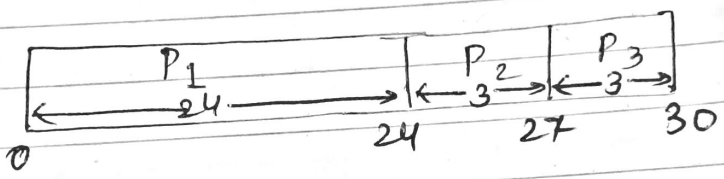
1) FCFS

Q) There are three process P_1, P_2, P_3. Burst time of these processes is 24, 3, 3. Using FCFS scheduling algo draw the Gantt chart and calculate waiting time, turnaround time of each process and average waiting time and throughput.

Sol)

Process	BT	WT	TAT	CT
P_1	24	24-24 = 0	24	24
P_2	3	27-3 = 24	27	27
P_3	3	30-3 = 27	30	30

Gantt chart



Let the arrival time be 0 for all process as if is not given

$$\text{Avg. waiting time} = \frac{0+24+27}{3} = 17$$

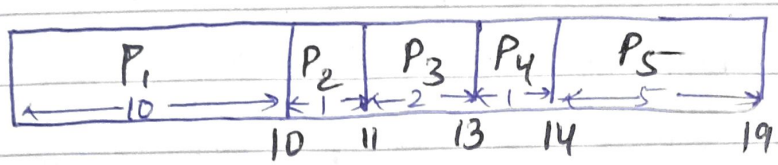
$$\text{Avg. TAT} = \frac{24+27+30}{3} = 27$$

$$\text{Throughput} = \frac{3}{30} = \frac{1}{10} = 0.1 \text{ unit.}$$

Q)

Process	BT	WT	TAT	CT
P_1	10	0	10	10
P_2	1	10	11	11
P_3	2	11	13	13
P_4	1	13	14	14
P_5	5	14	19	19

Gantt Chart



Avg. WT = 9.6

Avg. TAT = 13.4

Throughput = $\frac{5}{19}$ = 0.26

Q)

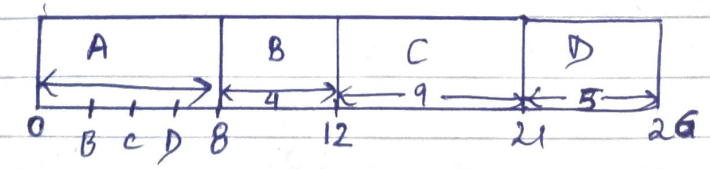
Process	AT	B.T.	TAT	WT	C.T.
A	0	8	8	0	8
B	1	4	11	7	12
C	2	9	19	10	21
D	3	5	23	18	26

AWT = 8.75

ATAT = 15.25

Throughput = 0.15

Gantt chart

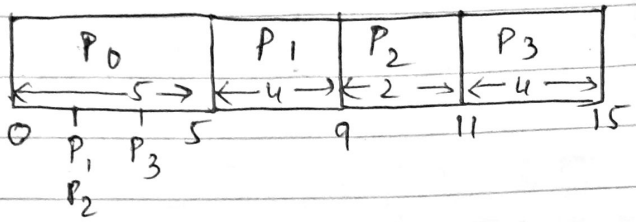


Q)

Process	AT	BT	TAT	WT	CT	Priority
P_0	0	5	5	0	5	4
P_1	2	4	7	3	9	2
P_2	2	2	9	7	11	6
P_3	4	4	11	7	15	3

Find TAT, WT, AWT, ATAT and throughput

Gantt chart



AWT = 4.25

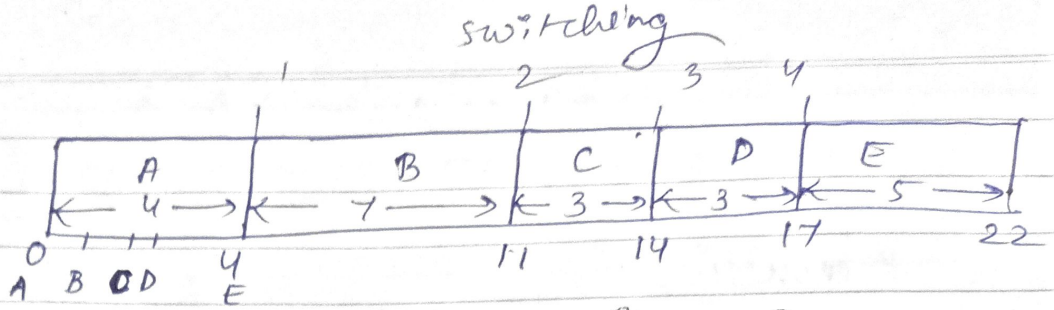
ATAT = 8

Throughput = $\frac{4}{15}$ = 0.266

Here, ignoring given priority because it is FCFS based ques

Q) Consider the set of processes given in table and FCFS scheduling algo.

Process ID	AT	ET	CT	TAT	WT
A	0	4	4	4	0
B	2	7	11	9	2
C	3	3	14	11	8
D	3.5	3	17	13.5	10.5
E	4	5	22	18	13



i) If the scheduler takes 0.2 units of CPU time in content switching for the completed jobs and 0.1 unit of additional CPU time for incomplete jobs for saving there content Calculate the percentage of CPU time wasted.

AWT = 6.7

ATAT = 11.1

Throughput = $\frac{5}{22}$ = 0.227

Content Switching

completed job = 0.2 × 4 = 0.8

incompleted job = 0.2 + 0.1 = 0.3

CPU time wasted = $\frac{0.8}{22} \times 100$ = 3.636 %

2) Shortest Job First

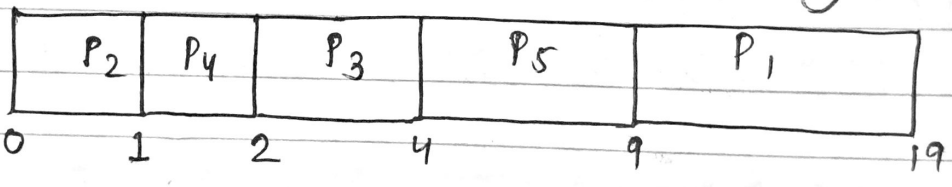
Preemptive (Burst time) | non-Preemptive (Complete)

i) SJF (Non-Preemptive) → We will execute that process first, after completion of present job which has ↓ B.T.

