1. Good Evening
2. Let's begin at ~~9:10~~ 9:15 pm
3. Intro to Operating System

DBMS

5 5 15

1. Homework → Joins, Aggregates, SQ
2. Extra DBMS
   - ↳ Deadlocks → Doubts
   - ↳ Full-text seach ↳ Schema
   - ↳ SP
   - ↳ UDF.
   - ↳ Trigger.

Operating Systems

1. Intro to OS → Theoretical

Process
Threading
Concurrency
Synchronization
Deadlocks
— Practical

6. Memory → Theoretical

$\boxed{\text{Operating System}}$

↳ How abblication work behind the

Scenes?

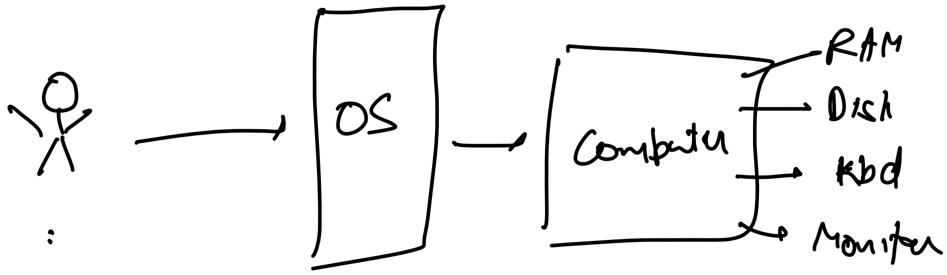
Agenda

1. Intro to OS
2. Multi programming vs Uni Programming.
3. Process
4. CPU Scheduling
5. Scheduling algos → FCFS , SRTF

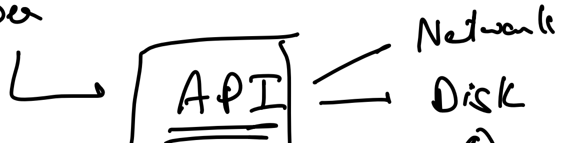
Intro to OS

→ What is an OS?
- ↳ User.
- ↳ Developer
- ↳ Applications

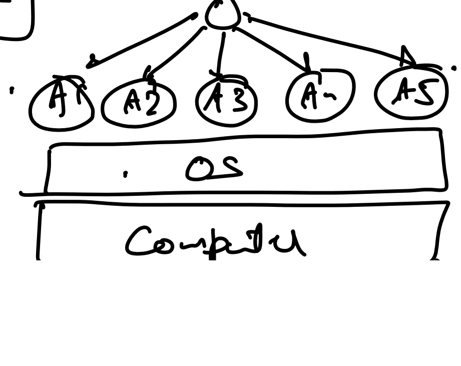
User



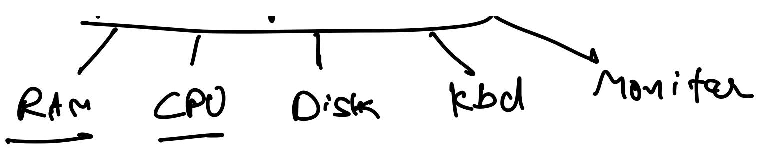
Developer



Applications

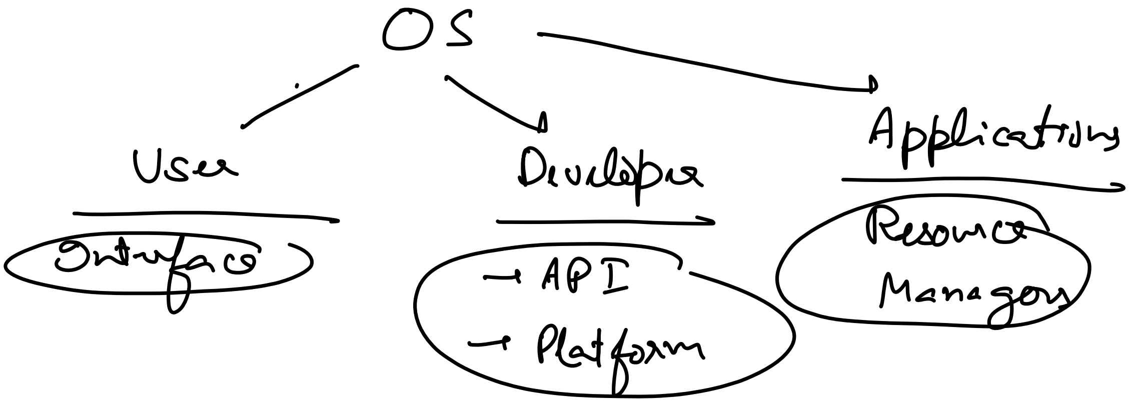


OS works like a resource manager?



