :- valid boundary for particular
process, ~~where~~ within
which it can access.

OS → dega koi 'trap' to indicate that (violating)
memory space given to you

agar 'out' file generate some address
beyond the memory limit.

(6) List of open files! :-

if 'reading' files parallelly from many
processors → no ~~time~~ problem.

if 'writing' → then permission must be
given to 1 process at a
time, o/w problem!

(7) I/O status info

OS should know kine request ki I/O
device ki & konca process in execution mein
hai!

→ kya phr job service available, kaisa processor ho
kii ha in CPU

→ so OS should know/do correct
mapping!

WJMK
to mm.

CPU scheduling info! -

+ all processes are not equal!

↳ so CPU may give priority to some set of processes over others)

user process & system process
Priority wise

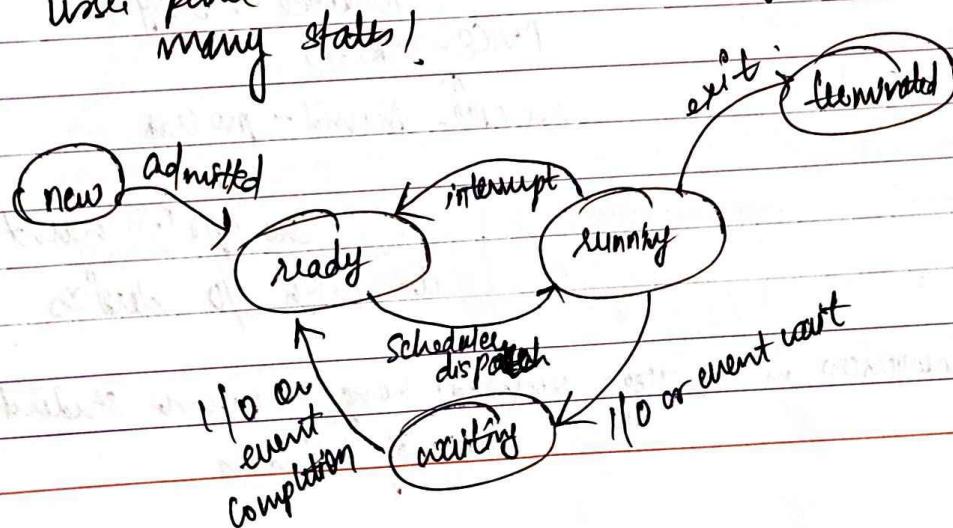
or OS

↳ Is mean also priority mean vary!

Process state :- states through which process can go through lifetime

- (1) new : process just created.
- (2) running : process now in execution, got CPU.
- (3) waiting : busy interrupt by OS or as high priority process arrived, or waiting for some I/O service
- (4) ready : process is ready for execution, but CPU busy on some work
- (5) terminated :
 - ↳ due to some problem / or when execution gets finished!

↳ Zara kisi nahi
ki program CPU ke pass ho & execution mein ho, uski pehli or bechhi mein it go through many states!



WORK = $\frac{S}{T}$ m.

• waiting se directly running state mein
can't go

↳ as CPU abhi kuchh aur ker na hoi,

running → ready

interrupt

↳ why?

⊕

kya pta zyada

let scheduler,
scheduling
execution on
basis of
Priority

priority wale aa jaaye, to abhi execute
hoga, usse ↑ priority wale aa jaaye

objective of OS → to maximize CPU utilization

I/O bound process → process which want less CPU,
more I/O device service

⊕

RAM mostly CPU → empty

I/O device → busy

(VICE-VERSA)

in CPU bound - process

↑

less CPU want to
do with I/O devices

• All resources in college utilized honge → when student have
mix of interests

W₀M₀ e₀

Q so will OS prefer CPU-bound or I/O-bound processes??

Ko kha saare resources
Utilize kro,

Krega only if process chalte hain!

Abh esen toh nhi kah skta, OS ki maini process tab execute hone dunga, jab aap kuch ~~print~~ print kro, bcoz I have printer. NO!

but
how to do
this?

① showed some state of Process (P_0) meanwhile, esse nhi ki siddha stop kde " ko & dusra load kde!"

and

reload state of (P_1) from PCB, in CPU.

as isko bhi pehle kahi roka hoga, ab want to resource!

tab tak

CPU → idle hoga!

tab tak ek ki state store karne hain & dusre ki load!

as
not
meaningful,
work nahi
hognay

Garsoni
new job,
baethi main
Starting se
have to
resume

ek context
store
(a.k.a.
context
switching
&
load)

called
Context
switching

WJMR = 30

some of them
placed in my
ready queue!

- Scheduling queues:- set of all processes waiting in secondary MM

(1) Job Queue

→ say first come first serve

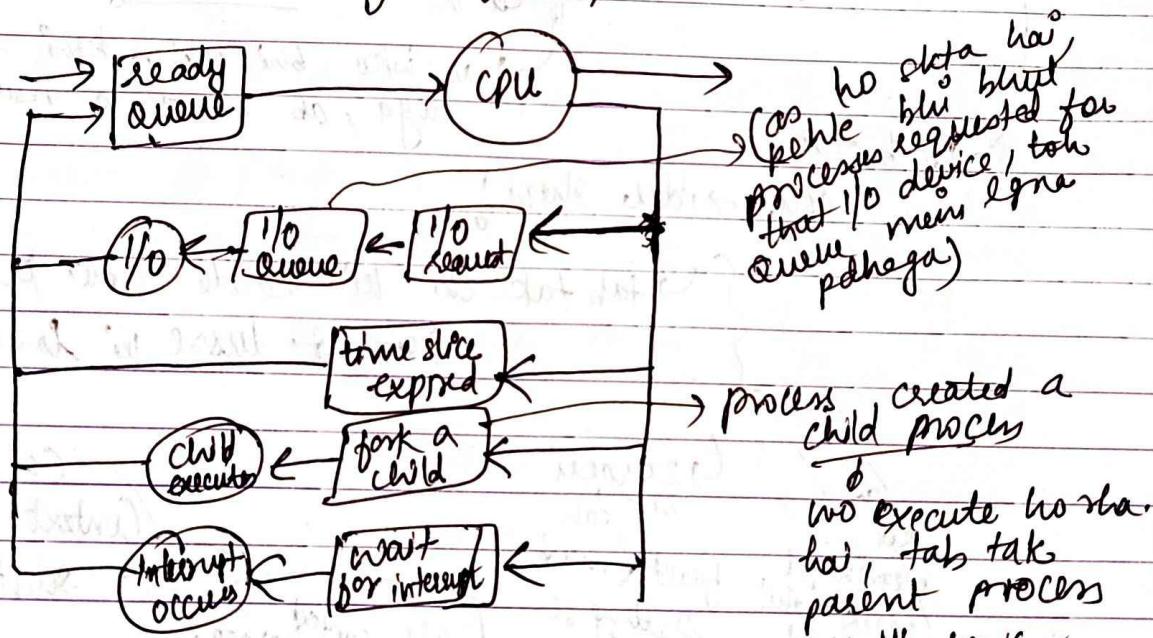
main mein execute ho sha hai &
maine abhi bhega process, toh wo
last mein lega queue mein

(2) Ready Queue

(3) ~~Device~~ Device Queues

many processes waiting for an I/O device
to do I/O operation

Queuing Diagram / Flow chart



Scheduler

long-term
(Job)

short-term (CPU)

dependency on frequency

of their use / scheduling or
since they get to take decision

short-term: cannot have simple algo for scheduling process

↳ as get very less time to think what is correct / ~~doing~~ wrong!

• Degree of multiprogramming

↳ as long-term wala decide secondary memory se kitne ~~to~~ MM mem select ~~keinge~~ krega, say '3' krye!!

so degree of multiprogramming = 3

↳ as user ko illusion hoga, (3) concurrently can execute, jo nahi hota!

(ham switch pte hain ek se dusre mein)

Interprocess communication (IPC)

(multiple processes want to communicate with each other)

- Two modes of IPC!:-

① shared memory! eg. Projector

↳ (one way to perform IPC among multiple processes) → token (pipe) passing

② msg passing! → socket passing

↳ eg. WhatsApp msg & channel B ek se dusre tak paas hata msg.

→ only 2nd can communicate at a time processes

$$WOMK = \frac{C}{30}$$

- before C1 : mostly focus on kernel
- lecture : • process management
• thread scheduling
• synchronization
- mostly in lab : • Thread and IPC
• system call
- sub 'C1'
main
ayegal

C2 will have applications (e.g. web, etc.)

- (e) mark existence of web server
+ function of porting and optimization
of the kernel
→ (above English)

कालेंट कार्डिनल स्टर मैट्रिक्स,
नेटवर्क वेब कालेंटर्स अप्लीकेशन।

WJMK १०
३०
०९.

8/8/2023
Tues

Tut 2

① Process :-

- A program in execution
- will give us running state of program.

② Process components :-

- ① Program code :- actual instruction
- ② Data :- will give variables, data structures etc.
- ③ Stack :- to store the data, we need stack.
- ④ Heap :- Dynamic allocation of memory.

PCB (Process Control Block)

↳ is a Data structures managed by OS.
↳ gives us very important info, like:

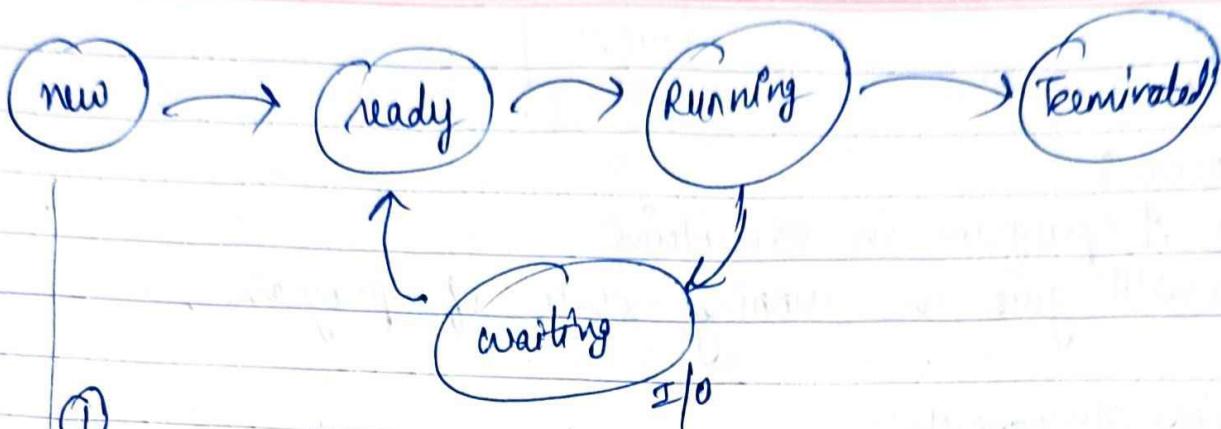
①	Process ID
②	Process State
③	Program Counter
④	Registers
⑤	Scheduling Info

which
next process
will involve

unique Id process has
been process → running,
waiting, ready etc.

which process will
involve first, then
second...
e.g. FCFS (first come,
first served)

Process State Diagram :-



① In 'new' state :-

one new process is created and ^{that} process wants to get admitted in the system.

② ready state :-

↳ process will enter in the system and waiting for CPU allocation.

③ Running / ~~Allocated~~ CPU :-

↳ The process got [?] the CPU and currently it is running.

