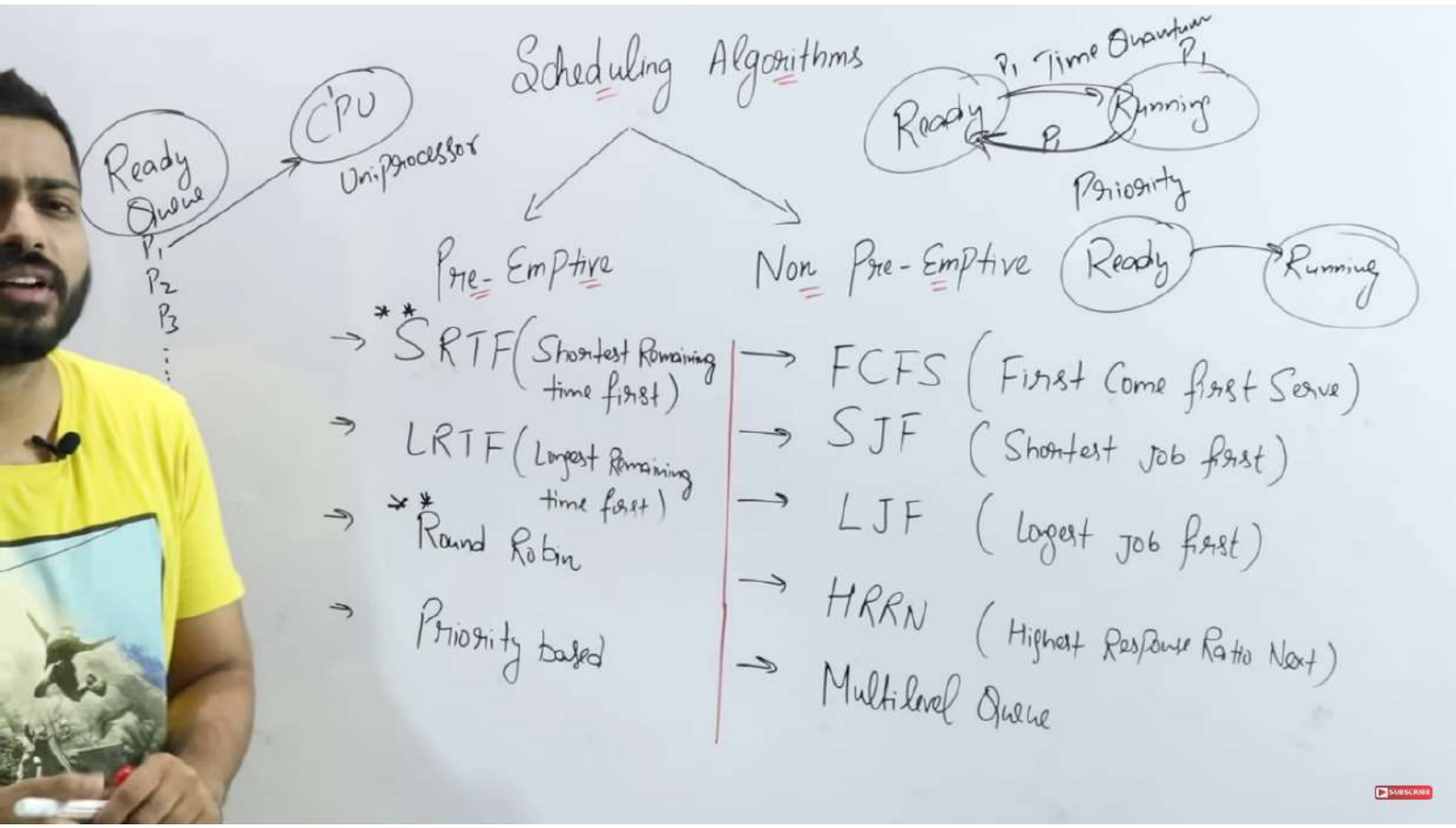

Scheduling Algorithms
Ready Queue
P1
P2
P3
CPU
Uniprocessor
P1 Time Quantum
Ready
Running
Priority
Ready
Running
Pre-Emptive
Non Pre-Emptive
** SRTF (Shortest Remaining time first)
LRTF (Longest Remaining time first)
** Round Robin
Priority based
FCFS (First Come first Serve)
SJF (Shortest job first)
LJF (Logest job first)
HRRN (Highest Response Ratio Next)
Multilevel Queue

"CPU Scheduling"

→ Arrival time: The time at which process enter the Ready Queue or State

Duration

→ Burst time: Time required by a process to get execute on CPU.

→ Completion time: The time at which process complete its execution.

→ Turn Around time: {Completion time − Arrival time}

12 − 11 = 1 hour = 60 minutes

CPU bound

I/O bound

→ Waiting time: {Turn Around time − Burst time}

60 minutes − 15 min = 45 minutes

→ Response time: [(The time at which a process get CPU first time) − (Arrival time)]

FCFS

Process No	Arrival Time	Execution Burst Time	Completion Time	TAT	WT	RT
~~P_1~~	0	2	2	2	0	0
~~P_2~~	1	2	4	3	1	1
~~P_3~~	5	3	8	3	0	0
~~P_4~~	6	4	12	6	2	2

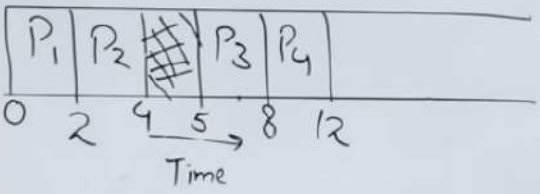
Criteria: "Arrival Time"