High

- → Work gets efficiently done applications
- → conflicts don't arise b/w people for use of resources. users
- → Objectives of the management are met



Uniprogramming vs Multiprogramming

Multiprogramming → OS which allows you to run more than one program at 1 time
e.g. any PC

Uniprogramming → OS which allows a

Single program to run at 1 time.

- ATM Machine
- Calculator
- Lift
- POS

→ Toaster

→ Microwave

→ Wash Machine

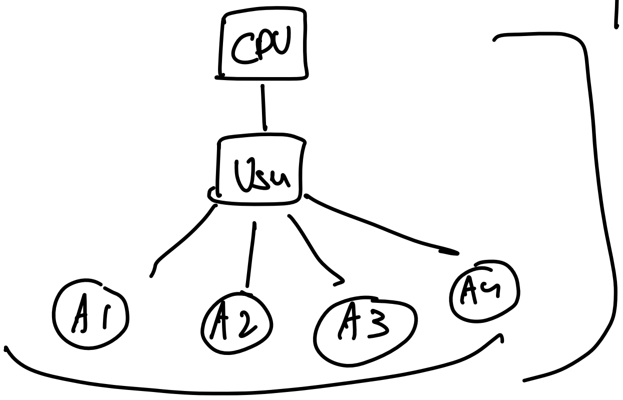
Multi-tasking
- Unix → Program
- Windows → Task

1 CPU
Single Core

Multi-programming

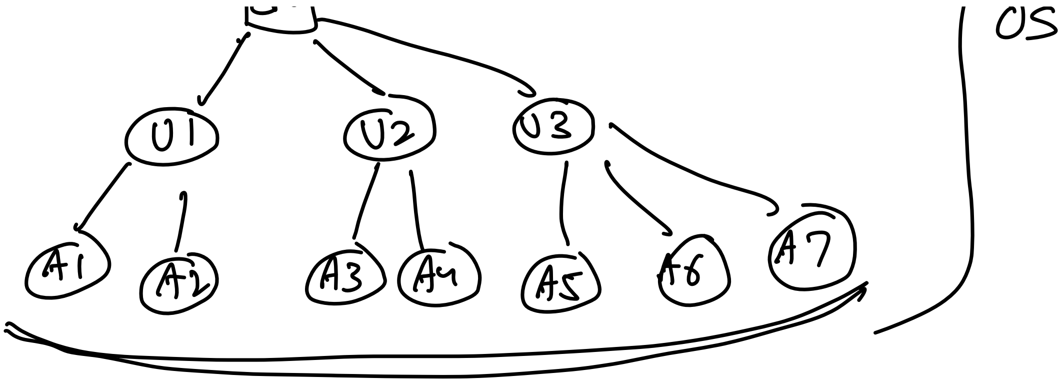
☆ # of users using the system at a given time

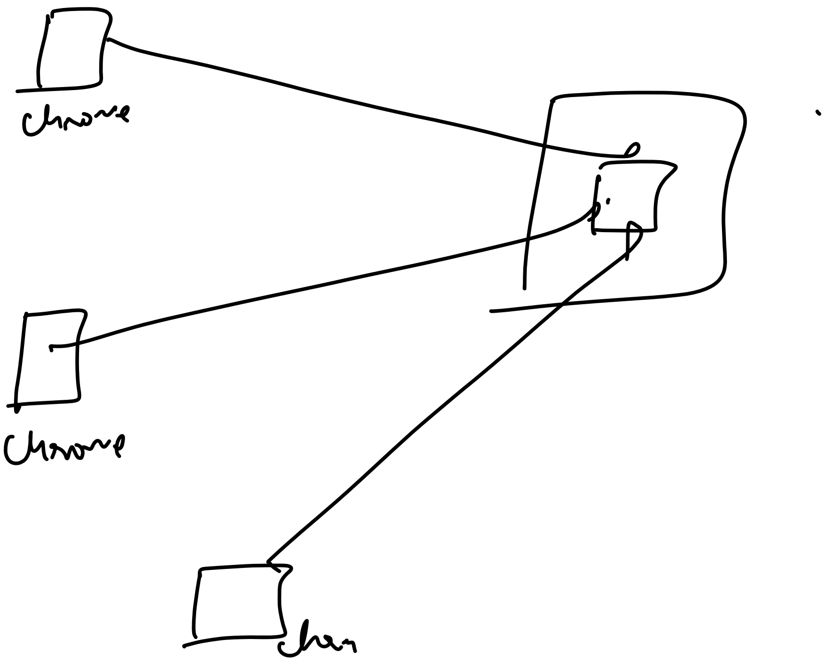
☆ Non-preemptive
Preemptive.