Process	BT	CT	TAT	WT
P_1	10	19	19	9
P_2	1	1	1	0
P_3	2	4	4	2
P_4	1	2	2	1
P_5	5	9	9	4

Find TAT, WT, CT, AWT, ATAT and Throughput

Gantt chart



Let A.T = 0 for all process

Now, AWT = 3.2

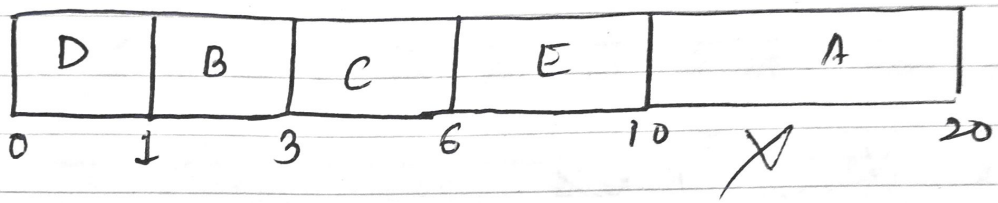
ATAT = 7

Throughput = 0.26

Q2)

Process	A.T.	E.T	Priority
A	0	10	3
B	0	2	1
C	1	3	3
D	2	1	5
E	2	4	2

Gantt chart

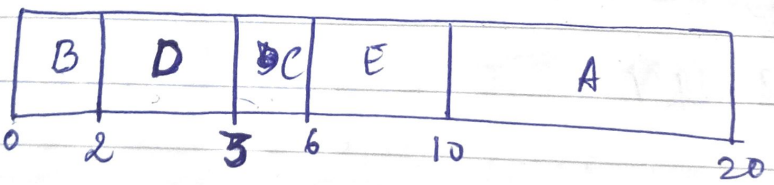


Acc to A.T. and E.T.

Ready queue i) [A, B] as A, B = 0 A.T.
so, B has less E.T.
so, add it to gantt chart

ii) [A, D] as D has less E.T. so, D add to gantt chart

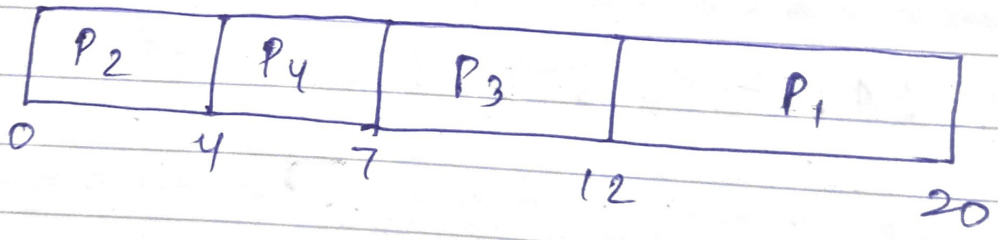
Similarly,



Q) No. of Process

Process	B.T.	AT	CT	TAT	WT
P_1	8	0	20	20	12
P_2	4	0	4	4	0
P_3	5	1	12	11	6
P_4	3	2	7	5	2

Gantt chart



AWT = 5

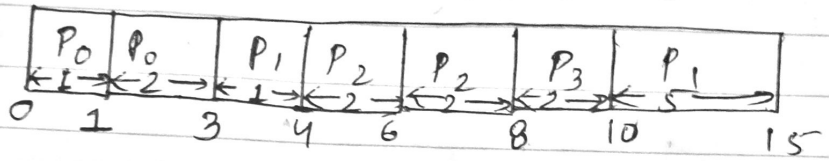
ATAT = 10

Throughput = $\frac{4}{20}$ = 0.2

3) Shortest Remaining Time first (SRTF) / SJF (Preemptive)

Process	AT	BT	CT	TAT	WT
P_0	0	3	3	3	0
P_1	1	6	15	14	8
P_2	4	4	8	4	0
P_3	6	2	10	4	2

Gantt chart



AWT = 2.5

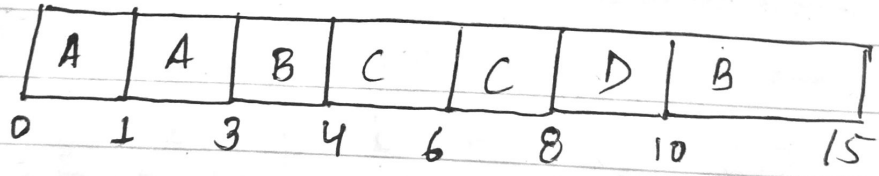
ATAT = 6.25

Throughput = 0.267

Q)

Process	AT	BT	CT	TAT	WT
A	0.000	3	3	3	0
B	1.001	6	15	14	8
C	4.001	4	8	4	0
D	6.001	2	10	4	2

Gantt chart



AWT = 2.5

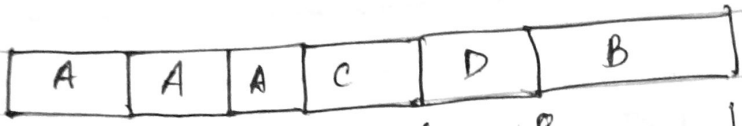
ATAT = 6.25

Throughput = 0.267

Q)

Process	AT	BT	CT	TAT	WT
A	0.000	4	21	4	0
B	2.001	7	15	13	6
C	3.001	2	6	3	1
D	3.002	2	8	5	3

Gantt chart



AWT = 2.5

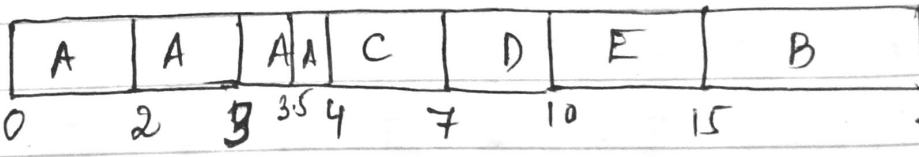
ATAT = 6.25

Throughput = 0.267

Q)

Process	AT	ET	CT	TAT	WT
A	0	4	4	4	0
B	2	7	22	20	13
C	3	3	7	4	1
D	3.5	3	10	6.5	3.5
E	4	5	15	11	6

Gantt chart



AWT = 4.7

ATAT = 9.1

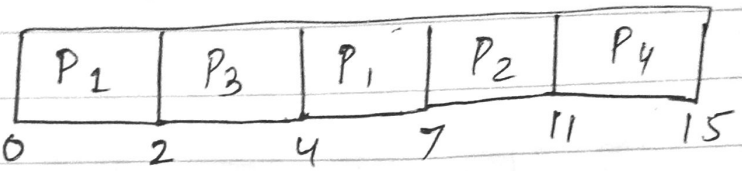
Throughput = $\frac{5}{22}$ = 0.227

Priority will be considered in priority scheduling Algo only.