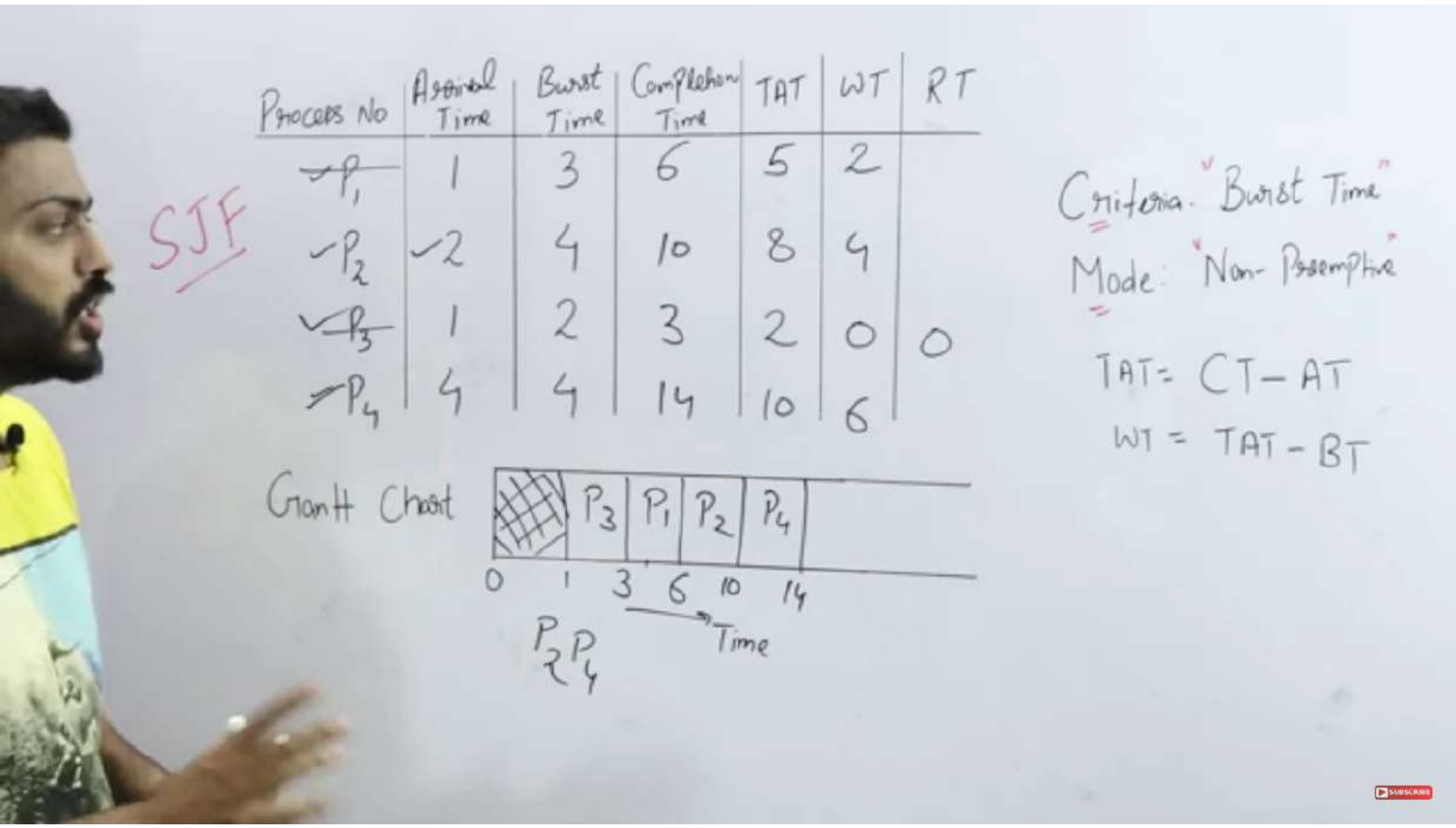
Mode: "Non-Preemptive"

CT - AT = TAT

TAT - BT = WT

Gantt Chart




SJF
Process No | Arrival Time | Burst Time | Completion Time | TAT | WT | RT
P1 | 1 | 3 | 6 | 5 | 2 |
P2 | 2 | 4 | 10 | 8 | 4 |
P3 | 1 | 2 | 3 | 2 | 0 | 0
P4 | 4 | 4 | 14 | 10 | 6 |
Criteria: "Burst Time"
Mode: "Non-Preemptive"
TAT = CT − AT
WT = TAT − BT
Gantt Chart
P3 P1 P2 P4
0 1 3 6 10 14
Time
P2 P4

SRTF

Process No	Arrival Time	Burst Time	Completion Time	TAT	WT	RT
P_1	0	~~5~~ 4 ~~3~~ 2 ~~1~~ 0	9	9	4	
~~P_2~~	1	~~3~~ 2 ~~1~~ 0	4	3	0	
P_3	2	~~4~~ 0	13	11	7	
~~P_4~~	4	~~1~~ 0	5	1	0	

P_1

Criteria: "Burst Time"

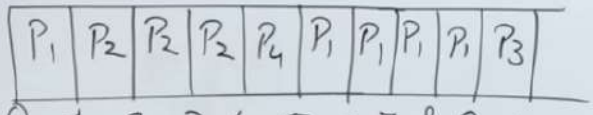
Mode: "Preemptive"