Single User
multi. programming
OS

Multi-user
multi programm



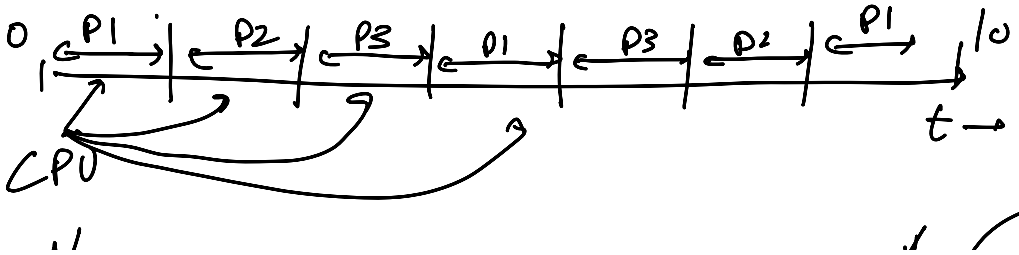


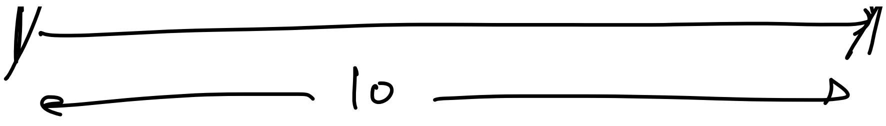
Preemptive vs Non-Preemptive

Preemptive → Preempt

Multi-

P1, P2, P3



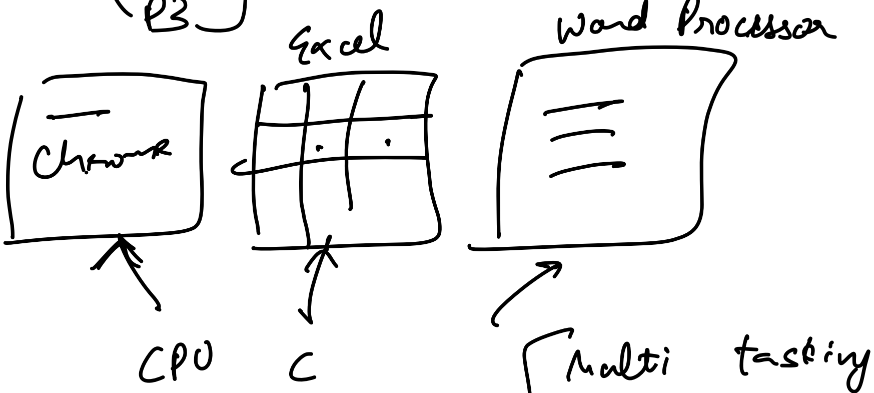


$\frac{1}{16}$ $\frac{1}{3\times10^9}$

3 x 10 $\quad$ 3 Ghz = $\boxed{3 \times 10^9 \text{ cycles/s}}$

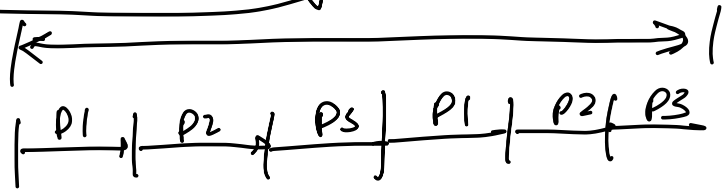
1 tn = 10^{-8} sec $\quad$ 16 photos in 1sec → Video