Q)

Process	AT	BT	Priority	CT	TAT	WT
P_1	0	5	4	7	7	2
P_2	2	4	2	11	9	5
P_3	2	2	6	4	2	0
P_4	4	4	3	15	11	7

Gantt chart



AWT = ~~7.25~~ 3.5

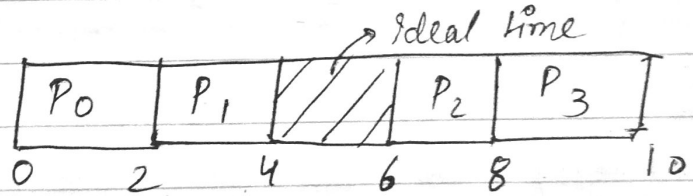
ATAT = ~~5.5~~ 7.25

Throughput = $\frac{5}{15}$ = 0.33

Q)

Process	AT	BT	CT
P_0	0	2	2
P_1	2	2	4
P_2	6	2	8
P_3	7	2	10

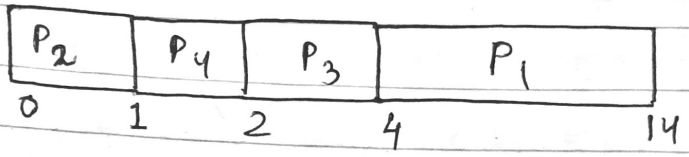
Gantt chart



Q) SJF non-preemptive

Process	BT	CT	TAT	WT
P_1	10	14	14	4
P_2	1	1	1	0
P_3	2	4	4	2
P_4	1	2	2	1

Gantt chart

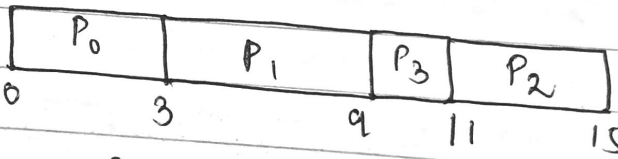


Q) Process AT BT TAT WT CT

P_0	0.0	3	3	0	3
P_1	1.0	6	8	2	9
P_2	4.0	4	11	7	15
P_3	6.0	2	5	3	11

SJF non-preemptive

Gantt chart



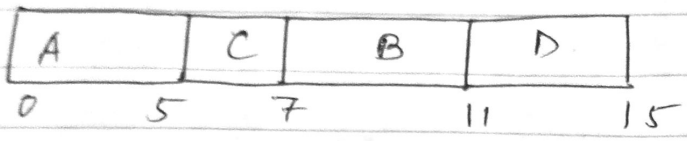
ATAT = $\frac{27}{4}$ = 6.75 AWT = $\frac{12}{4}$ = 3

Throughput = $\frac{4}{15}$ = 0.26

Time / Quantum or Time slice = only in PR

Q) Process | AT | BT | Priority | TAT | WT | CT

Process	AT	BT	Priority	TAT	WT	CT
A	0.0001	5	4	5	0	5
B	2.0001	4	2	9	5	11
C	2.0001	2	6	5	3	7
D	4.0001	4	3	11	7	15



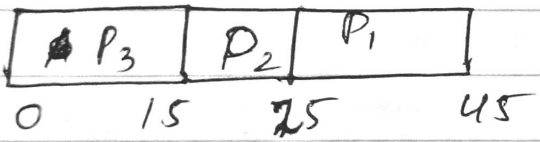
ATAT = 7.5

AWT = 3.75

Throughput = 0.26

Q) Process | AT | BT | TAT | WT | CT

Process	AT	BT	TAT	WT	CT
P_1	5	20	40	20	45
P_2	3	10	22	12	25
P_3	0	15	15	0	15

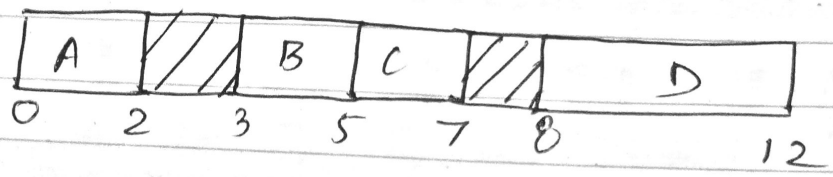


$ATAT = \frac{77}{3}$ $\quad AWT = \frac{32}{3}$

$Throughput = \frac{3}{45} = \frac{1}{15}$

Q) SJF preemptive

Process	AT	WT	Priority	TAT	WT
A	0.0000	2	4	2	0
B	3.0001	2	2	2	0
C	3.0001	2	6	4	2
D	8.0001	4	3	4	0



ATAT = 3

AWT = $\frac{1}{2}$ = 0.5

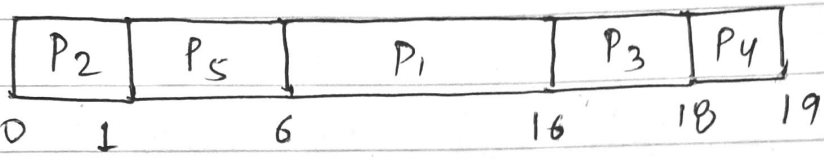
Throughput = $\frac{4}{12}$ = 0.33

4) Priority scheduling :- (By non-preemptive)

Q)

Process	B.T.	Priority	C.T.	TAT (CT-AT)	WT (TAT-BT)
P1	10	3	16	16	6
P2	1	1 (Highest)	1	1	0
P3	2	4	18	18	16
P4	1	5	19	19	18
P5	5	2	6	5	1