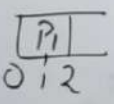
$TAT = CT - AT$

$WT = TAT - BT$

$RT = \{CPU\ first\ time - AT\}$

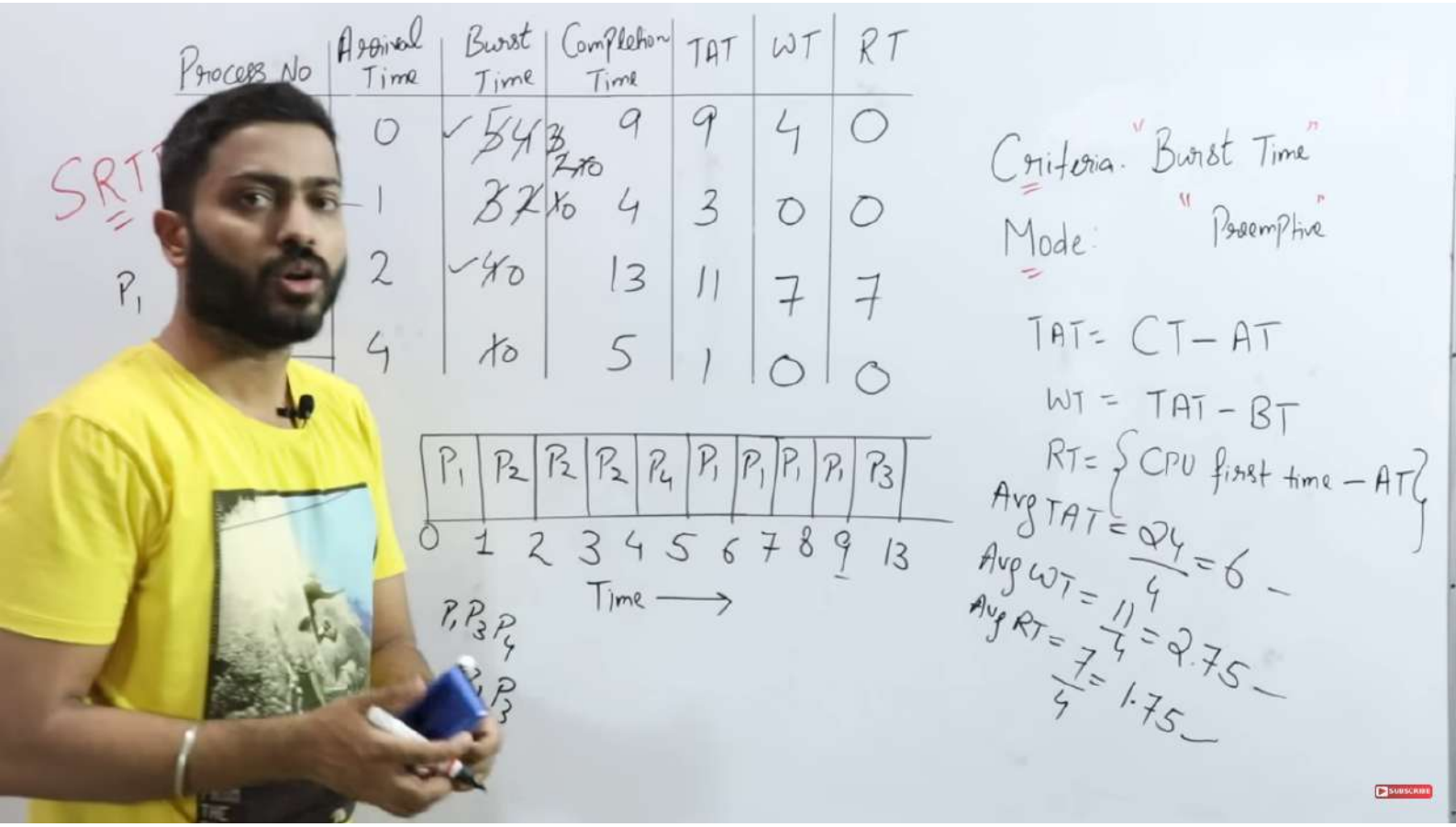
Gantt Chart

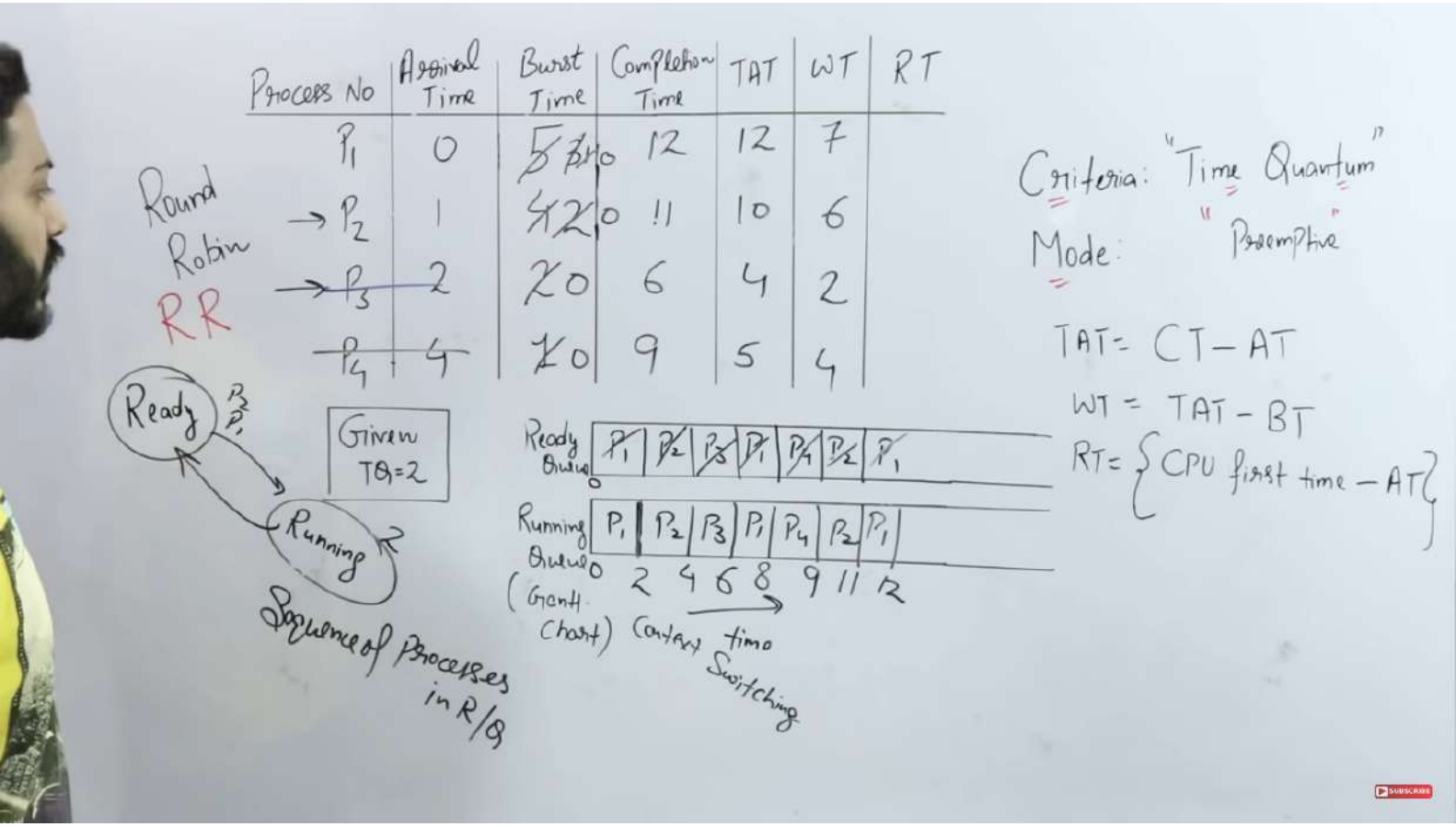


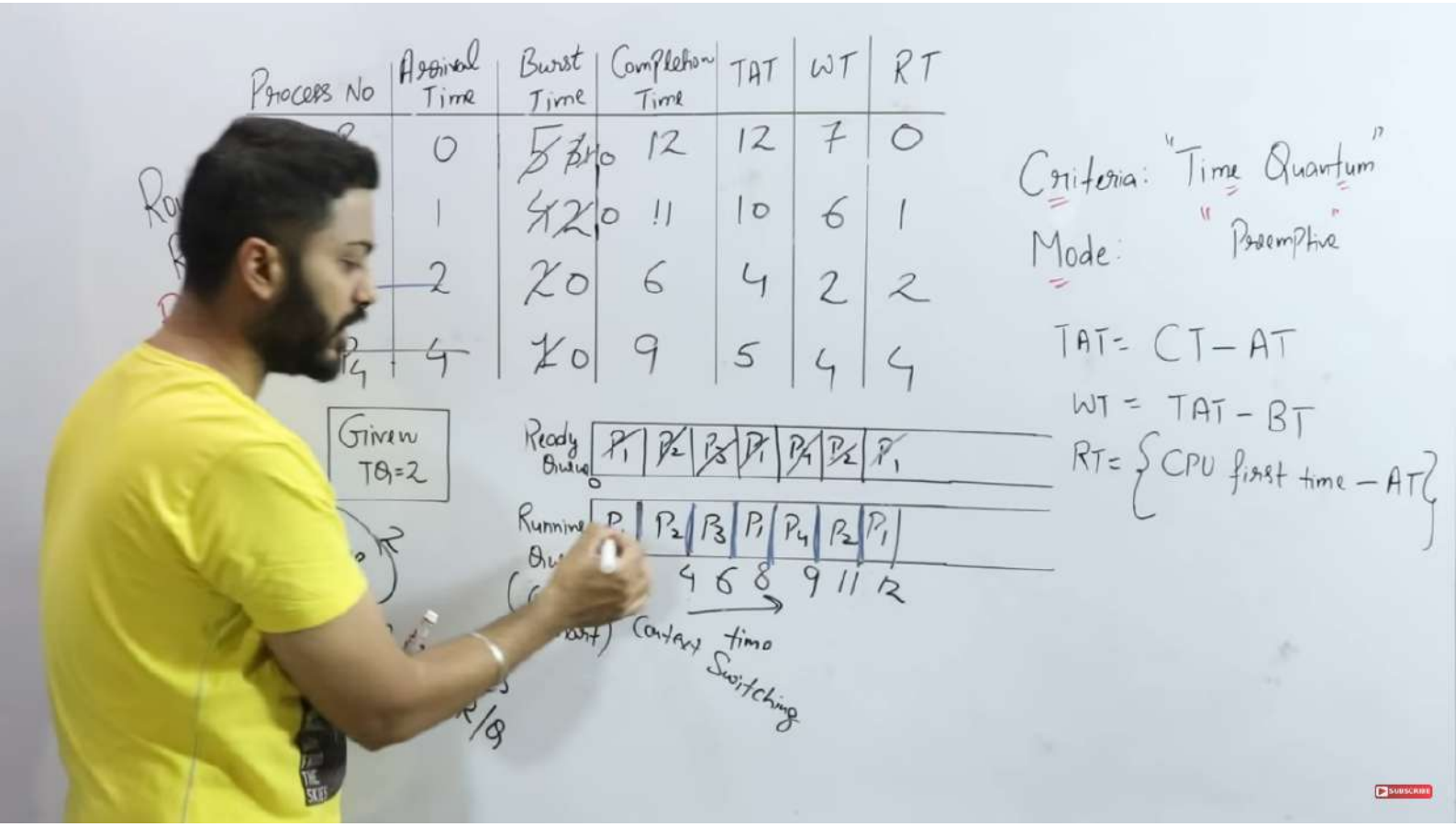


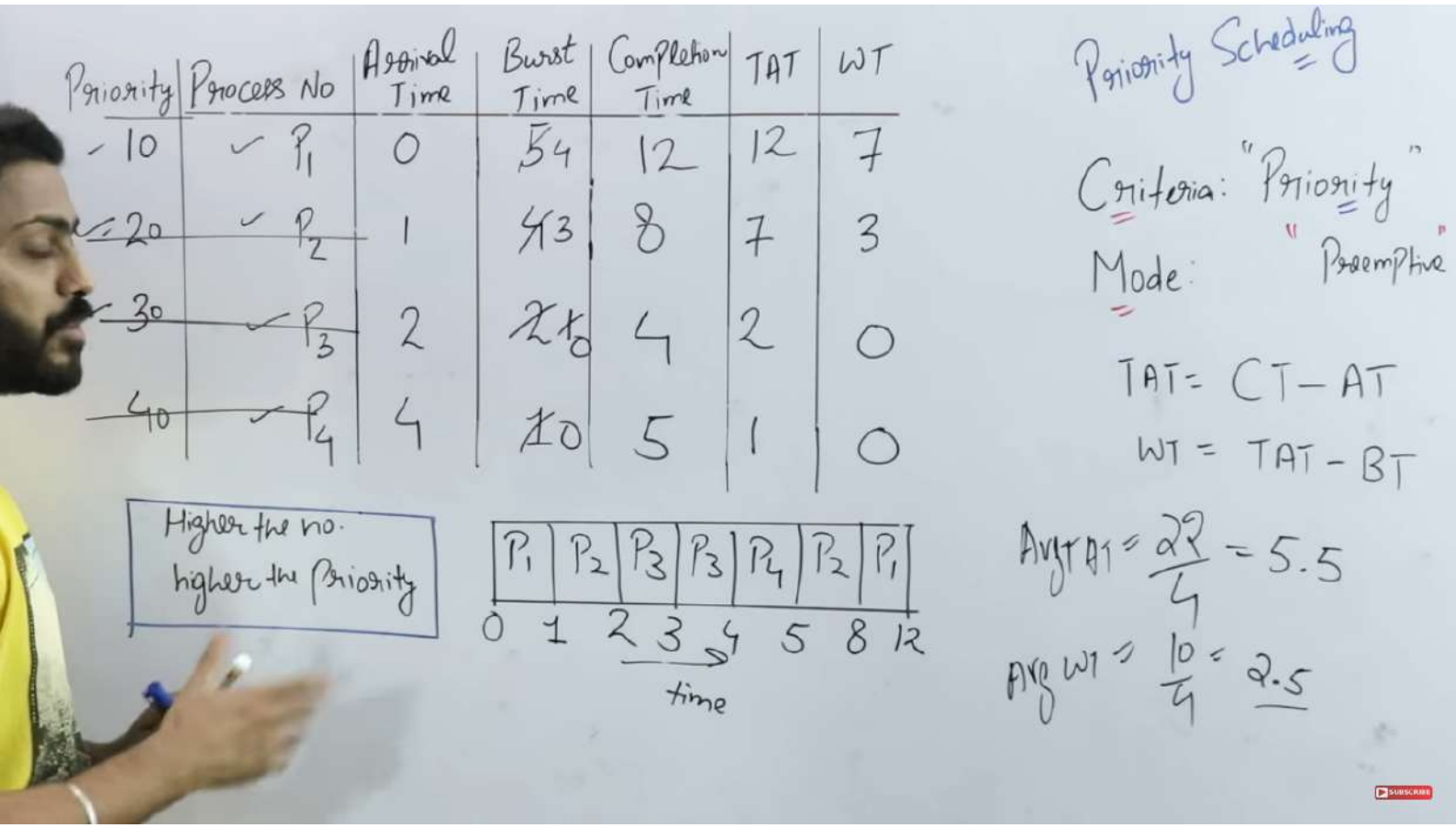
$P_1 P_2$
$P_1 P_2 P_3$
$P_1 P_2 P_3$

$P_1 P_3 P_4$
$P_1 P_3$


SRTF
Process No
Arrival Time
Burst Time
Completion Time
TAT
WT
RT
P1
0
1
2
4
9
4
3
0
13
11
7
5
1
0
P1 P2 P2 P2 P4 P1 P1 P1 P1 P3
0 1 2 3 4 5 6 7 8 9 13
Time
P1 P3 P4
Criteria: "Burst Time"
Mode: "Preemptive"
TAT = CT − AT
WT = TAT − BT
RT = { CPU first time − AT }
Avg TAT = 24/4 = 6
Avg WT = 11/4 = 2.75
Avg RT = 7/4 = 1.75


Process No | Arrival Time | Burst Time | Completion Time | TAT | WT | RT