CPU = P1, P2, P3 $\quad$ Multi-programming



- Multi tasking
- Multi programming
- Multi processing

Parallel programming

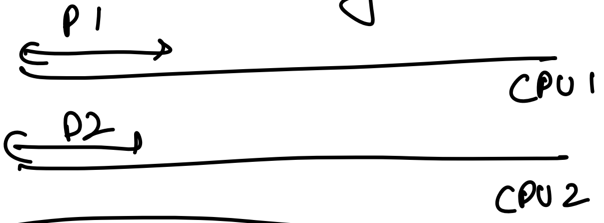


Pre-emptive multiprogramming

→ Switch CPU from one program to another before finishing it.

Parallel Programming vs [Multi programming, Multi tasking]

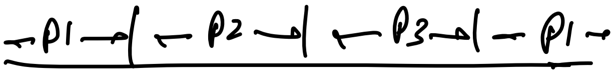
Real thing



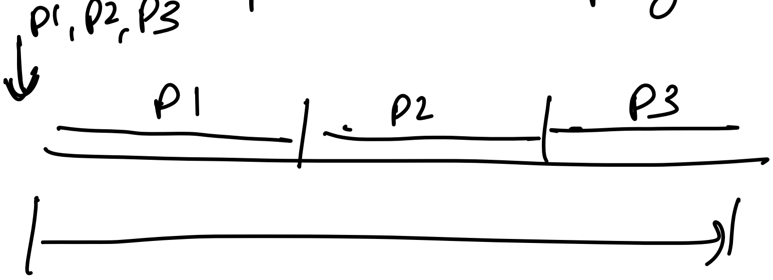
Parallel Programming

[Multi processing] Illusion

→ Running more than one programs at the same time

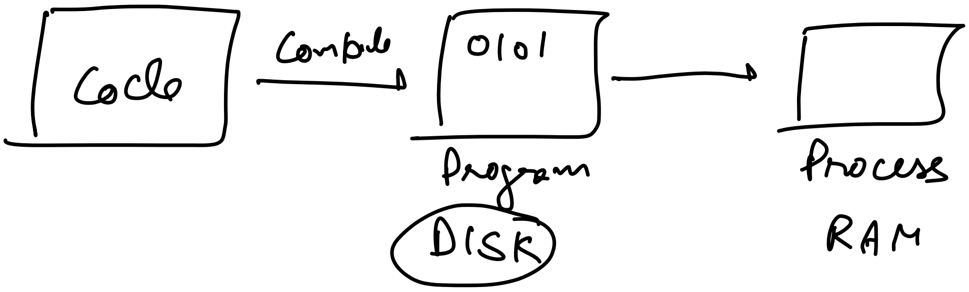


Non-emptive Multi-programming



PROCESS

1. Lifecycle of an application.



1. Code is written → Source Code

2. Code is compiled to form a set of instructions which can be run by CPU → program.

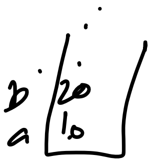
3. When program is run → it is brought to RAM from hard-disk & CPU starts executing it's instructions — process

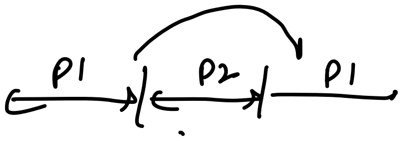
PCB (Process Control Block)

✓ a = 10 ✓
✓ b = 20 ✓ ✓
c = a + b → ☆
d = a - b

- → id
- → code section — set of instructions
- → program counter — next instruction to run
- → State of variables
- → priority (CPU Scheduling)
- → Memory Limits

PCB





How is an OS written?

Types of Processes

1. IO Bound Processes → Nw → Hard disk → Cmd Line → wait for user inputs

Word Processor

2. CPU Bound Processes → Translator

2. CPU bound Processes → Transcode
↳ Image Processing

3. Mix → Games [CPU]

IO Bound Process

1. Process is running
2. IO request is made via an interrupt (a signal to CPU)
3. CPU transfer contro to I/O unit for IO to be done

Waiting → Resource are not being efficiently?

