CDAC MUMBAI

Concepts of Operating System

Assignment 2

Samir Bharati (PG-DAC)

Part A

echo "Hello, World!" -> print “Hello World” in terminal

name="Productive" -> it will create a variable as ‘name’ stores the value ‘Productions’

touch file.txt -> it will create a file called “file.txt”

ls -a -> it will list all files including the hidden files

rm file.txt -> it will remove/delete the file.txt

cp file1.txt file2.txt -> it will copy the content of file1.txt to new file called file2.txt

mv file.txt /path/to/directory/ -> it will move file.txt to the given or specific directory

chmod 755 script.sh -> it will give the permission to the script.sh file as 7(owner : rwx), 5(group : r-x) 5(other user : r-x)

grep "pattern" file.txt -> it will search a specific word “pattern” in the file.txt

kill PID -> it will terminates the process with the given PID

mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt
-> it will make a directory named “mydir” and then change the current directory to mydir then create a file in this directory as ‘file.txt’ and print “Hello, World!” in file.txt and then print the content of file.txt

ls -l | grep ".txt" -> it will list files in long format, then filters only with .txt name

grep -r "pattern" /path/to/directory/ -> it search “pattern” word in reverse order throughout the file

cat file1.txt file2.txt | sort | uniq –d -> it will show only duplicate line in both files

chmod 644 file.txt -> it will give the permission to the file.txt file as 6(owner : rw-), 4(group : r- -) 4(other user : r- -)

cp -r source_directory destination_directory -> it will copy a folder and all its contents in reverse order (files + subdirectories) into the destination

find /path/to/search -name "*.txt" -> it will search inside its subdirectories for all files whose names end with .txt

chmod u+x file.txt -> it will give permission to the other user for the file.txt

echo $PATH -> it will display the list of directories the shell searches in when type a command

Part B

Identify True or False:

1. **ls is used to list files and directories in a directory.** -> TRUE

2. **mv is used to move files and directories.** -> TRUE

3. **cd is used to copy files and directories.** -> FALSE

4. **pwd stands for "print working directory" and displays the current directory.** -> TRUE

5. **grep is used to search for patterns in files.** -> TRUE

6. **chmod 755 file.txt gives read, write, and execute permissions to the owner, and read and execute permissions to group and others.** -> TRUE

7. **mkdir -p directory1/directory2 creates nested directories, creating directory2 inside directory1 if directory1 does not exist.** -> TRUE

8. **rm -rf file.txt deletes a file forcefully without confirmation.** -> TRUE

Identify the Incorrect Commands:

1. chmodx is used to change file permissions. -> chmod

2. cpy is used to copy files and directories. -> cp

3. mkfile is used to create a new file. -> touch

4. catx is used to concatenate files. -> cat

5. rn is used to rename files. -> mv

Part C

Question 1: Write a shell script that prints "Hello, World!" to the terminal.
Ans:
```
echo "Hello, World!"
```

Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.
Ans:
```
name="CDAC Mumbai"
echo "$name"
```

Question 3: Write a shell script that takes a number as input from the user and prints it.
Ans:
```
echo "Enter a number:"
read num
echo "You entered: $num"
```

Question 4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result.
Ans:
```
num1=5
num2=3
sum=$((num1 + num2))
echo "The sum of $num1 and $num2 is $sum"
```

Question 5: Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd".

Ans:
```
echo "Enter a number:"
read num

if (( num % 2 == 0 )); then
    echo "Even"
else
    echo "Odd"
fi
```

Question 6: Write a shell script that uses a for loop to print numbers from 1 to 5.

Ans:
```
for i in {1..5}
do
    echo $i
done
```

Question 7: Write a shell script that uses a while loop to print numbers from 1 to 5.

Ans:
```
i=1
while [ $i -le 5 ]
 do
      echo $i
       ((i++))
done
```

Question 8: Write a shell script that checks if a file named "file.txt" exists in the current directory. If it does, print "File exists", otherwise, print "File does not exist".

Ans:
```
if [ -f "file.txt" ]; then
   echo "File exists"
else
   echo "File does not exist"
fi
```

Question 9: Write a shell script that uses the if statement to check if a number is greater than 10 and prints a message accordingly.

Ans:
```
echo "Enter a number:"
read num

if [ $num -gt 10 ]; then
```

```
        echo "$num is greater than 10"
    else
       echo "$num is not greater than 10"
    fi
```

Question 10: Write a shell script that uses nested for loops to print a multiplication table for numbers from 1 to 5. The output should be formatted nicely, with each row representing a number and each column representing the multiplication result for that number.

Ans:

```
     for i in {1..5}
     do
         for j in {1..5}
         do
            printf  "%4d"  $((i * j))
         done
         echo
    done
```

Question 11: Write a shell script that uses a while loop to read numbers from the user until the user enters a negative number. For each positive number entered, print its square. Use the break statement to exit the loop when a negative number is entered.

Ans:

```
    while true
    do
        echo "Enter a number: "
        read num

        if [ $num -lt 0 ]; then
            echo "Enter a positive number"
            break
        fi

        echo "square: $((num * num))"
   done
```

Part D

Common Interview Questions (Must know)

1. What is an operating system, and what are its primary functions?
Ans: Operating system is a program that manages the computer hardware and it also provides the basics for application programs and acts as an intermediary between user and computer hardware. Its primary function is to manage computer hardware and software resources, provide a user interface and execute and manage applications.