P1 | 0 | 5 2 1 0 | 12 | 12 | 7
P2 | 1 | 4 2 0 | 11 | 10 | 6
P3 | 2 | 2 0 | 6 | 4 | 2
P4 | 4 | 2 0 | 9 | 5 | 4
Round Robin
RR
Criteria: "Time Quantum"
Mode: "Preemptive"
TAT = CT − AT
WT = TAT − BT
RT = { CPU first time − AT }
Ready
Running
Given TQ=2
Ready Queue: P1 P2 P3 P1 P4 P2 P1
Running Queue: P1 P2 P3 P1 P4 P2 P1
0 2 4 6 8 9 11 12
(Gantt Chart)
Context Switching
time
Sequence of Processes in R/Q


Process No
Arrival Time
Burst Time
Completion Time
TAT
WT
RT
0
12
12
7
0
1
11
10
6
1
2
6
4
2
2
4
9
5
4
4
Criteria: "Time Quantum"
Mode: "Preemptive"
TAT = CT - AT
WT = TAT - BT
RT = { CPU first time - AT }
Given TQ=2
Ready Queue
Running Queue
4 6 8 9 11 12
time
Context Switching


Priority | Process No | Arrival Time | Burst Time | Completion Time | TAT | WT
10 | P1 | 0 | 4 | 12 | 12 | 7
20 | P2 | 1 | 3 | 8 | 7 | 3
30 | P3 | 2 | 2 | 4 | 2 | 0
40 | P4 | 4 | 1 | 5 | 1 | 0
Higher the no. higher the Priority
P1 P2 P3 P3 P4 P2 P1
0 1 2 3 4 5 8 12
time
Priority Scheduling
Criteria: "Priority"
Mode: "Preemptive"
TAT = CT - AT
WT = TAT - BT
Avg TAT = 22/4 = 5.5
Avg WT = 10/4 = 2.5

CPU Scheduling :-

→ A process execution consist of a cycle of CPU execution and I/O execution.

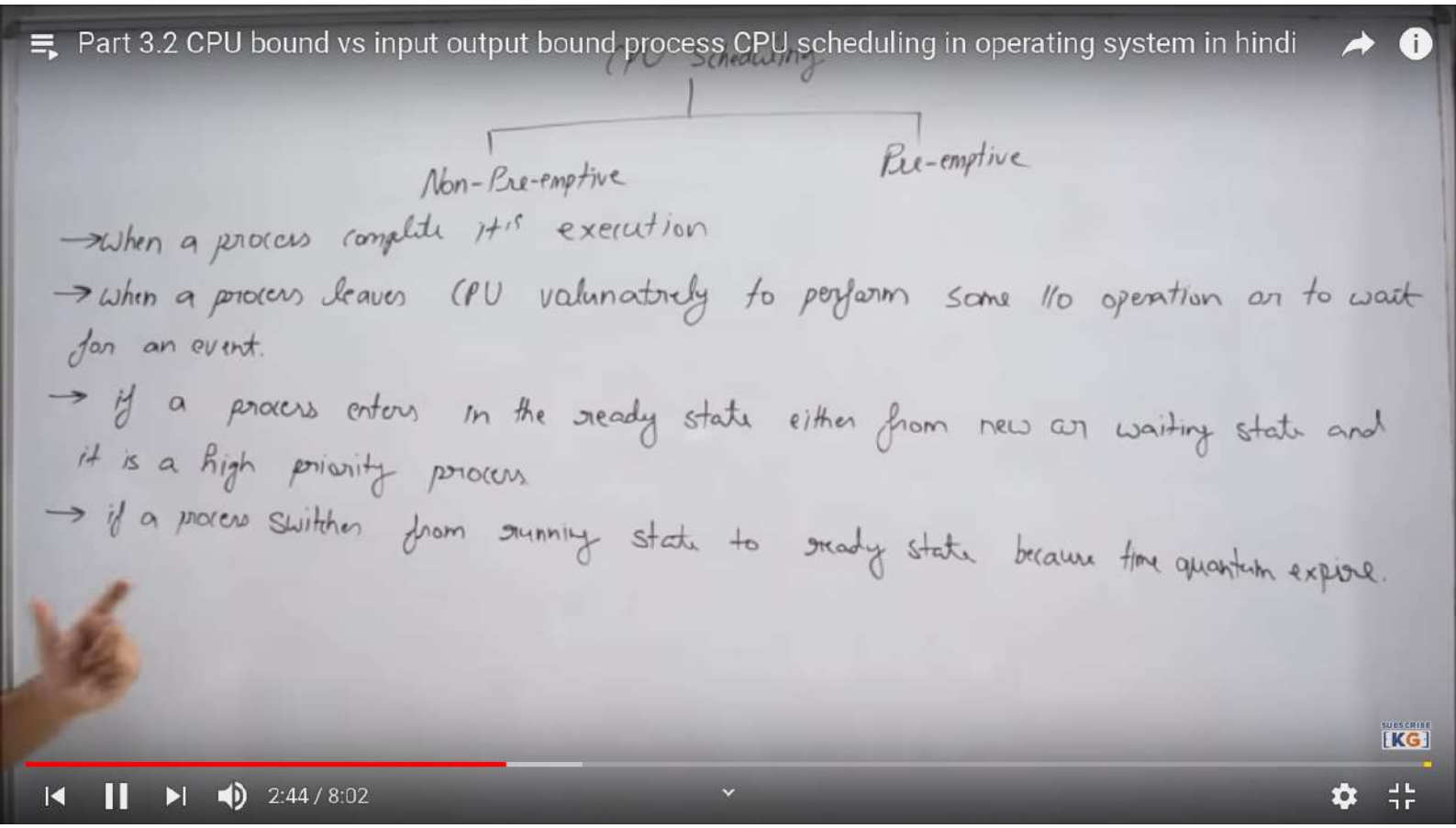
→ Normally every process begins with CPU burst that may be followed by I/o burst, then another CPU burst and then I/o burst and so on. eventually in the last will end up on CPU burst.