④ Waiting :-

↳ unable to continue executing because it waits for some event that is not present in process
eg : I/O operation

⑤ Terminated :-

↳ after executing the process finally process completes its execution and ready to remove from the system

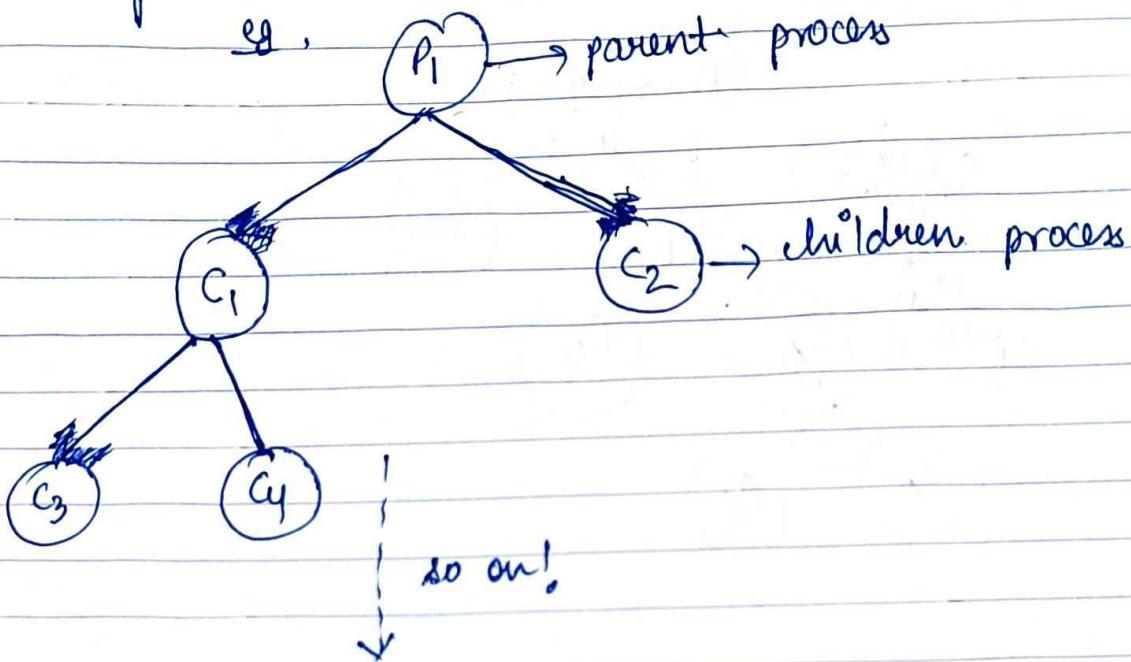
then a new ^{process} ~~system~~ will create & cycle
~~whole process~~ ~~will repeats!!~~

WORK
50
0.00

① Process Creation :- It is used for creating the new process.

↳ process can have sub-processes

e.g., $P_1 \rightarrow$ parent process



② There are 10 steps for process creation :-

① Allocating memory :- To create a process, we need memories to store the data.

② Loading program codes :-

↳ After that, we need a program code / process ~~code~~ code to execute.

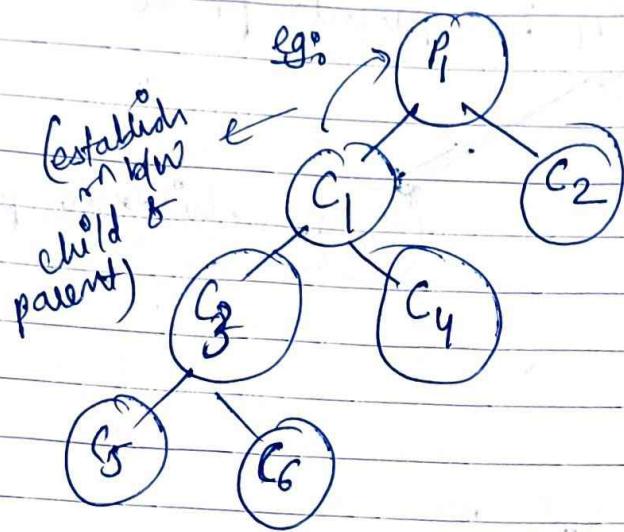
③ Setting up Data structures :-

To manage, we need various data structures.

e.g.: PCB (process control block)

WJMIC
30
20

- ① setting up the initial state : initiating the 1st step
② Establishing process relation :-



- ⑥ Resource allocation :-

↳ for further execution :-

e.g.: I/O, O/P streams.

- ⑦ Scheduling :- e.g. ① FCFS (first come first serve)

② priority scheduling etc.

- ⑧ Execution :-

After scheduling, the created process will be executed.

- ⑨ Terminated:-

After execution, the process ~~got~~ got terminated from the system.

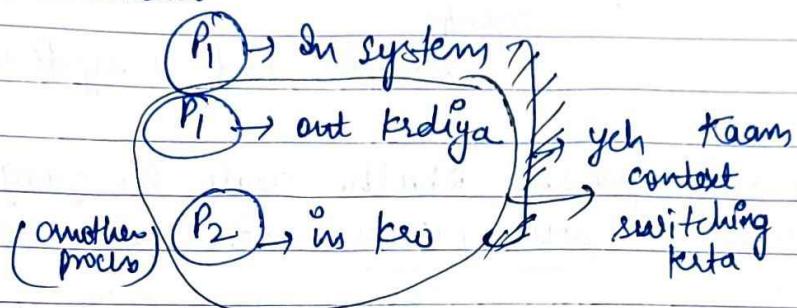
WORKS
= MM.

⑩ clean-up :-

After termination from the system, system cleaned-up and works for another processor.

Context Switching :-

↳ switches context

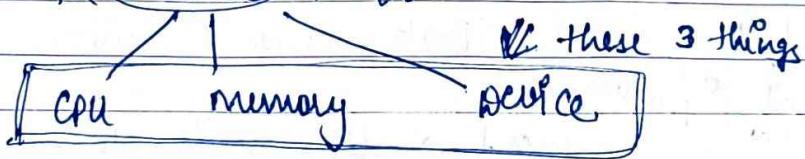


⑪ System-call :-

↳ Through this system interacts with process.

↳ (f^m) → interacts with kernel → e.g., to control

or method.



⑫ what is system call?

↳ it is a method or function for a computer program to request a service from the kernel of the OS on which it is running.

API :- (Application Programming Interface)

↳ links the OS and a process, allowing user

(act as) (b/w)

level program to request OS services.

→ common & mostly used API system by

OS → is → POSIX API

$$WOMK = \frac{C}{30}$$

```
printf("Hello ");
```

Library
K
andar
hata

↳ uses write() system-call

↳ Hello

yeh API sunjata

kiss system call
ki° zaronat
hai° g usse
bulaata.

Scant

↳ requires `read()` system-call

* ese nhi process khudh create ho jaaye, pehle
wala process se create hota!

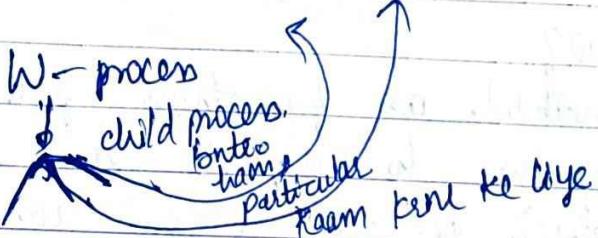
↳ known as init process

fork() yeh create hota jab OS start kete ham, then issi aur child process bakte jaate hain.

fused to create
child process

used to create
child process

Word → edit, write etc.



→ have 2 parts: parent, child

↳ basically duplicate!
creates new child process, The init process is "ancestor" of all

- ① fork() parent child process
 - ② exec() → makes a process execute a given executable
 - ③ wait() → causes a parent to block until child terminates.
 - ④ exit() → terminates process.
 - ⑤ open(), read(), write(), close(), chmod()

↳ Basic system calls.

Types of system call:-

- process control
- file management
- device "
- information "
- communications

Process Control:-

[fork();]

int main (int argc, char *argv[])

{ printf ("HelloWorld (pid: %d)", (int) getpid()); }

int rc = fork();

if (rc < 0) → (child process not created)

else { if (rc == 0) → (child process created); }

printf ("I am child (pid: %d)", (int) getpid());

}

else {

printf ("I am parent (pid: %d)", rc, (int) getpid());

(?) exit rutu likha, but as ending bracket, API ko pta chl jata, ab end ho gyab

Pl.C → ka pid = 162

fork() bawaya.

parent, child
pid = 163.

but can be still available in process list ⇒

WJMK 30 Jan.

isko zombie process kہلے
hain! → Now so that
child process acche se
execute hو skے, zombie
ko khatam kene ke lie,
we use wait();

Output:-

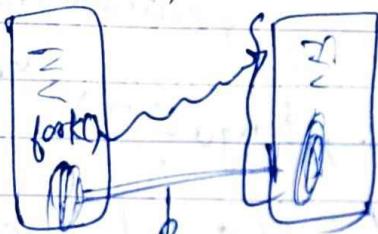
HelloWorld pid: 162

I am child (pid: 163)

① Jab ek parent se child execute hota.

Jab tak iski^o execution
khatam naa ho jaye
yeh block shega!

Parent (P) Child (C)



ab 'P' mein change kro, yaa 'C' mein,
ek hi baat !!

② process list! - helps OS

↳ usko pta chتا, kab konse
process execute hogen, taum kiiss
state mein hais!

next aage
wala
program
execute
ki
agar child
process
execution
barriate
horne k baad!

previous code

```
else { int we = wait(NULL);  
printf("I am parent (pid : %d)", rc,  
(int) getpid()); }
```

WSMK 3/20

① exec() → so that duplicate create new pro!

↓ and new part of program execute to stein!

② main()

{ rc = fork();

if (rc < 0)

fork failed;

else if (rc == 0)

{ printf("I am child (pid : %d); (int) getpid());

char * myargs[];

myargs[0] = strdup("WC");

myargs[1] = strdup("P3.C");

myargs[2] = ~~strdup~~ NULL; → file name.

execve(myargs[0], myargs);

}

else { int wc = wait(NULL);

printf("I am parent of %d (WC : %d)
(pid : %d)", rc, wa, (int) getpid());

}

(or current process)

\$ ls → this unix command creates a child

or any
other

so same
data fetch beta
parent ke data,

and parent se same
data, API par show
hota hai !!

WEEK 3
1.

कालतंत्र कार्यदृष्टि एवं मिश्न,
प्रिमिकरण का लक्षण अधिकारी।

9/8/23
Wednesday

lec-4

CPU Scheduling :- (Ch 6)

means a state of processes ready to execute

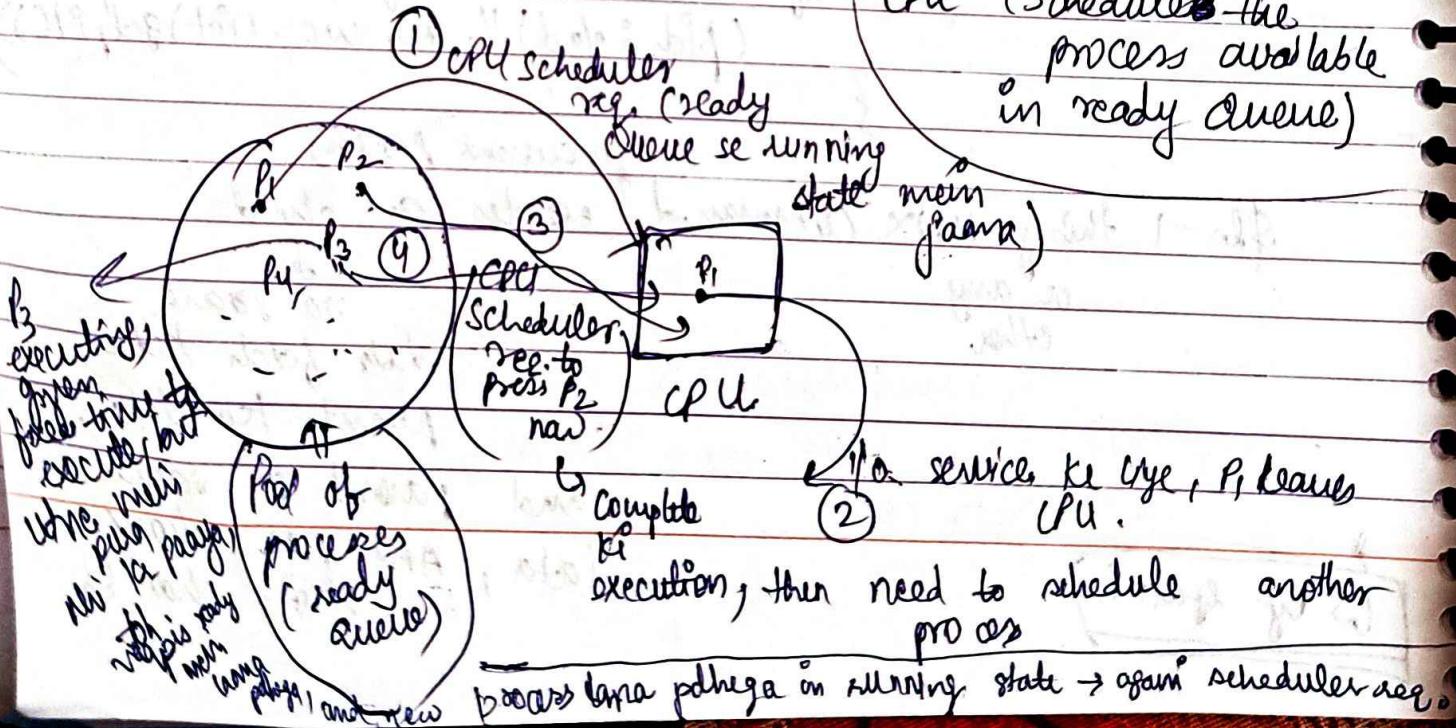
But ek vaar mein can choose only 1.