Process another program

CPU Scheduling (5 minutes)

⟷ P1 ⟷ P2

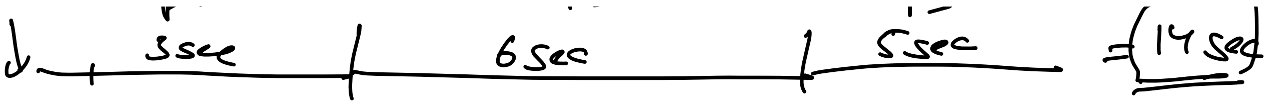
Pros

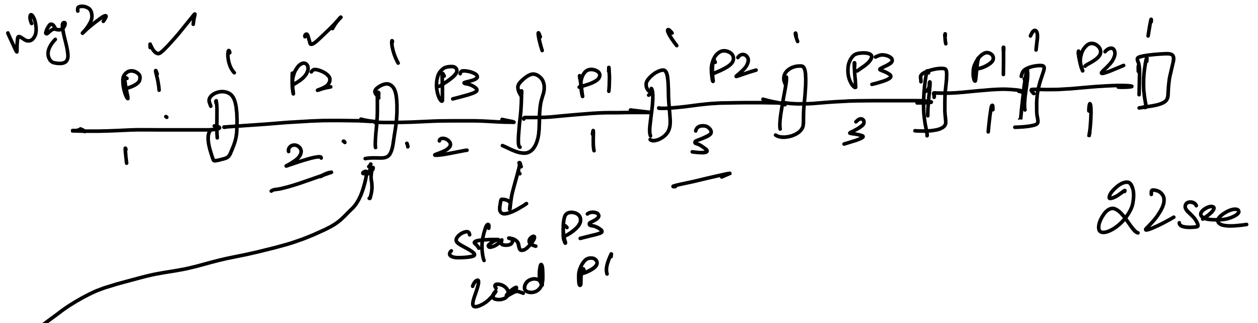
1. Efficiently use of CPU
2. Interactivity

Cons

P1 → 3 sec
P2 → 6 sec
P3 → 5 see

Way 1   P1   P2   P3





PCB of P1 = program counter, state of variables

PCB of P2 =

Context - switch time

CPU Scheduling Algorithms

Break → ~~10:49~~ 10:52 to 11:00

Win 3.1

1. FCFS (First Come First Serve)

- → Non preemptive algorithm]
- → Runs when a process finishes.
- → CPU picks the process with smallest arrival time

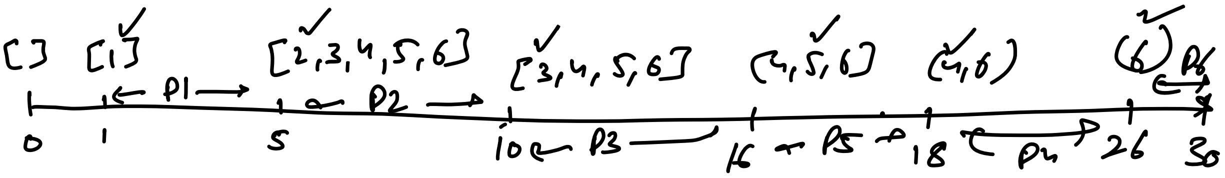
Context Switch Time = 0 [Assumption]

Pid	Arrival Time	Burst time
1	1 ✓	4
2	2 ✓	5

→ Time req d

3	3 ✓	6
4,	5 ✓	8.
5	4 .	2
6.	.5 ✓	4

1. Predicted
2. Analysis



Asgn → function

└ I/P List <Process>

└ O/p List <Output>

Process

└ id

└ arrival Time

└ burst Time

Output

└ id ✓

└ start ✓

└ end ✓

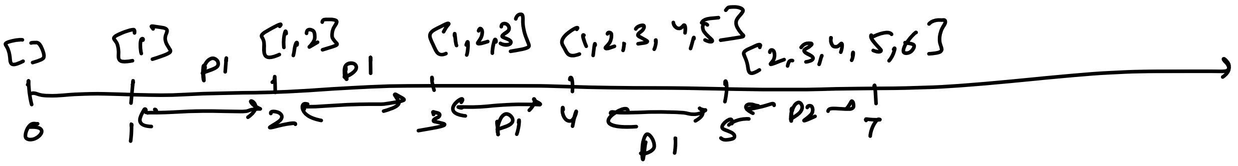
Scheduling Algorithm 2

└ SRTF (Shortest remaining time first)