Gantt chart



AWT = 8.2

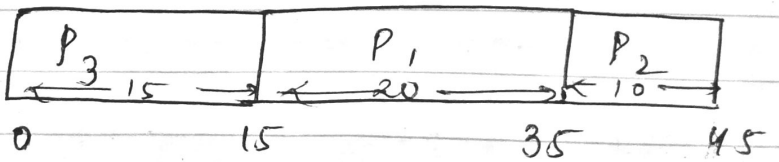
ATAT = 12

Throughput = $\frac{5}{19}$ = 0.263

Q)

Process	A.T.	BT	Priority	CT	TAT	WT
P1	5	20	10 (Highest)	35	30	10
P2	3	10	9	45	42	32
P3	0	15	8	15	15	0

Gantt chart

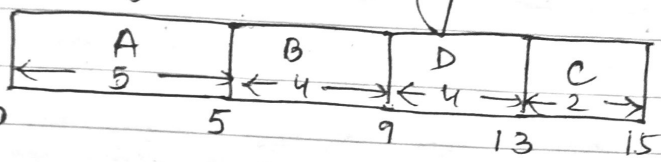


Q) 8

Process	AT	BT	Priority	CT	TAT	WT
A	0.0000	5	4	5	5	0
B	2.0001	4	2	9	7	3
C	2.0001	2	6	15	13	11
D	4.0001	4	3	13	9	5

Let B has highest priority.

Gantt chart



AWT = 4.75

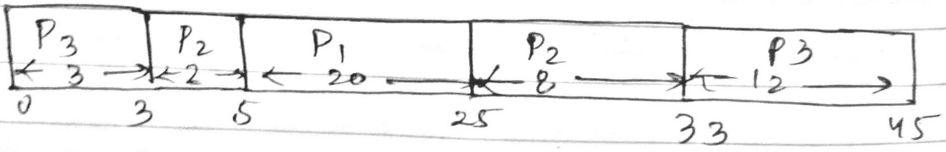
ATAT = 8.5

Throughput = $\frac{4}{15}$ = 0.266

Q) Do using <u>preemptive</u> priority scheduling

Process	AT	BT	Priority	CT	TAT	WT
P1	5	20	10 (highest)	25	20	0
P2	3	10	9	33	30	20
P3	0	15	8	45	45	30

Gantt chart



AWT = 16.66

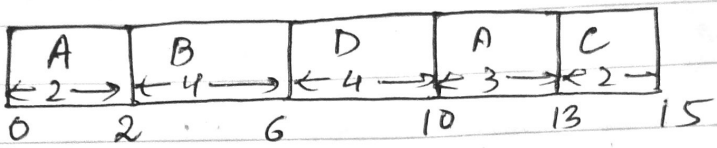
ATAT = 31.66

Throughput = $\frac{3}{45}$ = 0.066

Q)

Process	AT	BT	Priority	CT	TAT	WT
A	0.0000	5	4	13	13	8
B	2.0001	4	2 (Highest)	6	4	0
C	2.0001	2	6	15	13	11
D	4.0001	4	3	10	6	2

Gant chart



AWT = 5.25

ATAT = 9

Throughput = $\frac{4}{15}$ = 0.266

5) Round Robin (RR) scheduling :—

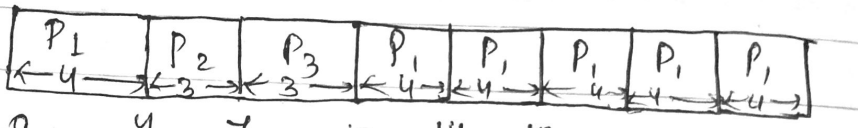
- It is designed especially for time sharing system.
- A small unit of time called "Time quantum" or "Time slice" is defined.
- It is Preemptive algo.
- The ready queue is treated as circular queue

Q)

Process	Burst Time	CT	TAT	WT
P1	24	30	30	6
P2	3	7	7	4
P3	3	10	10	7

Time quantum = 4 millisec

Gantt chart



ATAT = 15.66
AWT = 5.666

Q)

Process	BT.	CT	TAT	WT
P	4	11	11	7
Q	1	2	2	1
R	8	16	16	8
S	1	4	4	3
T	2	8	8	6

Time quantum = 1

starvation → CPU waiting for job to execute but never get process

Imp
- ideal time is only in SRTF.
- .001 will be considered only in RR algo.

Gantt chart

P	Q	R	S	T	P	R	T	P	R	P	R	R	R	R	R
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

AWT = 5

ATAT = 8.2

Throughput = $\frac{5}{16}$ = 0.312

Q)

Process	A.T.	B.T.
A	0.000	4
B	2.001	7
C	3.001	2
D	3.002	2

a) Time quantum = 2

b) Time quantum = 1

Gantt chart

A	B	C	D	A	B	C	D	A	B	A	B	B	B	B
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

✗

no break in round robin

B

✗ draw acc. to time quantum

a) Gantt chart

A	A	B	C	D	B	B	B
0	2	4	6	8	10	12	14	15

✓

Process	CT	TAT	WT
A	4	4	0
B	15	13	6
C	8	5	3
D	10	7	5

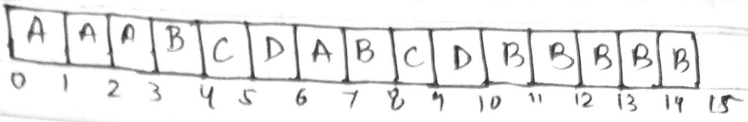
AWT = 3.5

ATAT = 7.25

T.P. = 0.266

Preemptive i.e. two types.
→ Non-
Passive entity → which do not execute (i.e. Proc
Active entity → which executes (i.e, process)

b) Gantt chart



Process	CT	TAT	WT
A	7	7	3
B	15	13	6
C	9	6	4
D	10	7	5

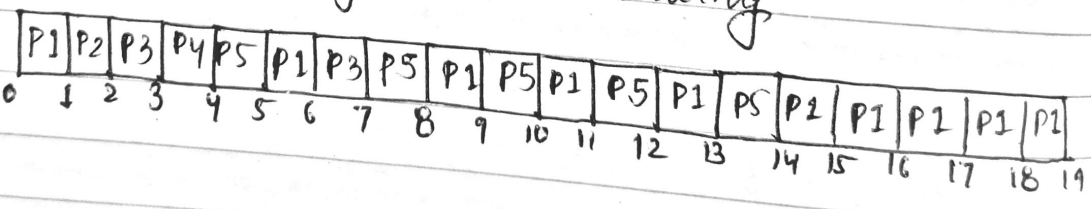
AWT = 4.5
ATAT = 8.25
T.P. = 0.266

Q)