So need scheduling!

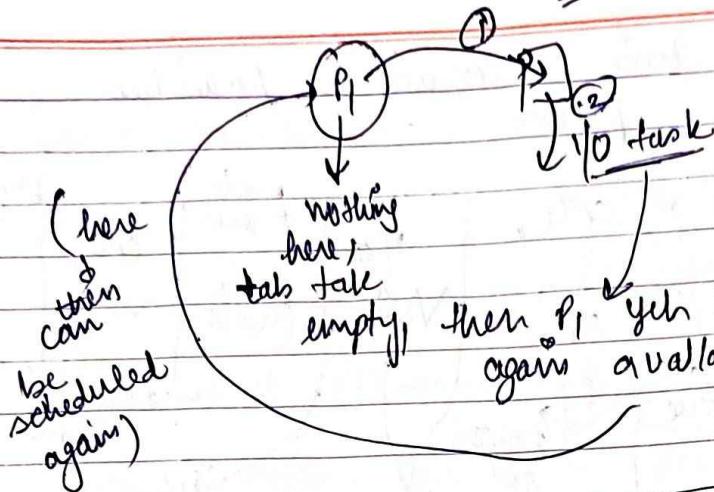
CPU burst → when process is executing in CPU
I/O burst → jab " ko I/O device / service ki operation ho, wait beta, jab service available, kaam complete keta hai & then go to CPU.

Scheduler

long-term → short-term
CPU (Scheduler) → CPU duty to schedule the process available in ready queue

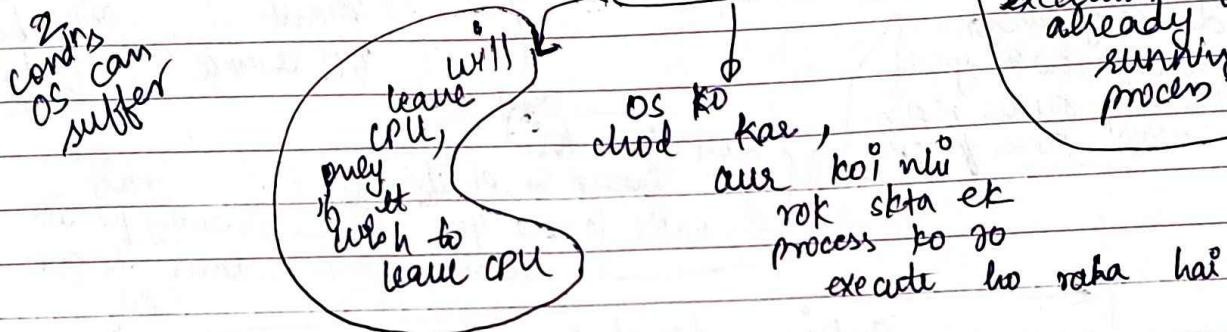


WJMK 3(MM).



{ process scheduling is **non-preemptive**.

some other process, may preempt the execution of already running process



- (1) Dispatcher?
- (2) Dispatch latency?

Scheduling Criteria

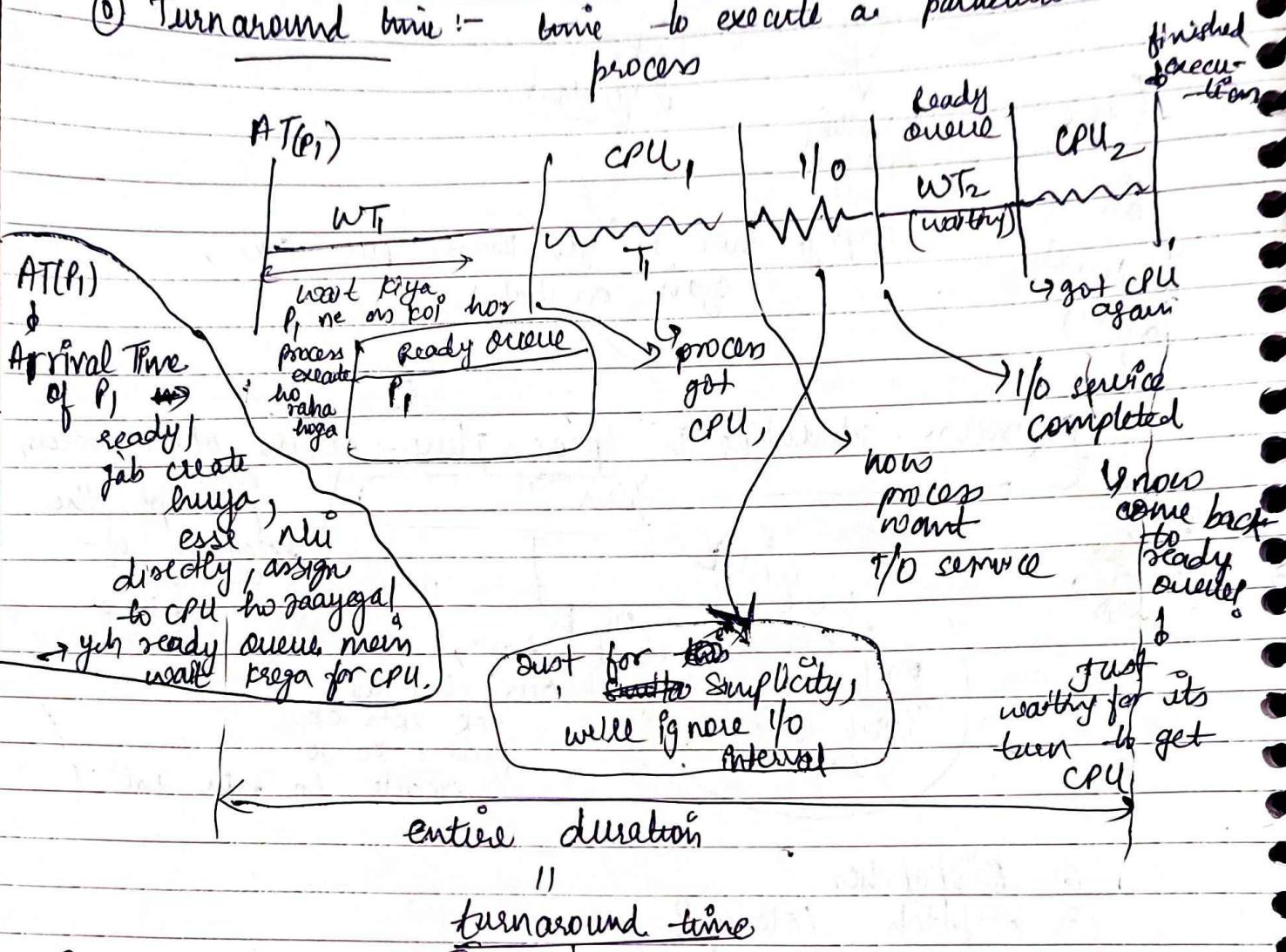
Objectives for designing scheduling! -

- CPU utilization:- keep CPU as busy as possible
- Throughput:- how many processes complete their execution in given time.

↳ as CPU busy
↳ some processes execute
↳ may generate more throughput.
Difference??

WORK ~~30~~ ~~20~~

① Turnaround time :- time to execute a particular process



② Waiting Time:- If want to reduce it, need to reduce waiting time.

↓ amount of time a process has been waiting in the ready queue (as ignoring I/O part, but actual mein use kya hoga bhi wait keta hai)

③ Response Time:- Jaisa ki process will generate useful output only after execution, user file can also generate!

WOME $\frac{10}{30} = \frac{1}{3}$

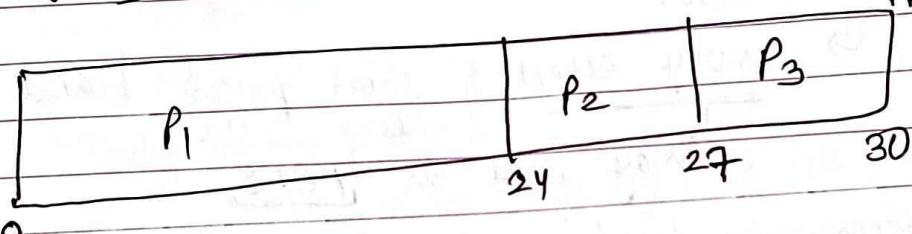
key main process generates useful output just after allocation in cell.

Burst Time is unit of CPU time, a process requires to complete its execution.

1) means, 3 units of CPU time req. to complete execution.

Assume, Arrival Time of all processes is same!

(FCFS)
All came at same time,
then we'll consider their ID! (or BT?)
suppose processes arrive in order:
 P_1, P_2, P_3 .



Waiting time for P_1

$$\text{Avg waiting time} = \frac{(0+24+27)}{3}$$

$$= 17.$$

Waiting time for $P_2 = 24$

$$P_3 = 27$$

Job arrives at time 0
execution begins at time 24
waiting time for P_1 is 24

Turnaround Time of $P_1 = (TAT) \circ P_1 = 24$
↳ 0 to 24

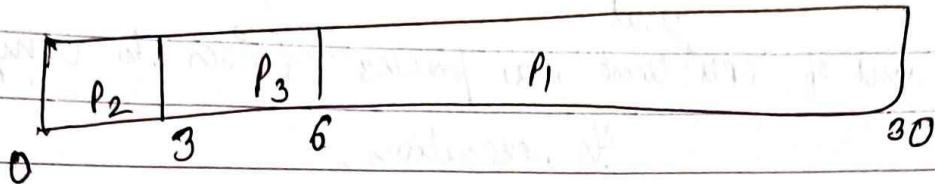
$$(TAT) P_2 = 27$$

$$(TAT) P_3 = 30$$

$$\text{Avg TAT} = \frac{(24+27+30)}{3} = 8+9+10 = 27$$

WJMk

① let's change order of execution :- P_2, P_3, P_1



waiting

time
for P_2

$$\text{avg. waiting time} = \frac{(6+0+3)}{3} = 3.$$

pehle (17) tha, abhi (3)

(Time difference)

so order matters a lot

④ Convey effect :- short process behind long " ".
may exist in FCFS.

so large process ko load mein karna (गो)

⑤ SJF (Shortest Job First) scheduling :-

↳ shortest burst time for execution wala

process (shortest process) pehle kro execute /

↳ but how to predict, kitna time mega execute kriye main, execute kene se pehle !!

↳ not possible to know, before execution

(benchmark)

↑
other
protocols
should
try to

Major Drawback !!

→ why designed them?

↳ bcoz will always give
optimal result!

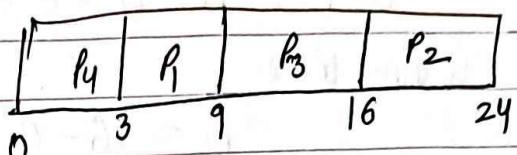
→ like gitne close gya sets hain, saarein !!

WORKS

Ex :
Burst Time

P ₁	6
P ₂	8
P ₃	7
P ₄	3

SJT scheduling chart



$$\text{Avg waiting time} = \frac{(3+16+9+0)}{4} = 7.$$

TAT $\rightarrow P_1 = 9$

$P_2 = 24$

$P_3 = 16$

$P_4 = 3$

Avg TAT = $\frac{9+24+16+3}{4}$

$$= \frac{24+16+12}{4} = 6+4+3 = 13$$

Completion

Time

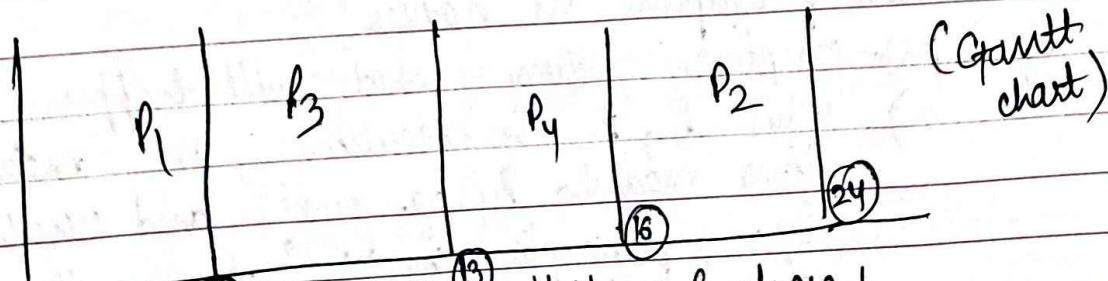
by No!

why looking similar?