CPU bound process :- There are those processes which require most of time on CPU.

I/o bound process :- There are those process which require most of the time on I/o devices or peripherals.

Conclusion :- A good CPU scheduling idea should choose the mixture of both so that both I/o devices and CPU can be utilised efficiently.

Part 3.2 CPU bound vs input output bound process CPU scheduling in operating system in hindi



2:44 / 8:02

Part 3.3 Burst time Waiting time Arrival time Exit time Turnaround time Response time Average

CPU Scheduling terminology:-

→ Burst time/execution time/ running time:- is the time process require for running on CPU.

→ Waiting time:- time spend by a process in ready state waiting for CPU.

→ Arrival time:- when a process enters Ready state

→ Exit time:- When process completes execution and exit from system

Turn around time:- Total time spend by a process in the system

$$T.A.T = E.T - A.T = B.T + W.T$$

Response time:- Time between a process enters ready queue and get scheduled on the CPU for the first time.

Criteria for CPU Scheduling algorithm :-

→ Average waiting time

→ Average response time

→ CPU utilization

→ Throught put

First Come First Serve (FCFS) :-

→ Simplest Scheduling algorithm. It assign CPU to the process which Arrives first.

→ easy to understand and can easily be implimented using Queue data structure

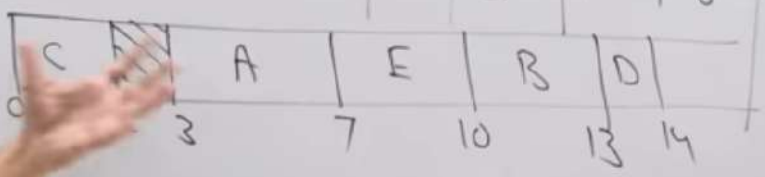
→ Always non-pre-emptive in nature.

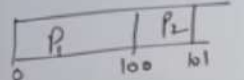
T.A.T = BT + W.T

W.T = TAT - B.T

Pid	A.T	B.T	TAT = E.T - A.T	W.T = TAT - B.T
A	3	4	7 - 3 = 4	4 - 4 = 0
B	5	3	13 - 5 = 8	8 - 3 = 5
C	0	2	2 - 0 = 2	2 - 2 = 0
D	5	1	14 - 5 = 9	9 - 1 = 8
E	4	3	10 - 4 = 6	6 - 3 = 3

Gantt chart :-

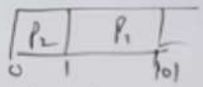




Pid	A.T	B.T	W.T
P1	0	100	0
P2	1	1	99

$= 49.5$

Pid	A.T	B.T
P1	1	100
P2	0	1



Convoy effect: Smaller process have to wait for long time for bigger process to release CPU

Advantage: Simple to use, easy to understand, easy to implement, must be used for back ground process where execution is not urgent.

Disadvantage: - Suffer from convoy effect, normally higher average waiting time, no consideration to priority, Worst time, should not be used for interactive system.

Shortest Job First (non-pre-emptive) / Shortest Remaining time First (SRTF) (Pre-emptive)

→ Out of all available process, CPU is assigned to the process having smallest burst time requirement (no priority, no seniority)

→ If there is a tie, FCFS is used to break tie

→ Can be used both with non-pre-emptive and pre-emptive approach

→ pre-emptive version (SRTF) is also called as optimal as it gurantee minimal average waiting time.

Pid	A.T	B.T
P_1	3	1
P_2	1	4
P_3	4	2
P_4	0	6
P_5	2	3

Shortest Job First (non-pre-emptive) / Shortest Remaining time First (SRTF) (pre-emptive)

→ Out of all available process, CPU is assigned to the process having smallest burst time requirement (no priority, no seniority)

... is a tie, FCFS is used to break tie

... both with non-pre-emptive and ...proach

...ion (SRTF) is also called as ... guarantee minimal average

Pid	A.T	B.T	TAT	W.T
× P_1	3	1	7-3=4	4-1=3
× P_2	1	4	16-1=15	15-4=9
× P_3	4	2	9-4=5	5-2=3
× P_4	0	6	6-0=6	6-6=0
× P_5	2	3	12-2=10	10-3=7

P_4	P_1	P_3	P_5	P_2

0 6 7 9 12 16

Shortest Job First (non-pre-emptive) / Shortest Remaining time First (SRTF) (pre-emptive)

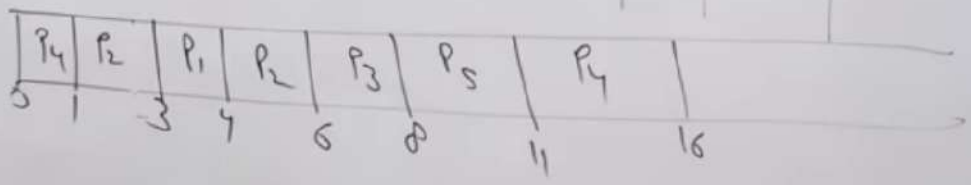
→ Out of all available process, CPU is assigned to the process having smallest burst time requirement (no priority, no seniority)