Process	B.T	C.T.	WT.	TAT.
P1	10	19	9	19
P2	1	2	1	2
P3	2	7	5	7
P4	1	4	3	4
P5	5	14	9	14

Time quantum = 1 , by RR scheduling

Gantt chart



AWT = 5.4
ATAT = 9.2
Throughput = $\frac{5}{19}$ = 0.263

A B A B C D B C D
2 4 6 8 10

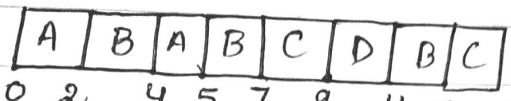
Imp

Q)

Process	A·T	BT	C·T.
A	0·000	3	
B	1·001	6	
C	3·001	4	
D	6·002	2	

a) T Q = 2
b) T·Q = 1

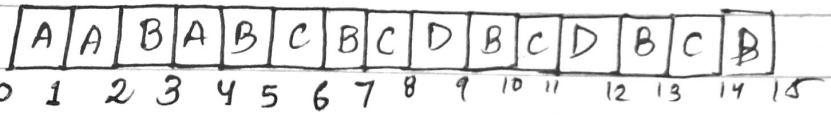
a) Gantt chart



ABA

Process	C·T.	TAT.	W·T.
A	5	5	2
B	13	12	6
C	15	11	7
D	11	5	3

b) Gantt chart



Process	C·T	T·A·T	W·T.
A	4	4	1
B	15	14	8
C	14	10	6
D	12	6	4

While solving above ques remember that queue is circular in nature.

→ emptive ,

long term scheduler ⇒ loads new process
CPU ⇒ loads currently executing process
and CPU is faster than LTS.

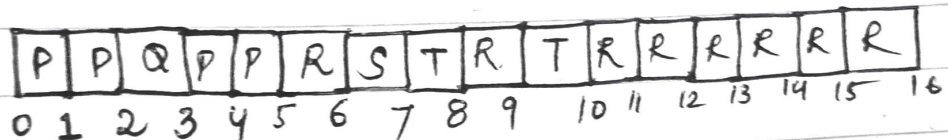
Q)

Process	B-T	A-T
P	4	0
Q	1	01
R	8	4
S	1	4
T	2	6

Time quantum = 1

Gantt chart

Gantt chart



P P Q P P R S T R

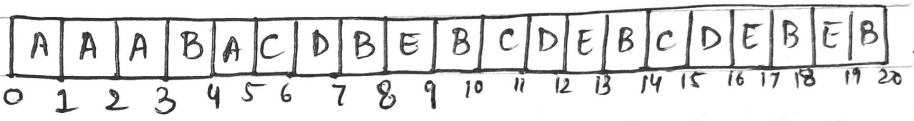
P P S P P R S T R T.

Q)

Process	B.T.	A.T.
A	4	0
B	7	2
C	3	3
D	3	3.5
E	5	4

i) Round Robin (TQ = 1)
ii) Round Robin (TQ = 2)
iii) SJF
Comment on your result, which one is better and why? Also compute the response time of each process.

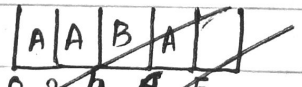
sol) i) Gantt chart



A A A B A C D B E . . . E B
20 21 22

Process	C.T.	R.T.	TAT	WT
A	5	0 - 0 = 0	5	1
B	22	3 - 2 = 1	20	13
C	15	5 - 3 = 2	12	9
D	16	6 - 3.5 = 2.5	12.5	9.5
E	21	8 - 4 = 4	17	12

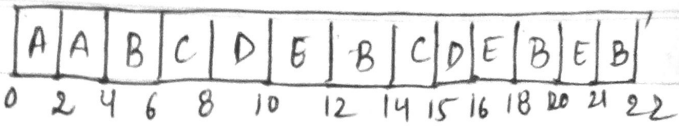
ii) Gantt chart



A A B A C D B E

i) ATAT = 12.9
AWT = 8.5

ii) Gantt chart

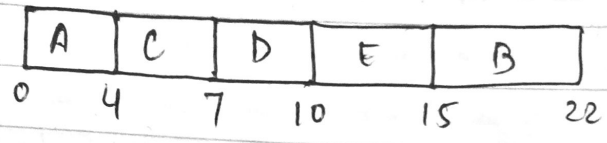


A A B C D E

Process	C.T	R.T.
A	4	0-0=0
B	22	2
C	15	3
D	16	4.5
E	21	6

ATAT = 13.3
AWT = 8.7

iii) Gantt chart



Process	C.T.	R.T.
A	4	0
B	22	13
C	7	1
D	10	2.5
E	15	6

ATAT = 9.1
AWT = 4.7

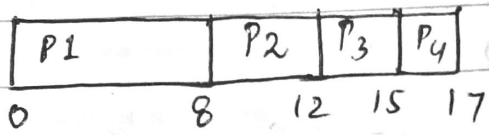
In above ques, SJF is better coz its AWT and ATAT is minimum

→ RR has the drawback of content switching.