↳ bcoz we are assuming, ab take, all available at same time, same starting se available han!!

	(AT)	(BT)
P ₁	0	6
P ₂	0	8
P ₃	4	7
P ₄	10	3

then TAT, completion may vary!



GP₁, P₂ available
T₂ select, bcoz has less BT
 $\rightarrow P_2, P_3$ available $\rightarrow P_3$ chose!
 $\rightarrow P_2, P_4$ available \rightarrow chose P₄

WOMK ST

$$WT : P_1 = 0 - 0 = 16$$

↑
waiting time

$$P_2 = 16 - 0 = 16$$

$$P_3 = 6 - 4 = 2$$

$$P_4 = 13 - 10 = 3$$

kab execute hone lga - kab aaya.

$$TAT : P_1 : 6 - 0 = 6$$

$$P_2 : 24 - 0 = 24$$

$$P_3 : 13 - 4 = 9$$

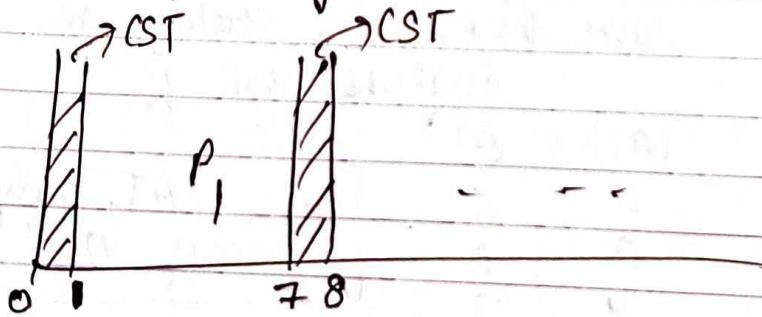
$$P_4 : 16 - 10 = 6$$

||

TAT = completion time - arrival time.

④ of 1 unit time req. for context switching (CST)

(context switch time) \rightarrow then, need to update gantt chart:



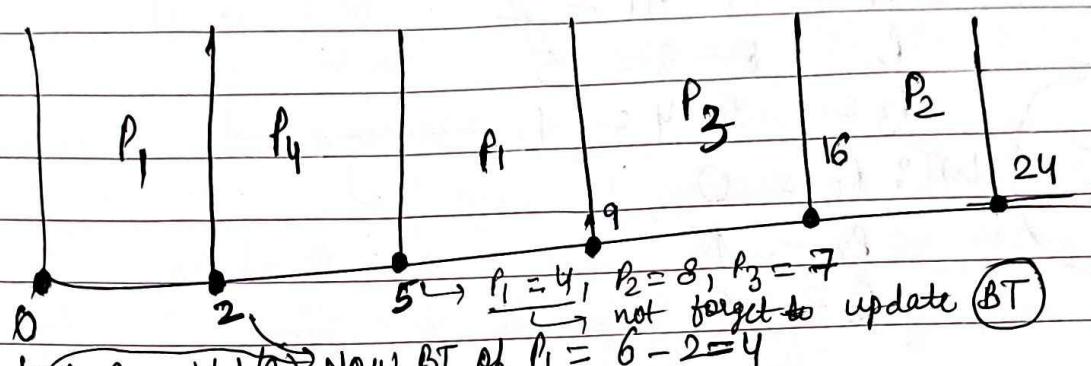
These all non-preemptive in nature!

If pre-emptive, then what will happen?

\Rightarrow while P_1, P_2 is execution, in mean time maalo by aa gyao! and scheduling algo, preemptive in nature, then what will happen?

NDSMBC
5/5

	Burst Time (BT)	AT
P ₁	6	0
P ₂	8	0
P ₃	7	4
P ₄	3	2



(say after every unit of time, checking state of ready queue, kya aoi new aaya?)

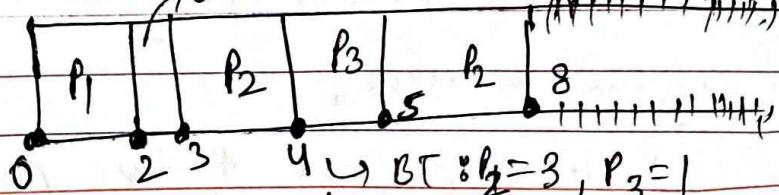
agar aaya ha!

may be jo execute ho sha ha,
use accha candidate aaya ho!

→ so how unit of time ke baad checking,
kam best candidate!

	GT	AT
P ₁	0	2
P ₂	3	4
P ₃	4	1

CPU → idle → as no process ready!



↳ our ready queue empty, nothing available

$$W_{\text{WTF}} = \frac{3}{50} = 0.06$$

$$\text{Completion Time} \Rightarrow CT : P_1 = 2$$

Time

: $P_2 = 8 \rightarrow$ when completed its execution

$$: P_3 = 5$$

$$WTF: P_1 = 0$$

~~P₂~~

$$WT = TAT - BT$$

$$TAT: P_1 = CT - AT = 2$$

$$P_2 = 8 - 3 = 5$$

$$P_3 = 5 - 4 = 1$$

★
TAT = CT - AT

$$WT: P_1 = 0$$

$$P_2 = 1$$

$$P_3 = 0$$

① Priority Scheduling \Rightarrow

Priority no. (integer) ass. with each process.

smallest integer \leq highest priority

0 has \rightarrow highest priority.

lower value indicates higher priority !

\leftarrow (non-preemptive) \rightarrow (pre-emptive)

SJP \rightarrow also Priority scheduling
 priority = inverse of predicted next
 CPU burst time.

starvation \rightarrow lower priority processes may never execute

↳ soln? \rightarrow aging technique!

as time progresses \uparrow the priority of process

after waiting time, increment crosses certain value (then decrement priority no. by

WORME 30
30

by \pm , means increasing priority by

↳ if we get tie, then FCFS.

↳ one which comes before is executed if priority no. same.

Priority no.

$$P_1 = 100$$

↳ not getting chance, to execute
↳ starvation !!

↳ if age of process in ready queue reaches ~~20~~ 20 min, then after every ~~20~~ min, ↓ it by 1 or any other no.

↳ (aging time)

then after how much it will definitely get CPU ?

↳ if priority no. = 0

↳ AP bn gyi !

$$a = a + (n-1)d$$

$$0 = 100 + (n-1)(-1)$$

$$100 = n-1$$

$$\Rightarrow \boxed{n = 101}$$

① RR (Round Robin) \Rightarrow

~~fixed time~~

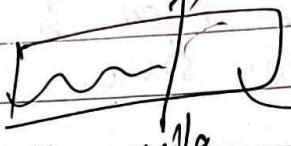
PCFS with fixed time
slice to each
process

may complete
execution & leaves CPU,
or may not completed but
time slice has expired, may have to
leave CPU!

WORKING

- ① one at front/head of queue → is getting chance to execute!

→ execution complete, chodega CPU,



→ pot aur execute hone lg jaaega,

~~acc. to FCFS!~~

- ② rear of queue pe kya hoga??

→ do currently executing?

↳ Round Robin

↳ preemptive (By default)

↳ as esee ho skta,
pura execute karne se
phle, chodna
phle CPU,

→ Time slice

↳ of very less!

↳ very large

↳ very similar
to FCFS

↳ jo aa sha
kar, complete
karke
nikal skta
hai!)

Must >

Context

switching time!!

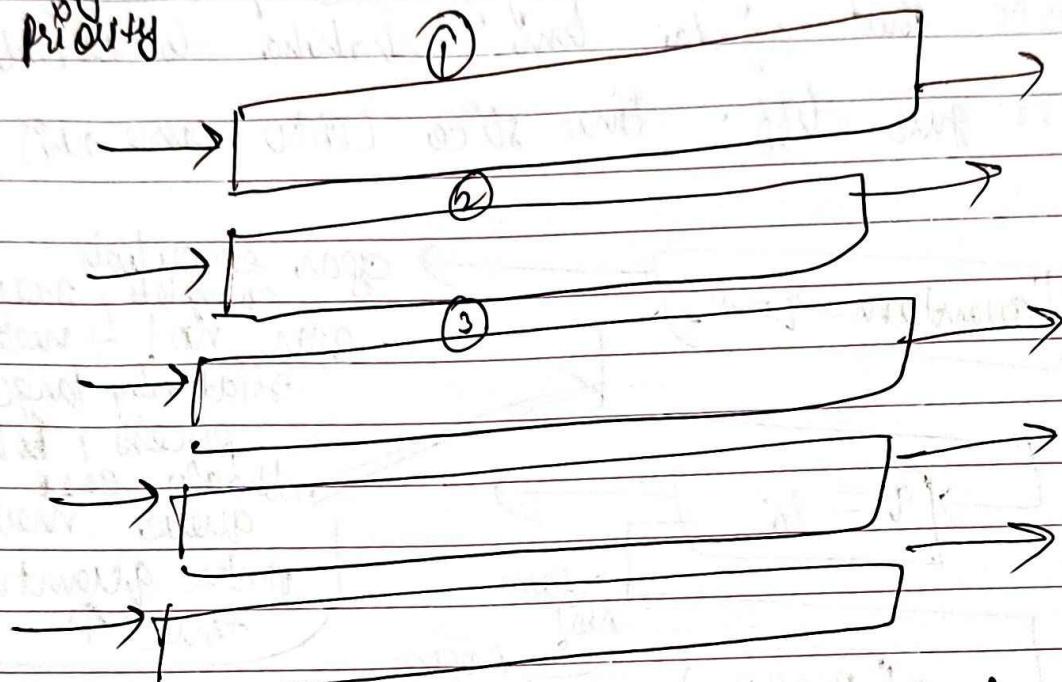
↳ o/w overhead may be too
high!!

(RR ka response time better than
SJF)

WJMF

① Multi-level Queue Scheduling :- (Priority Scheduling & FCFS ka mixture)

highest priority



lowest priority

ii) (multiple queues)

② process picked from ^{list of} queues (jisko ↑ priority)

→ agar ① 'khaali', tab ② par
daayenge, agar ② "", tabh ③ par
daayenge!

→ Then us queue mein se konsa, process
lena, i.e. acc. to FCFS !!

③ Multilevel Feedback Queue :- (modification of RR)

1,2

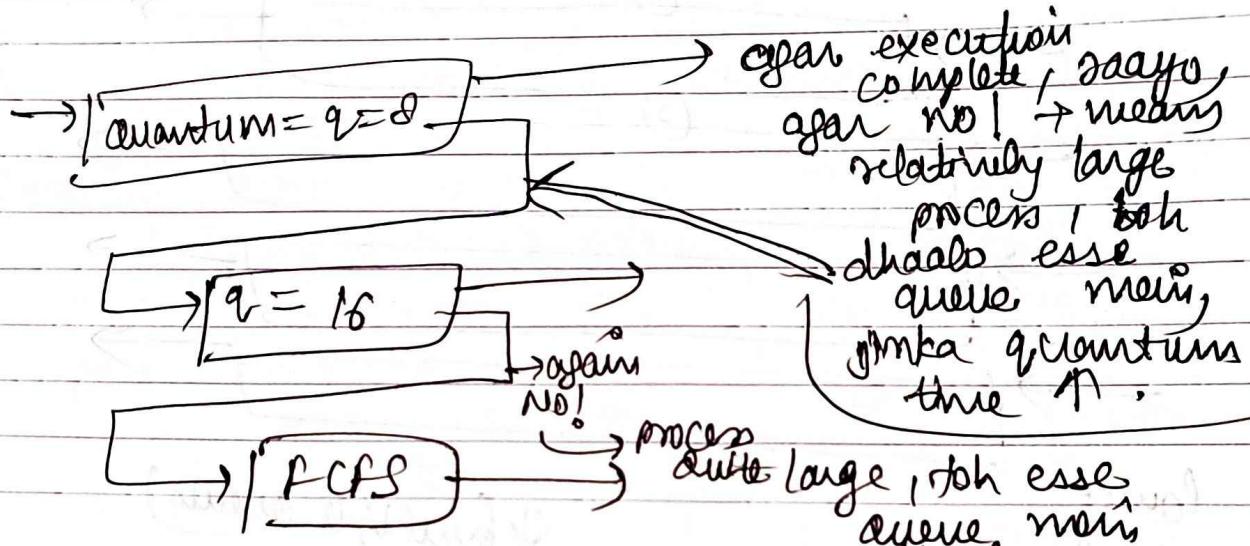
WDM₁ = $\frac{e}{30}$ m.