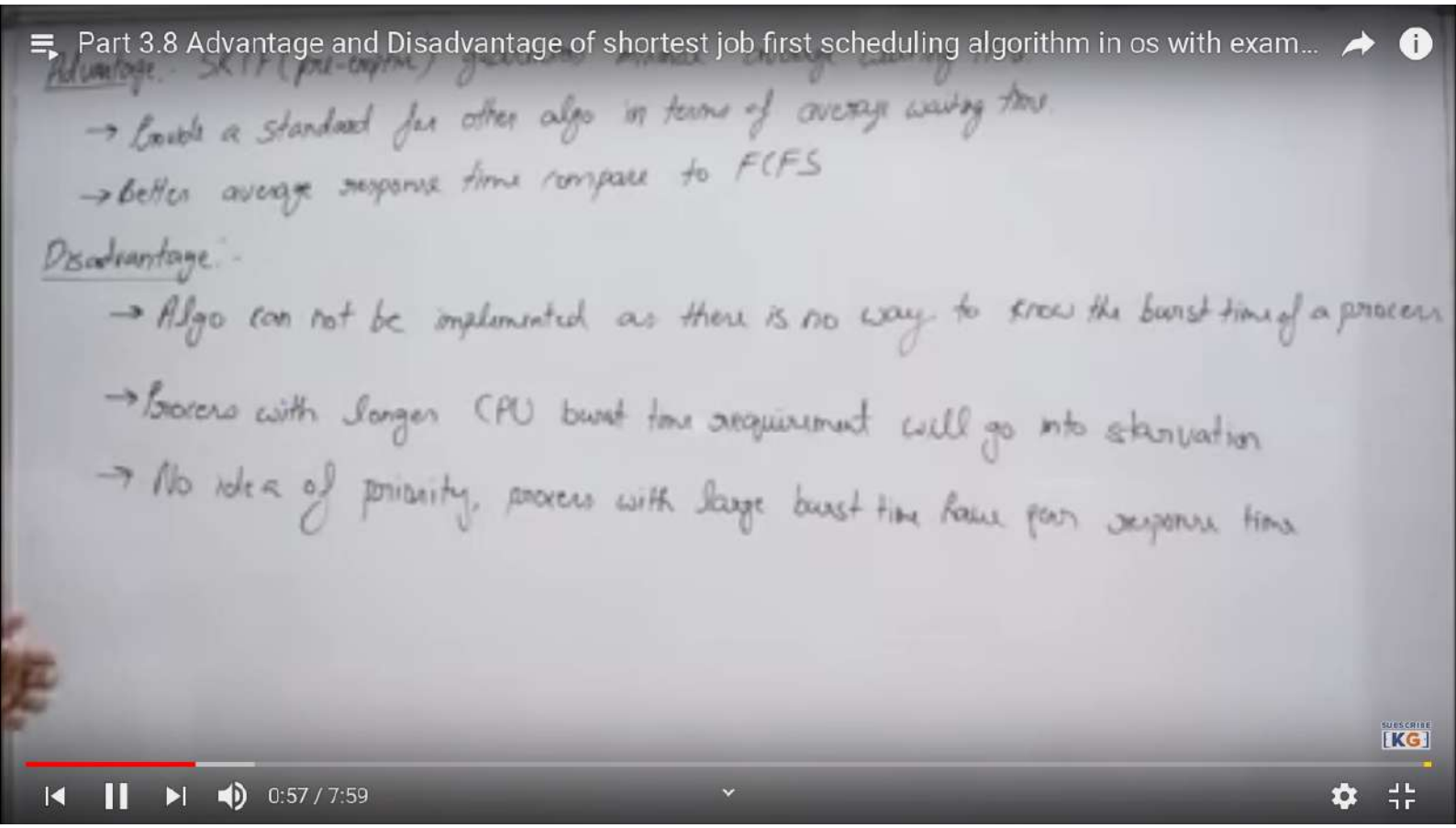
→ If there is a tie, FCFS is used to break tie

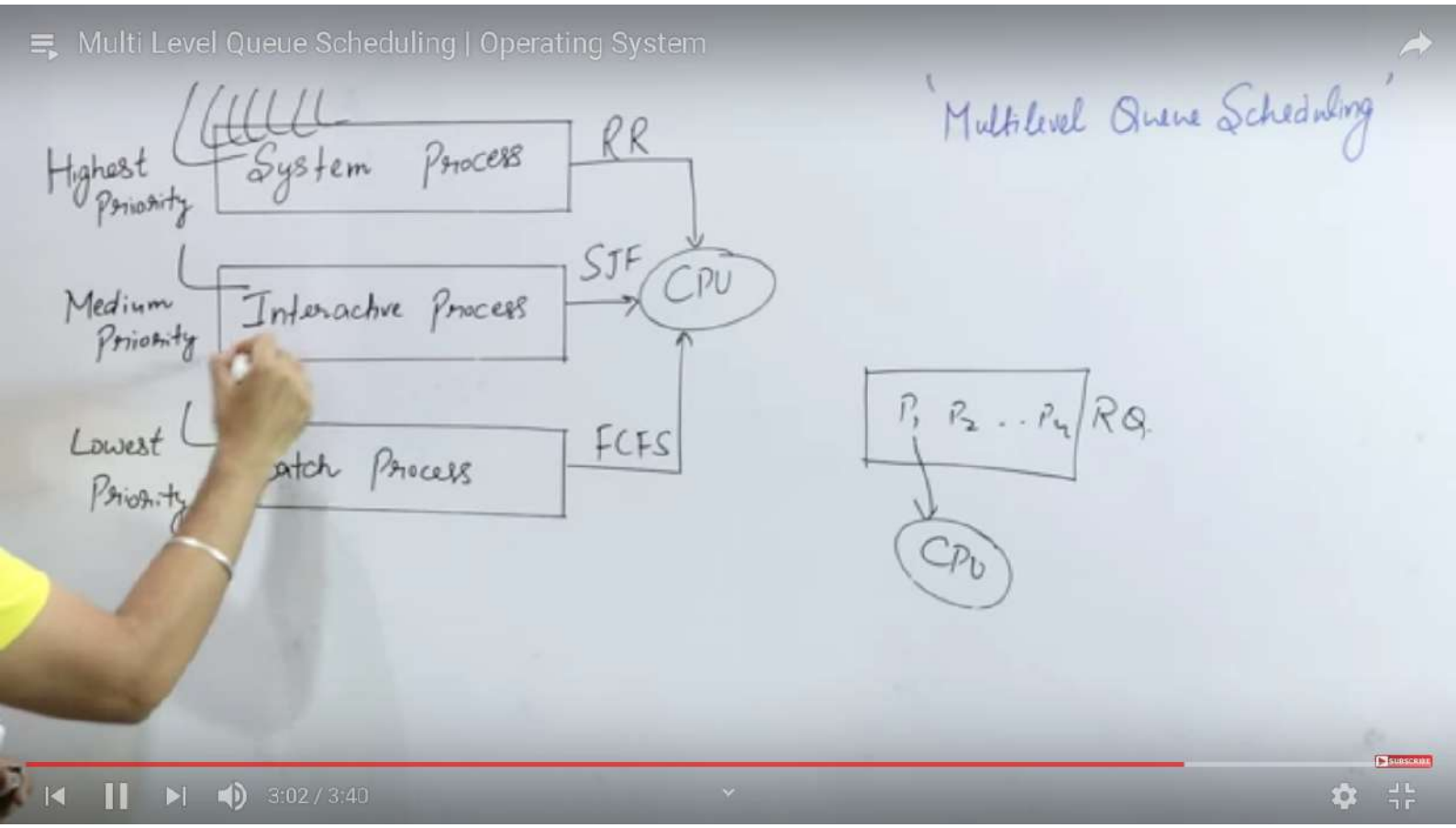
→ [illegible] used both with non-pre-emptive and [illegible] approach