Q) Solve by FCFS :—

Process	B.T.	C.T.	TAT	WT.
P1	8	8	8	0
P2	4	12	12	8
P3	3	15	15	12
P4	2	17	17	15

Gantt chart



AWT = 8.75

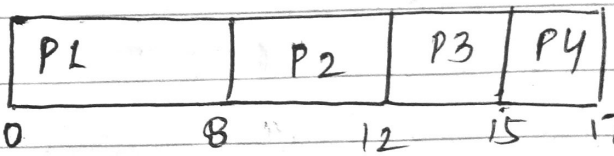
ATAT = 13

Throughput = 0.235

Q) Solve by FCFS :—

Process	BT	AT	CT.	TAT.	W.T.
P1	8	0	8	8	0
P2	4	2	12	10	6
P3	3	6	15	9	6
P4	2	7	17	10	8

Gantt chart :—



AWT = 5

ATAT = 9.25

Throughput = 0.235

* Consider a system with n CPU processors and m processes. Then find the minimum no. of processes and maximum no of processes in the following ~~queue~~ states :—

i) Ready state
ii) Running state
iii) blocked state

Sol)

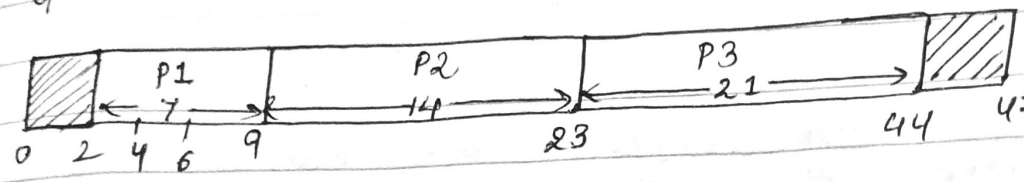
	Min	Max
i) ready	0	m
ii) running	0	n
iii) Blocked	0	m

9) 3 processes arriving at time 0 with total execution time of 10, 20 and 30 units respectively. Each process spends the first 20% of execution time doing I/O. 70% time doing computation and last 10% time doing again I/O. Compute % of CPU ideal time for SRTF.

Sol]

Process	A-T	BT	(20%) I/O	(70%) Computation	(20%) I/O
P1	0	10	2	7	1
P2	0	20	4	14	2
P3	0	30	6	21	3

Gantt chart :—



% Ideal time $= \frac{2}{44} \times 100 = 4.54\ \%$

[∵ Not considering last 3 unit time because no process is being executed after that so we will take till the execution ie - 44 unit in which 2 unit time is ideal in starting only.]

a) No. of process 3 P_1, P_2, P_3 :—

i)

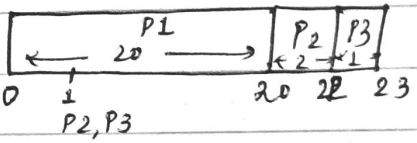
Process	A.T	B.T.
P1	0	20
P2	1	2
P3	1	1

ii)

Process	A.T	B.T.
P1	1	20
P2	0	2
P3	0	1

Using FCFS :—

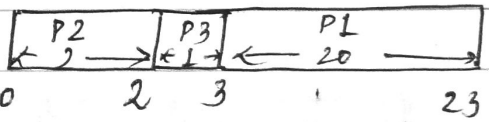
Sol) i) Gantt chart.



Process	TAT	WT	CT
P1	20	0	20
P2	21	19	22
P3	22	21	23

AWT = 13.33, ATAT = 21, TP = 0.130

ii) Gantt chart



Process	TAT	WT	CT
P1	22	2	23
P2	2	0	2
P3	3	2	3

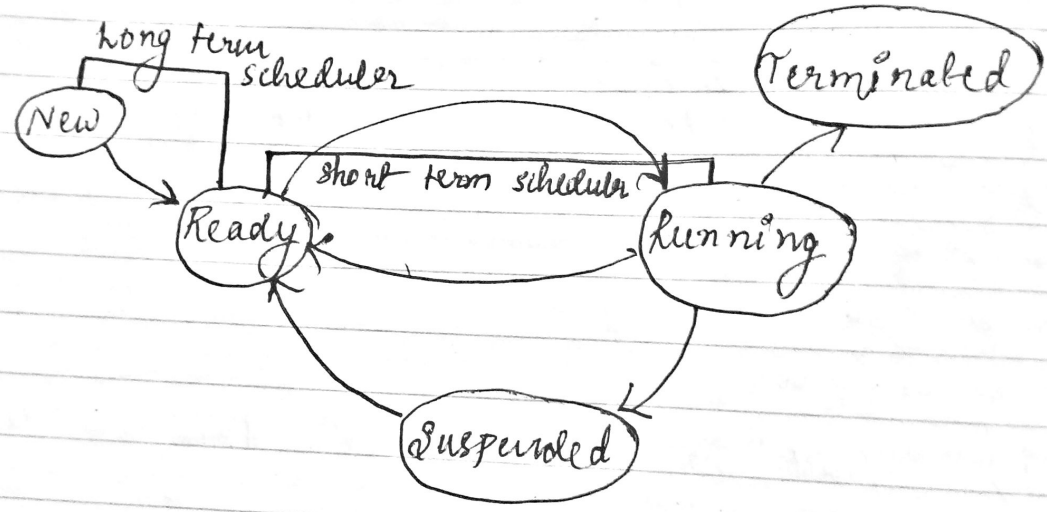
AWT = 1.33, ATAT = 9, TP = 0.130

FCFS suffers from drawback called "convey effect".

If a process with higher BT enters CPU first then it will increases A.W.T. called "convey effect"

* Schedulers

1) Long Term Scheduler
2) Short Term Scheduler



* Longterm Scheduler

- It defines degree of multiprogramming (i.e, no. of process present in memory).
- Value of degree of multiprogramming must be a medium value. because if

its value is high then for new job space has to be free which increases overhead. If its value is less than space vacant in the memory will be more.

- Selects proper mix of I/O process and CPU bound process
- Its frequency is less (i.e. execution frequency)

* Short term scheduler

- Used to load process already exists i.e. in ready state in running state
- Its frequency is high
- Selects which process to execute.