RR: gives fixed time slice, hote chlo execute!

↳ but hain vo context switching, ho jaayegi us process ki

disko hain jyada time chahiye to execute!

↳ so give diff. time slices (86% same n%)!



ab paka

execution complete ho

hi jaayegi!!!

WJMK $\frac{e}{\pi} = \infty$.

16 Aug | 2023
Wednesday

कालौंडर का सिंचारित स्नाद भरिएँ,
नियमजनक कालौंडर अभियन्त्रित।

[UC-5].

- ① Process Synchronization :-
↳ why required?

Process communication → more than 1 process trying to communicate with each other by some source.
↳ eg.

- ① shared resource
(Projector)

↳ main ukhon aapko dikhega,
aap ukho mujhe dikhege,
but maine kuch likha, beech mein aap mein kuch,
fir matne.. so on.

then for other may be confusing!
so synchronization seq.

P1

{

- -
Count ++;
= }

P2

{

- -
Count --;
= }

Count → shared variable for process P1 & P2.

Let initially, count = 5.

Count ++ → not single instruction.

Preemptive → any process can stop any other at any time

- ↓
① mov count R0 (move value of count to register R0)
② inc R0
③ mov R0, count

→ 3 instructions we actually execute

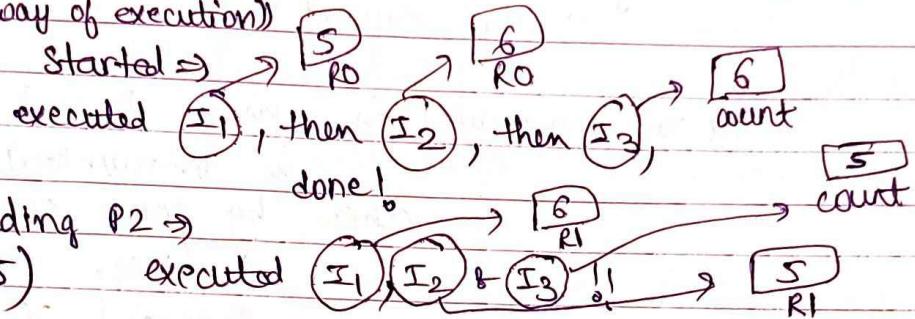
WDMK
30.

11/11 for count--;

- (1) mov count RI
- (2) dcr RI
- (3) mov RI, count

Case 1 (1 possible way of execution)

Process P1 started



then, loading P2 →
(finally, count=5)

executed (I1), (I2), (I3) !!

done!

Case 2

P1 : (I1), (I2) executed

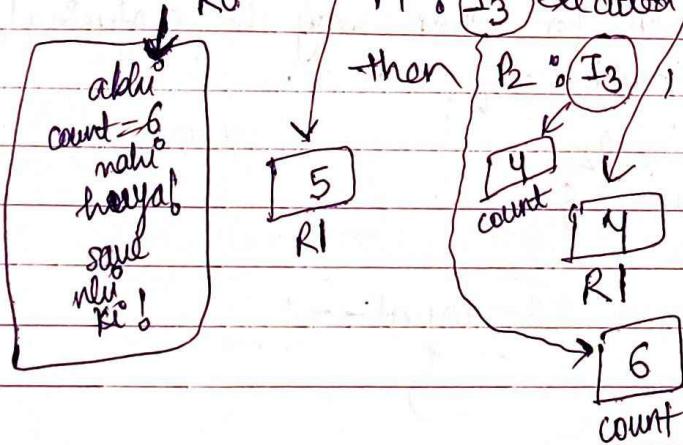
Preemption possible at any time

Now P1 pre-empted
and P2 loaded,

P2 : (I1), (I2) executed,
then P2 got pre-empted,

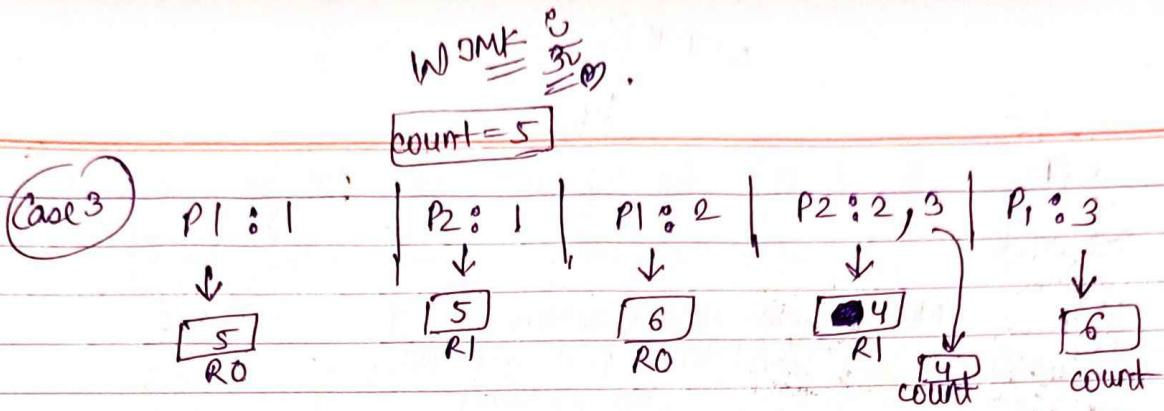
P1 : (I3) executed, done,

then P2 : (I3), done !!.



So, finally, count = 4.

so, different final value, depending upon the order of the execution.



finally, count = 6.

so, there is inconsistency in output, depending on order of execution.

problem \Rightarrow race condition

depending on order of finishing \rightarrow output matters,

bcz we've shared resource (output), which is used by many processes, then much have process synchronization, o/w can never predict the output (agar preemptive

hui scheduling).

(as pta nahi kab kisne, kisko pre-empt kar diya, as synchronization nahi pta).

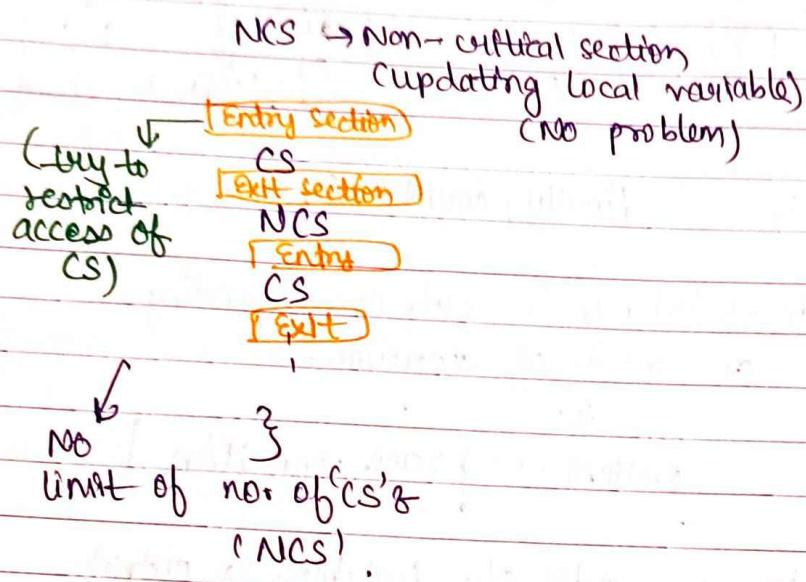
① Critical Section:-

that part of program, where updating the shared resource.

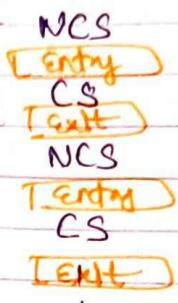
e.g. $\text{count}++;$ \rightarrow in P1 } $\text{count}--;$ \rightarrow in P2 } are critical section

- ② what will be output depending on the order of execution?
race cond & The order of execution of instructions defines the final output.

P1



P2



problem may occur \rightarrow we have no control over access & execution of CS,

ϕ
needed some mechanism!!

\hookrightarrow to h. such lines of code before entering in CS \rightarrow Entry section

& such code lines after exiting from CS \rightarrow Exit section.

prob'l'm \rightarrow many processes changing shared resource simultaneously!

① Requirements of Synchronization solution! -

① Mutual Exclusion :-

WOMEN

(CS → dressing room in shopping mall

↳ ok person ~~is~~, ek time jaa skta

↓ ↳ mutual exclusion

make liye some mechanism chahiye

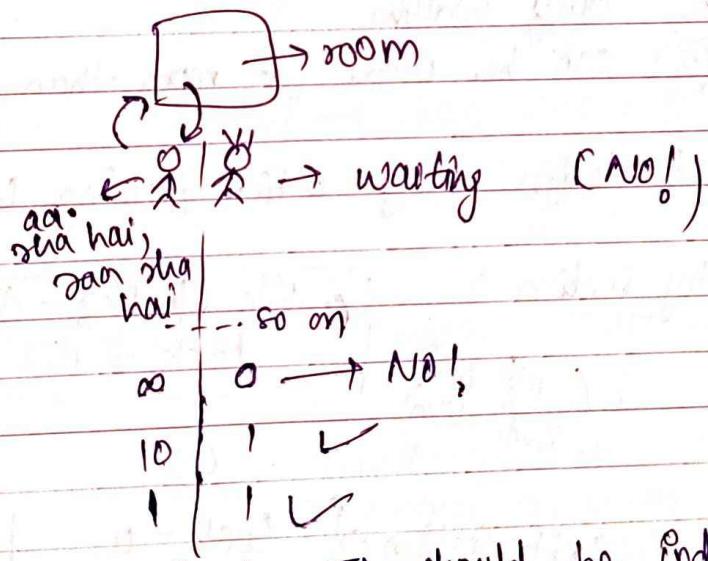
② Progress :- if process not interested to go "inside room", toh dusre "ko, to " " " " " ke naa sake!

① & ② → are primary requirements

↳ These two must be there!

③ Bounded waiting :-

↳ There must be some bound, taanki pta ho, dusre ko kab chance mile to enter in CS.



④ Architectural neutrality :- It should be independent of hardware

③ & ④ → Secondary requirement

W.M.K 30.

① Two types of synchronization Methods :-

① with busy waiting

② w/o " "

(1 andar room ke, dusra
continuously knocking !!)

(1 andar room ke, dusse
+ dekha somehow, pta
chla koi hap
andar! → wait kiya,
so gyi, sab dusra
baahar aayie, uski
responsibility huu
wake up karna & andar
thejabhi)

Use of "lock variable" :-

features of this variable :-

- ① software mechanism
- ② Busy waiting
- ③ can be used for more than 2 processes

want to design entry section , using lock variable!

entry section ⇒

(not single instruction)

while ($lock \neq 0$)

lock = 1

CS

→ not locked ! po nho
andar,

↑ enter,
abhi lock
kediya !

exit section ⇒

lock = 0

WORME

Enter kiya, lock kene se pehle (status kiya), koi aur enter kar gya (status), koi aur ... so on!

Better way??

↪ But entering room, I'll have tag [FULL],
~~check~~ ~~empty~~ ~~andar gaadunga~~ wo gaadunga doore par, full
check ~~empty~~ ~~andar gaadunga~~ empty or not, and if empty, will entry room!
on [I am in]

→

1. Load

lock RO

has previous value of lock

2. store

#1

lock

→ (impression other to #1 full)

3. cmp

RO

→ (to check empty hai)

pehle se yaa locked

↪ as hmne ek apna

lock pehle

taa diya,

dekha hai nhi empty or not!

Jump to step 1

← 4. JNZ step 1

CS

5. Store #0, lock

↪ preempt after I → problem

→ I2, I3 → no problem now!

P1: | P2: | . - . | Pn: |

↪ problem still there!