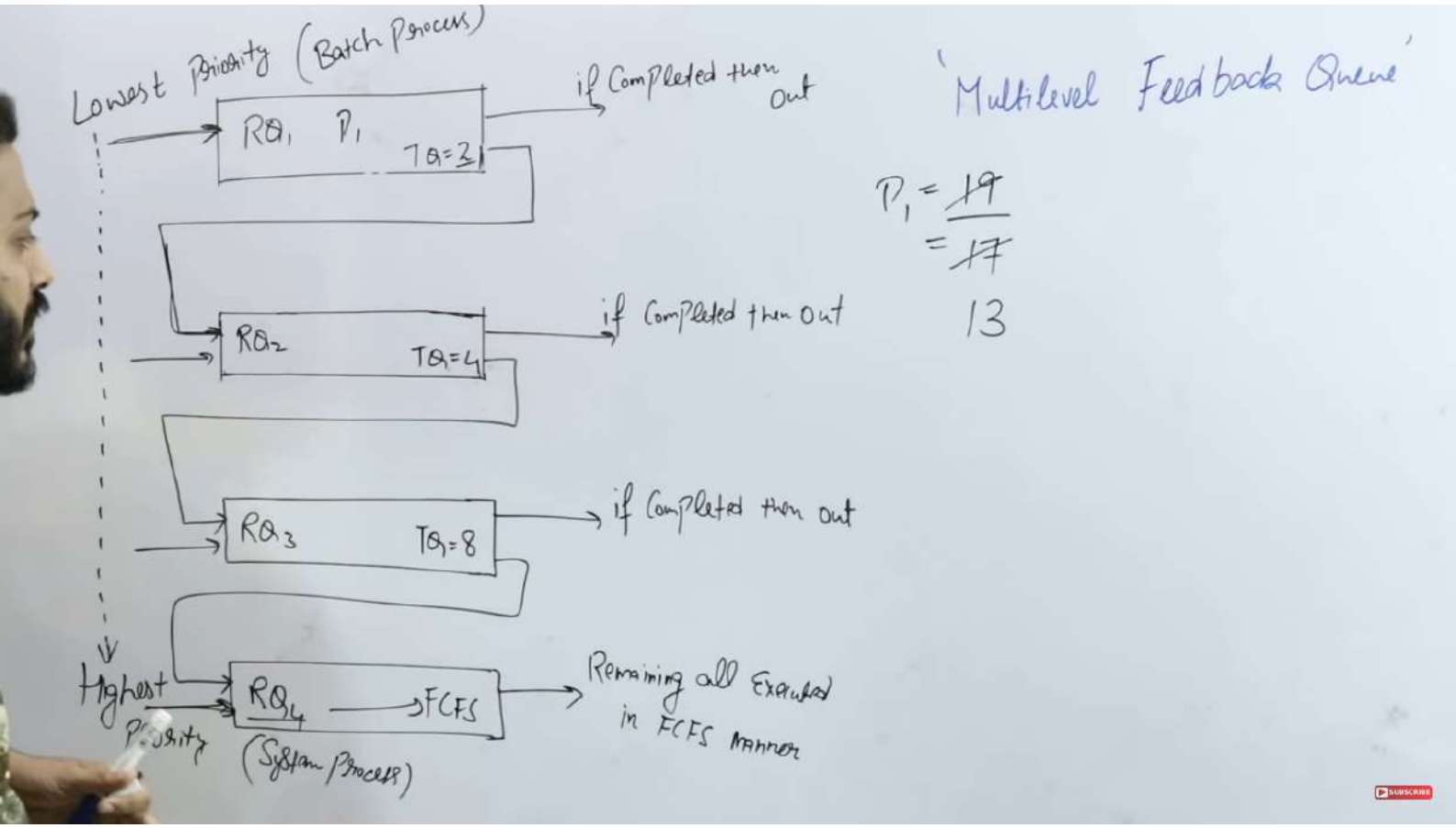
→ [illegible] version (SRTF) is also called as [illegible] as it gurantee minimal average [illegible] time.

Pid	A.T	B.T	TAT	W.T
~~P_1~~	~~3~~	~~1~~		
P_2	1	~~4~~ ~~3~~ 2		
~~P_3~~	~~4~~	~~2~~		
~~P_4~~	~~0~~	~~8~~ 5		
~~P_5~~	~~2~~	~~3~~		




Part 3.8 Advantage and Disadvantage of shortest job first scheduling algorithm in os with exam...
Advantage:- SJF (pre-emptive) gives minimum average waiting time
→ Could a standard for other algo in terms of average waiting time
→ Better average response time compare to FCFS
Disadvantage:-
→ Algo can not be implemented as there is no way to know the burst time of a process
→ Process with longer CPU burst time requirement will go into starvation
→ No idea of priority, process with large burst time have poor response time
SUBSCRIBE
KG
0:57 / 7:59


Multi Level Queue Scheduling | Operating System
'Multilevel Queue Scheduling'
Highest Priority
System Process
RR
Medium Priority
Interactive Process
SJF
CPU
Lowest Priority
atch Process
FCFS
P1 P2 .. Pn
RQ
CPU
3:02 / 3:40


Lowest Priority (Batch Process)
RQ1 P1
TQ=31
if Completed then out
RQ2
TQ=4
if Completed then out
RQ3
TQ=8
if Completed then out
Highest Priority
RQ4
FCFS
(System Process)
Remaining all Executed in FCFS Manner
'Multilevel Feedback Queue'
P1 = 19
= 17
13