⇒ Process attributes

1 → Process id ⇒ Int no. and system generated

2 → Process state ⇒ running, ready etc

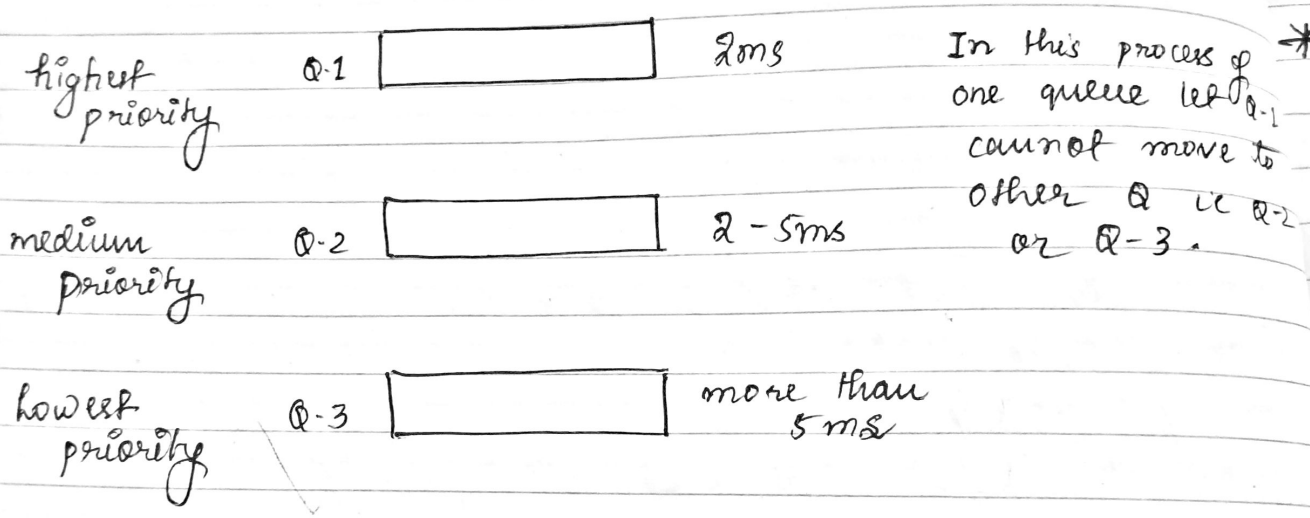
3 → Program counter ⇒ Hold the address of next instruction to execute

4 → Priority

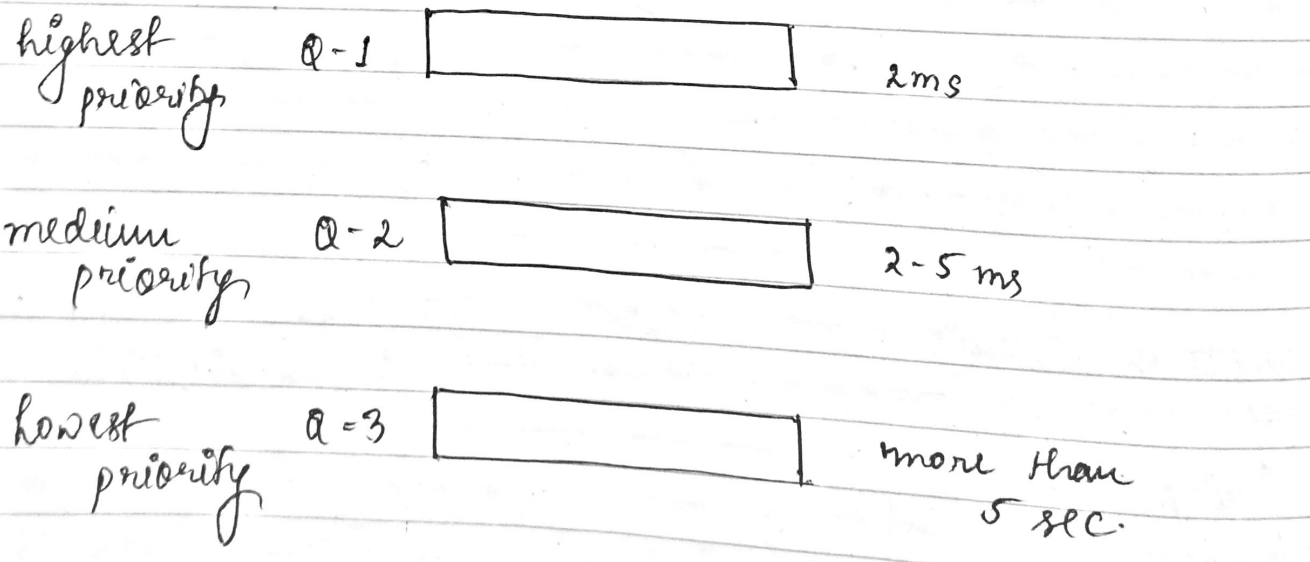
5 → General purpose register

6. List of open files
7. List of devices
8. Protection

* Multilevel Queue :— (Theory)



* Multilevel Feedback Queue scheduling :— (Theory)



All scheduling algos used or studied before Multiprocessor CPU scheduling works on uni-processor OS.

- terminal ⇒ Do not have memory and CPU, just have interface.

In this all the job enters in Q-1 get executed for 2ms if its remaining time is 2ms, then it remains in the same queue else will go to queue Q2 and similarly so on.

* Multi-Processor CPU scheduling :—

Models :—

i) Workstation model
ii) Processor pool model

Workstation model	Processor pool model
• Every sys. has own CPU and memory.	• Every sys. only has terminal connected through server to the one main system which has CPU and memory.

Approach :—

i) Migratory
ii) Non-migratory

Migratory	non-migratory
• We do not decide processor	• we decide which processor will execute the job
• processes can be preempt	• process can not be

DATE: / /

process concurrency → How one job execution on more than one processor

| preempt means cannot be forcefully stopped or access

⇒ process concurrent takes place in Migratory type approach.

* Process concurrency :—

S_1 $a = x + y$
S_2 $b = x + 1$
S_3 $c = a + b$
S_4 $d = c - 1$

Read set ⇒ x, y, z, ...
write set ⇒ a, b ...

$R(S_1) = \{x, y\}$
$W(S_1) = \{a\}$

$R(S_2) = \{x\}$
$W(S_2) = \{b\}$

$R(S_3) = \{a, b\}$
$W(S_3) = \{c\}$

$R(S_4) = \{c\}$
$W(S_4) = \{d\}$

Lab

a) Write a program to list the files and directories of a user defined directory.

```
echo "enter the directory path"
read dirpath → variable
ls $dirpath
```

make it executable then execute it

a) Write a dynamic script which takes month no. and year from the user and display the calander of specified date.

```
echo "enter the month no. and year"
read x _ y        → space
cal $x $y ↵
ctrl+D
$ chmod u+x var.sh ↵
$ ./var.sh
$ enter the month no. and year
  10 _ 2020       → space ↵
```

a) Add Two no.

```
echo "enter no.s"
read x y
expr _$x_+_$y_   → denotes space
```

Q)
read x y
echo "The sum of $x and $y is 'expr $x + $y'"

Q) Make two seperate files :—

1st file
mkdir filename

2nd file
rmdir filename

* Chron Job

$ Crontab -l → current user cron jobs
~~$ Crontab -u username -l~~
$ Crontab -l -u username

req. root permiss.