pid	arrival Time	burst Time	rt
1	1 ✓	4	~~3~~ ~~2~~ ~~1~~ 0
2	2 ✓	5	5 .
3	3 ✓	6	6 :

s	>	-	-	
4 .	4 ✓	8	8 . .	
5 '	4 ✓	2	2	0
6	5 ✓	4	4	

1. Preempive
2. Runs → finish, a new process comes
3. The process with shortest remaining time will be picked



Major con of SRTF

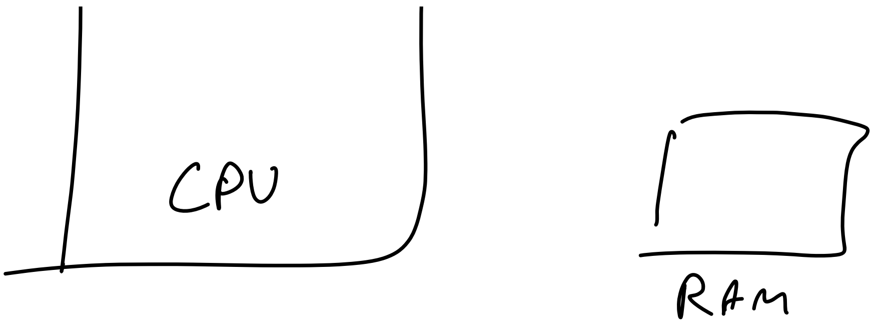
↳ Starving

Theoretical

Asgn → 1. SJF : Non prempive version of SRTF

2. Priority Scheduling

Registers

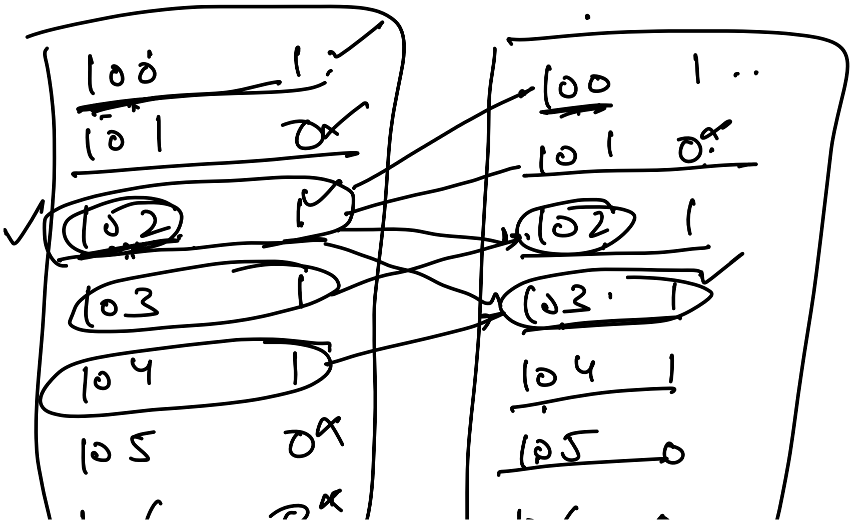


OBS

```
Select   MIN (ABS(p1.x - p2.x))
FROM     Point  p1
JOIN     Point  p2
    ON   p1.x != p2.x
```

x	
1	1
5	5
9	9

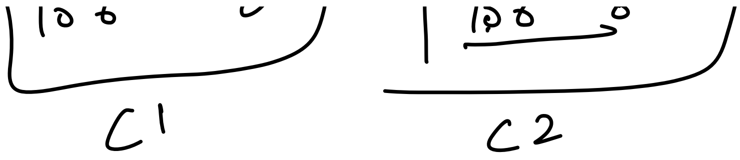
1	5
1	9
5	1
5	9
9	1
9	5



100	1
101	0
102	1
103	1
104	1
105	0

100	1 ..
101	0
102	1
103	1
104	1
105	0

```
Select c1.sid
from
Cinema  c1
JOIN  Cinema c2
ON   c1.f = 1 AND
```



C2.f = 1
AND
(c1.sid = c2.sid+1
or
) c1.sid = c2.sid-1

T1
T2
Down
Upload

Process