$\text{WOMK} \equiv \frac{1}{2} \text{v}$. $\text{WOMK} \equiv \frac{1}{2} \text{v}$.

(1), (2) \rightarrow two independent instruction nahi, single instruction \rightarrow atomic instruction hoga!
 only then problem will get solved!

hardware
men hoga
essa only
kuch, only
then applicable)

(require support
from hardware)

(2) means both
can happen w/ll
happen simultaneously

TSL (Test Set Lock)

problem solved !!

1. [had lock. R0] \rightarrow dekha empty, lock lgaya,
store #1 lock
no one can, 1st

2. cmp R0, #0

3. JNZ step 1

CS

b/w take photo,
and take ki haan,
empty!

4. store #0, lock

① Mutual exclusion ✓

② Progress ✓

as if not interested ~~to~~ to
go inside \rightarrow (1) w/ll never

③ Bounded Waiting X

as other process may wait for
so' time!

primary
requirements
fulfilled

① but may lead to Priority Inversion !!

Problem called
spin lock!

P1 : 1, 2 | P2 : 1, 2, 3

then another
process P2 came \rightarrow

priority ↑

\Rightarrow CPU will be given to P1.

P2 \rightarrow also interested in executing \rightarrow CS !

P1 new lock = 0 | jaldiya!

(2) men cmp R0, #0
(2) not toh P2
will jump to 1.

esse entry section

men as ly resolve karta shega

~~W₁ W₂ W₃~~ = ~~30~~ 00.

although P₂ k_i^o priority ↑ , still can't start
execution ⇒
problem → Priority Inversion.

WJMK $\frac{3}{5}$
= 3

17 Aug, 2023
Thursday

कालतं वार्षिक स्नान मरणो, निमुक्ते कालतं अस्फृत॥३॥

WJMK $\frac{3}{5}$
= 3

Lec-6:-

TSL \rightarrow satisfying all other 3,
but ~~not~~ not Architectural neutrality,
so want mechanism independent of architecture.

① Strict alternation approach or turn variable :-

(alternation happens b/w 2 processes
after certain interval)

\rightarrow s/w mechanism,

??

\rightarrow Busy waiting

\rightarrow designed for 2 process (sometimes 2 processes ke liye bhi work nahi ~~hota~~ hata).

2 processes $\Rightarrow P_0, P_1$

P_0

NCS

{entry section} \rightarrow while ($turn = 0$);

CS

Exit section $\rightarrow turn = 1$

false when
 $turn = 0$
while
loop
break

P_1

NCS

while ($turn \neq 1$)

CS

$turn = 0$

~~turn~~

$\rightarrow turn = 0$

keep on
repeating,
as no
body in
while,

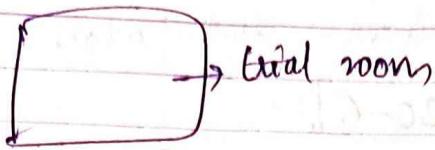
Gya
naya
har;

Synchronization process we propose, have

2 sections :- ① Entry

② Exit

WCMK = $\frac{C}{M}$.



Initially key
iske paas

→ andar jaayega Ichal ke, fir ② ko doga,
ao jaayegi andar... esse alternation
chali rhegi!

Iske paas key → can go andar.

turn = 0 ; → means its turn of P₀ to enter CS.

turn = 1 ; → ————— P₁ ————— and P₀
turn₀ = 0 → means its not P₀ turn ~~should be stopped~~ to
enter CS.

turn = 1 → P₀ will stuck, P₁ will go!

turn → variable → whose value indicates, kiske
turn to go inside,
we have to initialize with value 0 or 1.

① both can't enter CS at same time, as turn ka '1'
value hi ho skte hai :-

② so mutual exclusion guaranteed

(Exchanging key and who-ever has
key, only he/she can enter)

③ Progress:- only jo interested hai andar jaane mein,
only those should take part in deciding
whether jaan jaayega andar, kab jaayega?
→ Not interested wale stop nahi kar skte!

W_{MAX} = 30

initially turn = 0

P₀ → no Interest

(P₁) → interested to execute CS
stuck in while loop

execute nahi krega CS, and jab tak CS execute
nhi krega, won't execute exit section
turn = 1 hogi hi nhi kabhi!

→ so key hal, andar nahi jaa shor & dusse ko
key nahi de skta jab tak andar jaa pr
baahar na aaye & interested wala andar
nahi jaa paayega! 😢

so, progress not guaranteed.

③ Bounded wait is guaranteed.

P₀ → gya,
baahar aaya,
key di P₁ / P₀

P₁ → card go 😊

P₀
1.

P₁
1.

∞ | 0 → not possible!

(as alternatively, chance de zara ham)

④ as slow soln, so, Architectural neutrality 😊

But ② 😊 not guaranteed!

WJMK = 30
= 1.

Prob: key dena sei pehle check nahi, jisko key di,
 wo room ke andar jaane mein interested
 hai bhi yaa nahi?
 sof tsbo key deni chalihe thi, jo andar jaane
 mein interested ho!

① Interested [0] = T (True); → P₀ interested to go inside
 CS

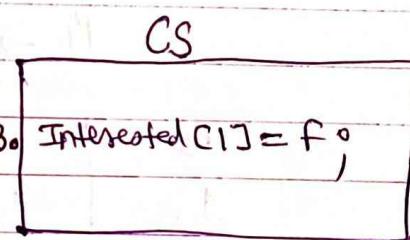
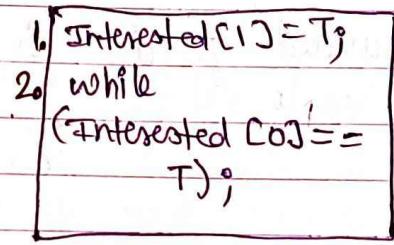
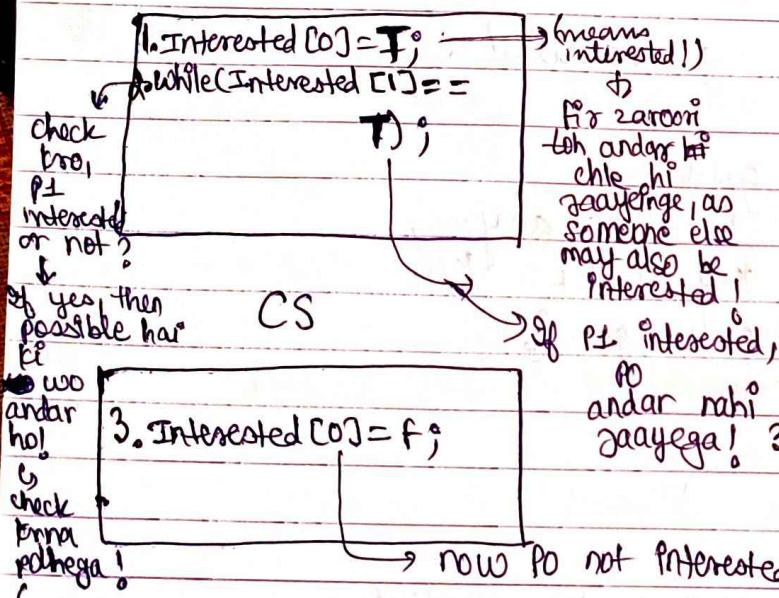
Interested [1] = T (True); → P₁ interested to go inside
 CS

for P₀

Non CS

for P₁

NCS



again process interested, it should tell, I am
 interested!!

if P₀ interested, dekhega kya P₁ interested? → if yes!
 may be inside! → wait kegar P₁ ke liye &

P₁ interested, dekhega kya P₀ interested? → if yes!
 may be inside! → wait kegar P₀ ke liye

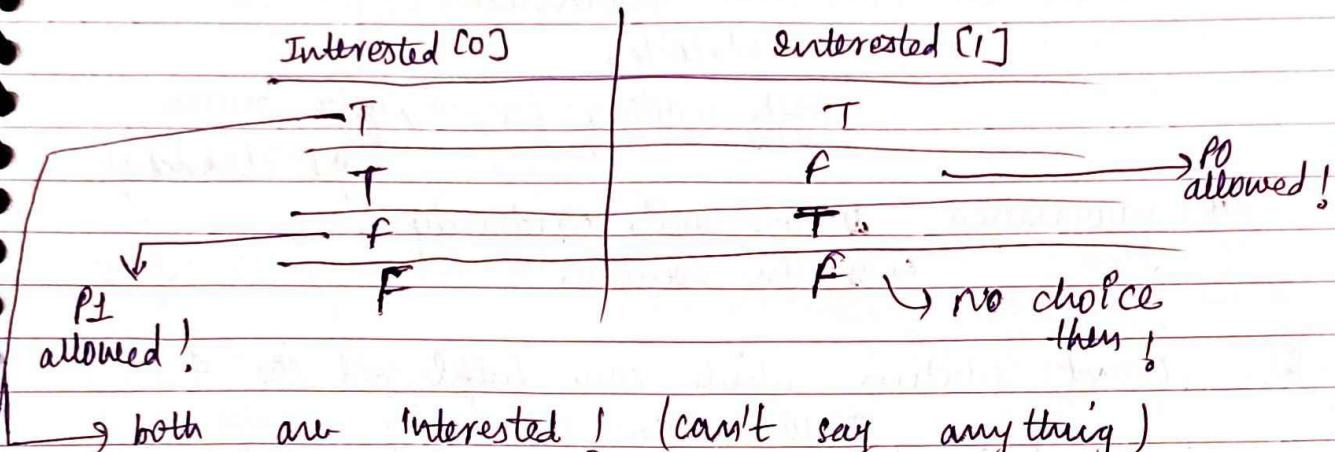
can't communicate with each other!

↑ fir tak toh jo is andar jaayega na hua nahi?

WJM 35 Q.

① mutual exclusion guaranteed!

& scenarios:-



both are interested! (can't say anything)

{ either P₀ or P₁ or none! } (who will be allowed or not)

P₀: 1, 2, CS | P₁: 1, 2 | P₀: 3 | P₁: 2, CS

Interested [0] = T (will get stuck in while loop)

FINE (done execution
for p1 after complete of CS)

P₀: 1 | P₁: 1, 2 | P₀: 2

(get stuck in while loop)

(both get stuck in while loop)

(none of them can break loop,
as true hogi while loop ki condition).

may occur in this scenario.

② progress guaranteed

~~guaranteed!~~

WJMK 25/01

still not perfect solution as deadlock may occur!

③ Bounded wait not guaranteed
(as deadlock)

④ guaranteed Architectural Neutrality
↳ as s/w solution.

① want solution which can fulfil all the 4 requirements.

Peterson's Solution:- First scientist 1 & 2 cond's satisfies, but

↳ merge kiya last 2 studied approaches ko,
to get result.

```
#define N 2
#define TRUE 1
#define FALSE 0
```

```
int Interested[N] = f
```

```
int turn
```

```
void entry-section (int process)
```

```
{
```

```
int other;
```

```
Other = 1 - process;
```

```
Interested[process] = True;
```

```
turn = process;
```

```
while (Interested[other] == True && Turn == process);
```

```
}
```

→ initialised to false, none of them is interested

WJMk 5/20.

void exit_section (int process)

{

6. Interested [process] = False; → Now I am not interested to go inside

}

① N → no. of processes

function

void entry_section (int process)

{

1. int other;

2. other = 1 - process;

3. Interested [process] = True;

4. Turn = process;

5. while (Interested [other]) {

}

I am interested
to go
inside)

and saying
that it's my turn to go inside!

2 processes

P0 P1

pt will

P0 interest, toh ^{pt} pass value '0'
to this fn,
if P1 interest, then (±).

process = 0
other = 1

process = 1
other = 0

agn ek '0',
toh dusra (±)
hoga !!.

checking kya dusra

interested? → if true!, I should
wait! (esse nhi 2nd baachan

hi rhein, → so checking! →

meri turn hi hai?