↓ tells cronjob of named user

either use sudo or go in root (i.e. #)

$ crontab -r → delete cron job of current user
$ crontab -r -u username

↓ delete crontab for specific user

$ crontab -e → edit the cron jobs of current user

$

Creation of cron job is divided into two parts :-

i) setting the time interval
ii) which command or file will execute.

min, hour, day, month, week day
0-59, 0-23, 1-31, 1-12, 0-6
~~Sat~~ ~~Friday~~ ~~Sunday~~
Sunday (T) Saturday

open .sh file

* * * * *
↓
every minutes

2 * * * *
↓
whenever the min will be 2.
ie. 1:02, 2:02 etc.

*/5 * * * *
↓
when the no. b/w 0-59 is completely divisible by 5.

* 0 * * *
↓
at 12 mid-night

* * 15 * *
↓
ever 15th date
ever min, every month 15th . etc.

if system is busy at 12 midnight it can be 12:05 etc.

* In the form of string

@hourly	when hour changes
@daily	when date change
@monthly	when month change
@weekly	when week changes
@annually	
@yearly	
@midnight	at midnight
@reboot	

(can vary)

> , used for redirecting the output
/dev/null , black hole
2 > &1 , show the error on terminal

Creating crontab :—

→ sudo
$ nano /etc/crontab ↵

go down

varun28 → space

* * * * * /home/varun28/v1.sh

ctrl + x ⇒ y ⇒ Enter

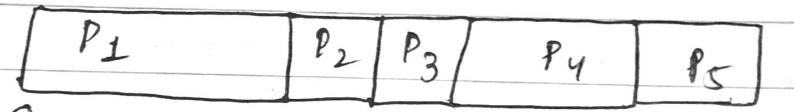
* FCFS

Q) No. of process is 5.

Process	BT	AT	CT	TAT	WT
P1	10	0	10	10	0
P2	1	0	11	11	10
P3	2	15	13	-2	-4
P4	5	16	18	2	-3
P5	3	17	21	4	1

Calculate WT, TAT, CT, AWT, ATAT and draw gantt chart.

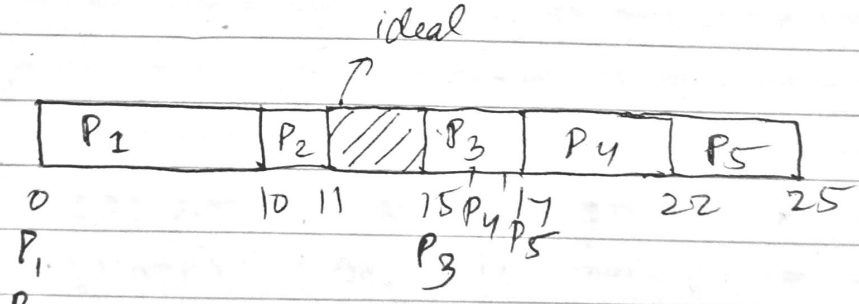
Gantt chart



AWT =

ATAT =

Gantt chart



	CT	TAT	WT
P_1	10	10	0
P_2	11	11	10
P_3	17	2	0
P_4	22	6	1
P5	25	8	5

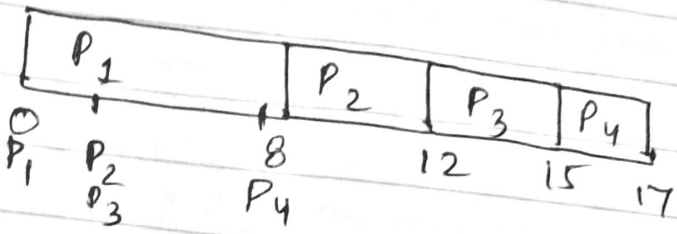
Q)

Process	BT	AT	WT	TAT	CT
P_1	8	0	0	8	8
P_2	4	2	6	10	12
P_3	3	2	10	13	15
P_4	2	7	8	10	17

TAT - BT

CT - AT

Gantt chart



AWT = 6

ATAT = 10.25

Throughput = 0.235

---

s1.sh

```
mkdir A
```

s2.sh

```
rmdir A
```

```
chmod u+x s1.sh
chmod u+x s2.sh
sudo nano /etc/crontab
```

⇒ now do as written on previous page

* * * * * muited /home/muited/s1.sh

* * * * * muited sleep 30 ; /home/muited/s2.sh

↓ in seconds

folder name

S3.h

mkdir "$(date +'%d-%m-%y-%r')"

create a folder every minute using crontab showing date, month, year and time,

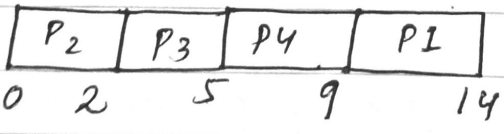
used to execute date command with %d, %m, %y and %r parameters showing in it.

Crontab :— It is a type of command or dmon which means it works in background and output is also shown in the background.

* SJF

Process	BT	CT	WT	TAT
P1	5	14	9	14
P2	2	2	0	2
P3	3	5	2	5
P4	4	9	5	9

Gantt chart



AWT = 4

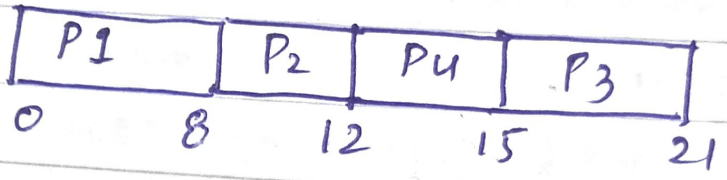
ATAT = 7.5

Throughput = 0.285

Q)

Process	AT	BT	CT	TAT	WT
P1	0	8	8	8	0
P2	8	4	12	4	0
P3	10	6	21	11	5
P4	12	3	15	3	0

Gantt chart



AWT = 1.25

ATAT = 6.5

TP = 0.190