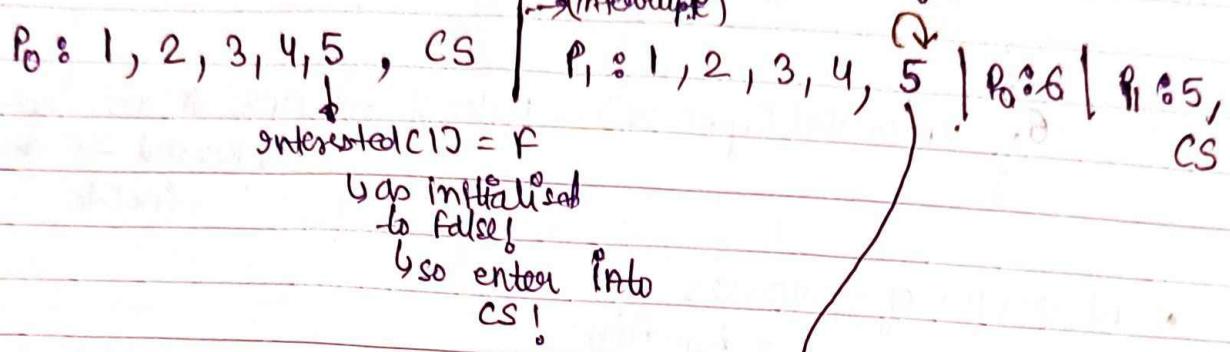
turn == process;)

(es turn containing my
process Id → means my turn to

go inside!)

WJMF
3v

both processes are executing parallelly!



interested[0] = True ✓
& turn = 0 ✓

↳ so condn of
while loop → true!
get stuck in while loop!

why happened ??

If both interested \Rightarrow

① I am ahead (other is just behind me)

② Other is ahead

↳ turn = 1 (hai agar abhi)

↳ p1 late!

↳ P0 pehle aaya & interested \rightarrow so may
be inside CS \rightarrow so I should wait!!

e.g.

blackboard
par roll no.

Utkha, 200

class mein aaye!

(overwrote)

↳ pelle aaya \rightarrow 2001 Utkha

for aaya \rightarrow Overwrite kiker 2002 Utkha!

↳ means yeh late
aaya!! \rightarrow so current value of
roll no., wo late aaya!

WDMK

$P_0 : 1, 2, 3$ | $P_1 : 1, 2, 3$ | $P_0 : 4, \overset{Q}{5}$ | $P_1 : 4, \overset{Q}{5}$ | $P_0 : 5, CS,$

↳ dead lock or not?

↳ No!


$twin = 0 ?$
false!

while loop
breaks

$6 | P_1 : 5, CS, 6$
~~turn = 0;~~

$P_0 : 1, 2, 3$ | $P_1 : 1, 2, 3, 4, \overset{Q}{5}$ | $P_0 : 4, \overset{Q}{5}$ | $P_1 : 5, CS$

↳ P_0 pehle aaya, itaayen interested!

P_1 late aaya,
par pehle twin = 1

Krdiya
and
wo
pehle

gya in CS!

P_0 , 1st process ko access nahi lega CS ka;
jo pehle twin ki value ko set



This solution satisfies all 4 conditions!

s/w soln \rightarrow so architectural neutrality is there!!

WJMK
= 30/31

23 Aug 2020
wednesday

कालीन वर्ष का संचयित रक्कर मासिक,
निरमुक्ततेर कालीतमा अगस्त्य ॥ ३०॥

lec-7

with busy waiting \rightarrow all processes will continuously check ki
can enter in CS or not.
w/o " " \rightarrow they can sleep if abhi can't enter CS!

Producers - Consumers prob:-

produce some items
↓
can produce only if items
present in buffer!

↓
consumes produced items
↓
can consume only if items
present in buffer!

↓
not exceed buffer memory!

Produce producing only buffer mein
sagaah hai
ekhne ke

produce k
ke saka hai?
↓
consumer ke saka hai?
↓
kuch count ekhne na
is ke liye !!

Consumer consuming
↓
can consume only if
kuch buffer mein
hoga

Code :

WJMk
= 30
= 100.

```
#define N 100 // size of the buffer
#define count 0 // items produced or present in buffer
void producer(void)
{
```

```
    int item;
    while (true)
```

```
{
```

```
    item = produce_item();
```

```
    if (count == N) → Buffer is full, so
        sleep(); → producer can't produce
                    any item
```

```
    insert_item(item);
```

```
    count++;
    if (count == 1)
```

```
wakeup (consumer);
```

```
}
```

```
}
```

```
void consumer(void)
```

```
{
```

```
    int item;
```

```
    while (true);
```

```
    if (count == 0)
```

```
        sleep();
```

```
    item = remove_item();
```

```
    count--;
    if (count == N-1)
```

```
wake_up (producer);
```

```
consume_item(item);
```

```
}
```

```
}
```

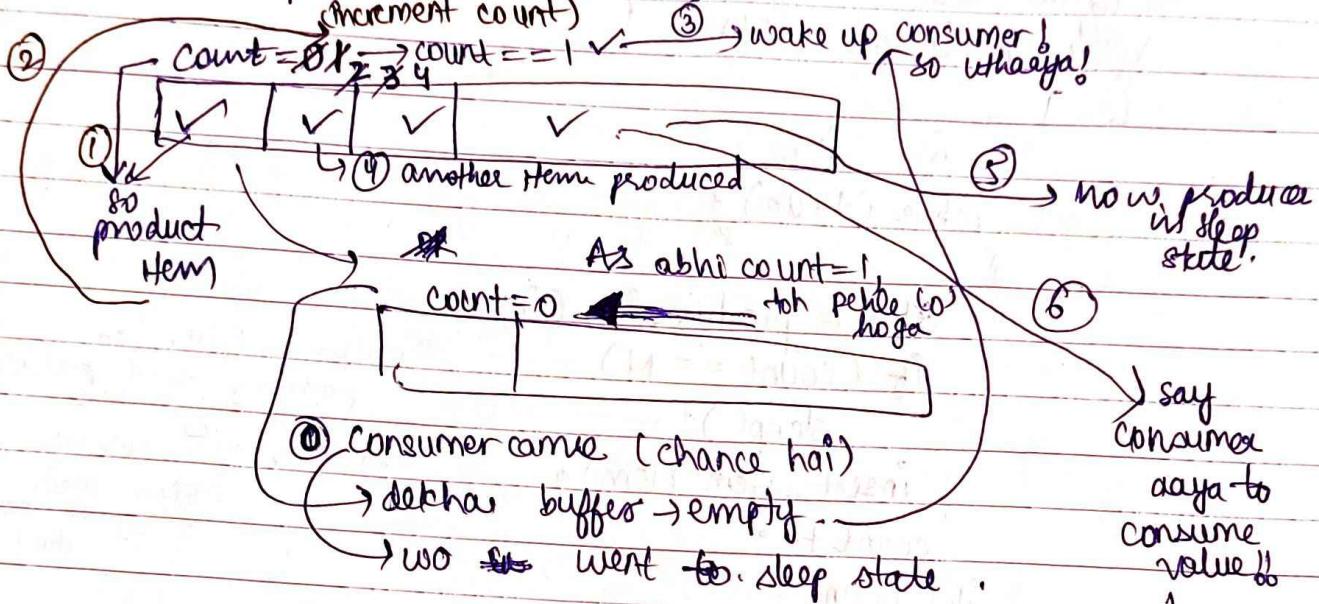
producer will go to sleep state, if found that buffer is full & empty.
consumer,

↑ so tha hai

isti responsibility to wake consumer if some item is produced.
(Vice-versa)

WJMKE $\frac{3}{5}$

If consumer consumed one item \rightarrow consumer's responsibility to wake up the producer to produce some item.



$$\text{count} == N-1 \checkmark$$

means pehle count = N

Buffer was full,

so there is chance that producer

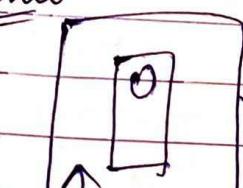
ayga hoga pehle to produce

something \rightarrow dekha full \rightarrow so gya hoga \rightarrow

ab usse uthana hai! (Consumer

ki responsibility)

bhelle



zoom!

1 person and

other " continuously

knocking (with
babbling)

Now

wf! busy waiting,

Don't wait,

Sleep! 😊

WJM
TUE

Dab wo baahar aayen, it's their responsibility dekhin,
ko baahar wali kisi -2 so toh nahi' gya, agar
haan, usko utthaayo!

Let's say \rightarrow if ($\text{Count} == N$) ke baad interrupt
aayga!

producer ne dekha count = N ,
toh sleep mein jaana chahiye,
lekin " state mein jaane se
pehle, pre-empt ho gya, toh
process still in ready queue, not in
blocked queue.

process in sleep state is in blocked queue.

Now,

As Count = N !

\hookrightarrow consumer ne 1 item consume ki!

\hookrightarrow count = $N-1$.

consumer ne socha may be
some producer sleeping!
 \hookrightarrow padega no producer sleeping!

bisi ko nhi utthaayega! then consumer
so wake-up (producer) signal will
get lost!

then,

producer waala process chla & so gya!

\hookrightarrow then consumer ne 1 item consume
ki!

\hookrightarrow abhi count = $N-2 \neq N-1$

\rightarrow toh consumer ko kisi maine pehle
signal bhi diya tha, agar ithna huya tab
utth gya hogya, sab ki producer wake
up call ka heart bata

WOME ~~32x32~~

rhega!
probm P8)

That wake-up signal() is lost!

↳ producer sota rhega!

↳ put some tag on door, jab utthaane jaayegi!

taante jab soe aaye bnda, wo tag
phle aur ~~sayega~~ na!

In this way, wake-up signal();
waste nahi hoga!

As pre-emption can come at any time, so wake-up
signal() may get waste!!

↳ Semaphore (Is that tag jo btaayega,
whether process should go to sleep state or
not,))

↳ it is a variable

(So jo bhi probm shared variable par
ho slai, wo isse par bhi hogi)

(pre-emption kab huyi, depending on that, iski
value will be inconsistent, as simultaneously
many processes can change its value)

↳ So need to add some features in it,

↳ so it is a variable on which read, modify
and update happens atomically!! in kernel
mode.

↳ (read/modify/update)

[either process completely
hoga yaar hogaa wii nahi]

WOME
30/30

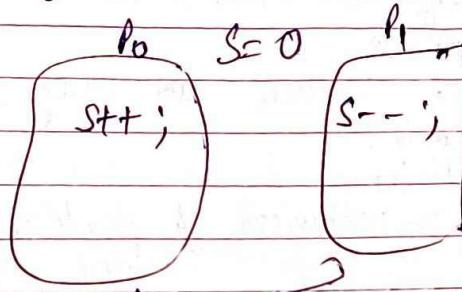
So, Semaphore is not a normal variable.

Synchronization & race cond' → problem
User mode user kerte hoga

Semaphores are of two types:-

- (1) Counting
- (2) Binary

semaphore++;
→ yaa toh pura
hoga, yaa hogea
ki mali!



race cond' won't
happen if atomic op's
in kernel mode!!

Counting Semaphore :-

struct semaphore {

int value; } ; (Indicates how many

resources available or
processes waiting
for particular
resource)

Up (Semaphore s)

{

s.value = s.value + 1;
if (s.value <= 0)

Down (Semaphore s) → jab bhi access mile resource ka
pebble. update pebbles

{ s.value = s.value - 1; (decrement value of
resource.)

select a process
from L;

wake-up();

{

{

means
pebble
available atleast
(or) busy state
has

and s.value = 0
or y means no
particular resource " " is available!

WORKS
= 30
= 10

① If a resource is having multiple instances \rightarrow then go for counting semaphore.

↳ used to count, how many processes are waiting for particular resource, or how many resources are available.

② binary

↳ used for mutual exclusion
↳ only 1 allowed inside at a time process

↳ when resource has only 1 instance.

③ In code:-

Let have

2 instances of particular resource -
value = 2



There are 2 fns to modify value of semaphore variable:-

- ① Down (to decrement value of semaphore variable)
- ② up (to increment value of semaphore value)

P₀ was requesting for these 2 resources

$$, \text{value} = 2 - 1 = 1 \text{ (decrement kyaar pehle).}$$

∴ value > 0,

↳ so no problem, return!

P₁ & P₂ aaya \Rightarrow value = 1 - 1 = 0 (0 not)

↳ so no problem, return!

As P₂ aaya & value = 0 - 1 = -1 < 0 (ohh ??!!)

WJMC

So, P_2 kya deega?

Pr P_3 aaya
↳ $P_2 \rightarrow$ Queue (L) mein place ho jaayega!

$$\Rightarrow \text{value} = -1 - 1 = (-2) < 0.$$

↳ oh!!

$L \leftarrow P_3$ (P_3 bhi L mein aaj jaayega)

$$\text{so, } L \leftarrow P_2 \barwedge P_3$$

Hanf essa h
mai jao!!

Indicates, 2 processes waiting for resource

of this type

Up operation:-

once resource used & u want to release it! \rightarrow so & occupy ek instance of resource \rightarrow abhi release krdge \rightarrow means available ho jaayega dusrau ke liye!

$$\text{value} = -2 + 1 = (-1) \leq 0 \text{ (Yes!)}$$

Indicates that there is a set of processes waiting for resource occupied by P_0 .
abhi iski responsibility, unko utheana!!

P_2 will be selected & will be assigned the resource!!

P_1 P_2 P_3 (P_2)

Now $\frac{1}{2}$ is waiting for resource!!
process

value decrement nahi
deega!

↳ simply will enter in CS!

sodha sleep
start se
uthega!

WORK

$$\text{for } S \text{ value} = -1 + 1 = 0 \geq 0$$

↓

so P_3 ko utthayega P_2 .

Now process is waiting for

P_4 aaya \rightarrow will no perform down operation \Rightarrow

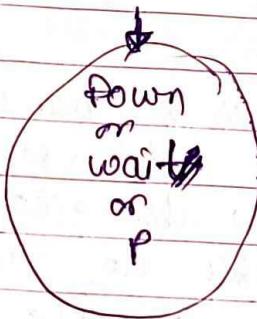
$$\text{value} = 0 - 1 = -1,$$

will go in L.

Incrementing



Decrementing



① If ~~value~~ only 1 instance available.

BINARY SEMAPHORE :-

struct B_semaphore {

enum value (0, 1) → value of semaphore can be 0 or 1 (as binary semaphore)

Queue type L;

value = 1

also known as mutex

as mutual exclusion ke tarekya janta hai

not available
available

L consists all processes

PCB which got blocked

while performing Down operation

WJMK = ~~3~~ 2

Down (B-semaphore S)

{

if (S_{\circ} . value == 1) → available han' resource, toh
{ assign ready processes.

{
}

S_{\circ} . value == 0; → and unavailable ready!

(means already assigned to some one),

~~else~~

{ put the process PCB in L ;

sleep();

↳ resource not available, put process in queue

{
}
}

↳ so jaago!

(Up) (B-semaphore S) process

{

if (S_{\circ} . L is empty)

↳ if queue is empty

{

S_{\circ} . value = 1;

(means no process waiting for that

}{

else

{

select a process from S_{\circ} . L;

↳ ~~it~~ value = 1).

}{

wake-up();

}{

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

<

WJMK

ans:-

mutex a, b;
a = 1, b = 0;

P₀

while (true)
{

Down(a);
Print("1");
Up(b);
}

while (true)
S

Down(b);
Print("0");
Up(a);

agar here sleep ke gya yeh print nahi hoga! jab tak koi water na - up kee!!

what will be the OUTPUT ??

()
0
1
0
!

irrespective of order of execution!!
(alternation se it is printing).

WOMEN
लड़कियां

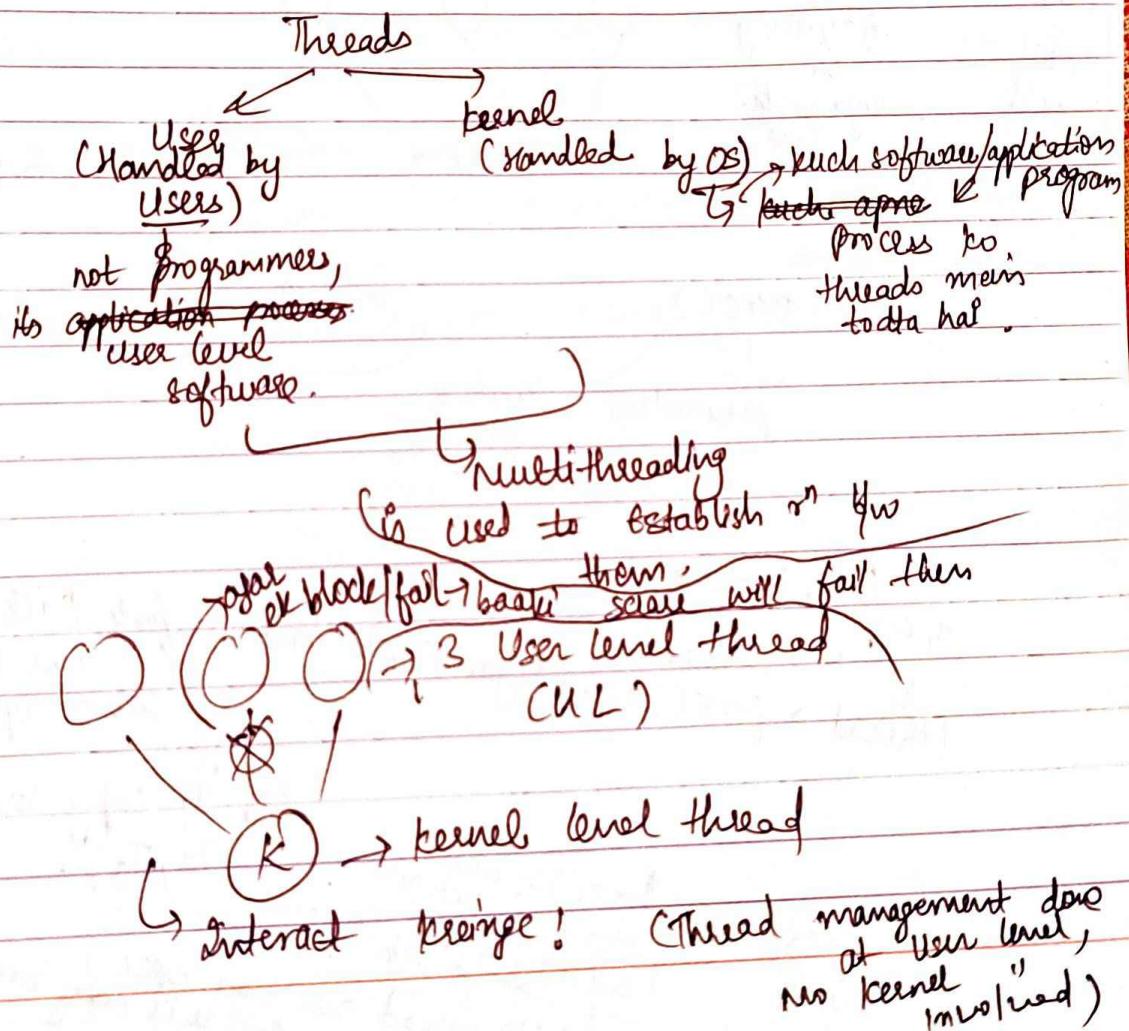
WOMEN
लड़कियां

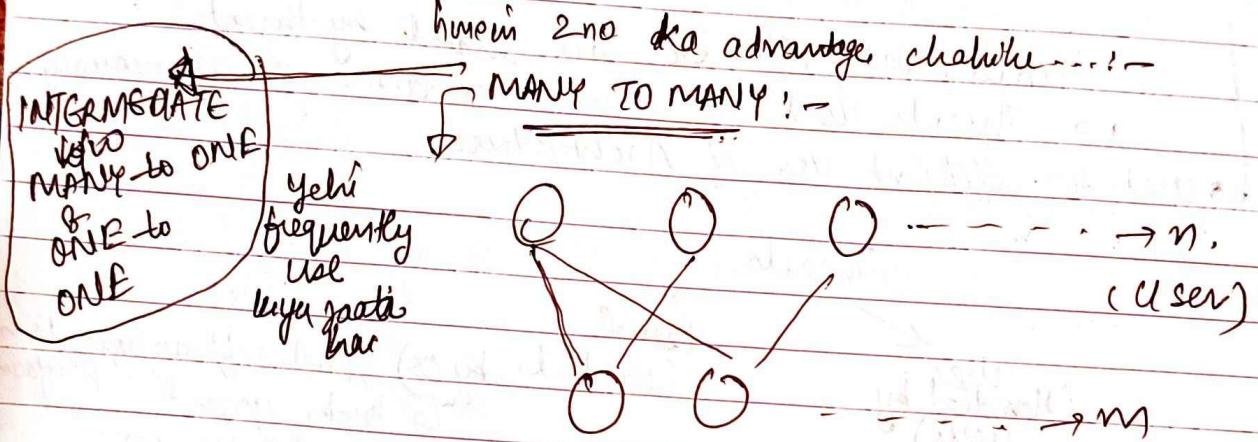
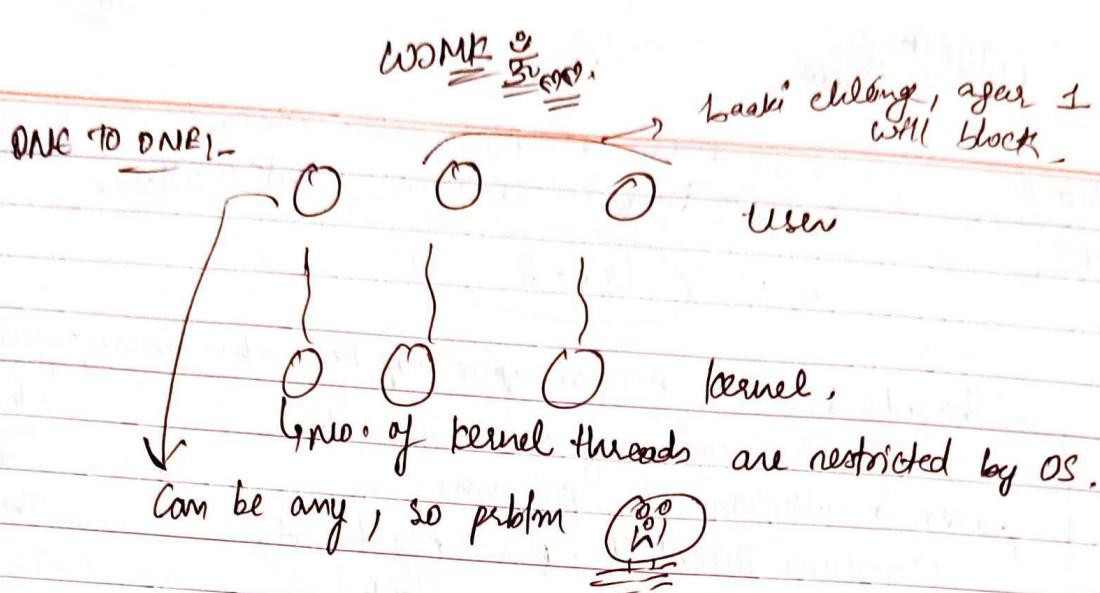
25 Aug/2023
FRIDAY.

कालतीकू कार्त्तिक द्वन्द्व मरणों,
निरमुक्तैर कालेक्षण अभिष्ठान॥५॥

Lec-8

- Threads :- Ek process apne aap ke sub-process mein totta
that → wo chhoti turde +
process → collection of programs.
→ contains Thread ID, PC, Register set, Stack.
↳ code, Data, files etc. are shared by threads.
→ Threads used to share a resource simultaneously.
↳ used for efficient use of Architecture.





exec()

non parametric

replace process?

MCN

1 process

T_1, T_2, T_3

3 threads.

1 thread \rightarrow fork() \rightarrow call kya \rightarrow its time se fork call huya, usse time se same copy bn jaata hai!!

if T_1 copy kena yaa T_2, T_3, \dots sare 32

as fork ne sab threads ki copy bnaao!

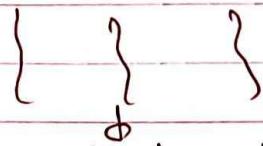
fork() \rightarrow pchle yeh call huya

exec() \rightarrow for yeh,

so thread call huya exec() sab koki copy bnaao!

WJMF C
5v enq

⑥ Asynchronous Cancellation:-



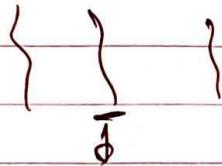
Thread nahi chahihe?

Yeh kaise htaayoge?

① siif isko

② yaa i-i like sbko ??

① Deferred cancellation:



apni mere cancellation se kisi ko
no fork, then waado,
o/w nahin !!

* { Matrix multiplication Using Threads
C GFG code & galvin Code) (Simp)

② ③ | w b| w User Threads & Kernel Threads.