2. Explain the difference between process and thread.
Ans: A process is a single application or program whereas a thread is a subprocess within that application program. Each process has its own address space in memory whereas threads share their address space.

3. What is virtual memory, and how does it work?
Ans: So it is a memory management technique that creates an illusion of having more RAM than physically available by using a combination of physical RAM and space on a hard drive (SSD). It allows programs to run as if they have a larger memory space than the actual RAM, even if they don't.

4. Describe the difference between multiprogramming, multitasking, and multiprocessing.
Ans: Multiprogramming involves loading multiple programs into the main memory concurrently, allowing the CPU to switch between them when one program is performing I/O operations or waiting for resources. The goal is to maximize CPU utilization by minimizing ideal time.

Multitasking is just an extension of multiprogramming where the CPU rapidly switches between multiple processes or tasks giving the illusion of simultaneous execution. This is achieved through time-slicing where each task is allocated a small time quantum of CPU time.

Multi processing involves the use of multiple CPU or processing units within a single computer system to execute multiple processes or tasks in parallel.

5. What is a file system, and what are its components?
Ans: It is the method of an operating system used to store, organise and manage data on a storage device providing hierarchical structure for files and directories.
Components are : File Allocation, Directory structure, Metadata, Security and Permission

6. What is a deadlock, and how can it be prevented?
Ans: A deadlock is a situation where two or more processes wait indefinitely for resources held by each other, causing no progress.
It occurs when the four Coffman conditions (Mutual Exclusion, Hold & Wait, No Preemption, Circular Wait) are true. Deadlock can be prevented by breaking these conditions, e.g., avoiding hold & wait or enforcing resource ordering.

7. Explain the difference between a kernel and a shell.
Ans: The kernel is the core component of an operating system which acts as the bridge between the hardware and the software application, it is responsible for managing the system resources such as CPU, memory and I/O devices and handling process scheduling and providing them calls that allows applications to interact with the hardware.

The shell is an interface that allows users to interact with the operating system, it acts as an interpreter, translating user commands into system calls that the kernel can understand and execute.

8. What is CPU scheduling, and why is it important?
Ans: It is the process by which an Operating System decides which of the ready processes gets to use the CPU at any given time.
It is important for maximizing CPU utilization, improving system responsiveness, Enable Multasking, Fairness and resource management.

9. How does a system call work?
Ans: If we have to use any functionality of the operating system then we have to go to the user more to kernel mode then system calls help us to do so.
E.g: open(), Read(), write(), close, Createfile, etc.

10. What is the purpose of device drivers in an operating system?
Ans: Device drivers act as translators between the operating system and hardware devices. They allow the OS to send standard commands, which the driver converts into device-specific instructions. Without drivers, the OS cannot communicate with hardware like printers, disks, or network cards.

11. Explain the role of the page table in virtual memory management.
Ans: A page table maps logical pages of a process to physical frames in memory.
It enables the CPU to translate virtual addresses into physical addresses efficiently.
This allows processes to run even if their pages are scattered in RAM.

12. What is thrashing, and how can it be avoided?
Ans: Thrashing occurs when the CPU spends more time swapping pages than executing processes. It happens due to excessive page faults when too many processes compete for

limited memory. It can be avoided by using working set models, increasing RAM, or limiting multiprogramming.

13. Describe the concept of a semaphore and its use in synchronization.
Ans: A semaphore is a synchronization tool represented by an integer variable.
It uses two atomic operations, wait(P) and signal(V), to control resource access.
Semaphores prevent race conditions by ensuring mutual exclusion in critical sections.

14. How does an operating system handle process synchronization?
Ans: The OS uses mechanisms like semaphores, mutexes, and monitors to coordinate processes.These tools ensure only one process enters a critical section at a time.
Thus, it prevents race conditions and maintains data consistency.

15. What is the purpose of an interrupt in operating systems?
Ans: The purpose of interrupt in OS is to allow hardware and software to signal the CPU to pause its current task and handle a specific event or request.

16. Explain the concept of a file descriptor.
Ans: A file descriptor is a non-negative integer that a process uses to uniquely identify and interact with open files or I/O resources in an operating system.

17. How does a system recover from a system crash?
Ans: The OS recovers using backups, checkpoints, and system logs.
It restores the system state and may restart affected processes.
File systems also use journaling to prevent data corruption.

18. Describe the difference between a monolithic kernel and a microkernel.
Ans: A monolithic kernel has all OS services (CPU, memory, I/O, device drivers) in one large kernel space. A microkernel keeps only core services (CPU scheduling, IPC, memory management) in kernel space.
Other services like drivers run in user space, improving modularity and reliability.

19. What is the difference between internal and external fragmentation?
Ans: Internal fragmentation is wasted space inside allocated memory blocks.
External fragmentation occurs when free memory is split into small scattered holes.
Both reduce memory utilization, but causes differ (fixed vs dynamic allocation).

20. How does an operating system manage I/O operations?
Ans: The OS uses device drivers and interrupt handling to communicate with hardware.
It employs techniques like buffering, caching, and spooling for efficiency.
The I/O subsystem ensures smooth data transfer between CPU, memory, and devices.

21. Explain the difference between preemptive and non-preemptive scheduling.
Ans: In preemptive scheduling, the OS can forcibly take the CPU from a process (e.g., Round Robin, Priority).
In non-preemptive scheduling, a process keeps the CPU until completion or blocking (e.g., FCFS, SJF).
Preemptive gives better responsiveness, while non-preemptive is simpler but less flexible.

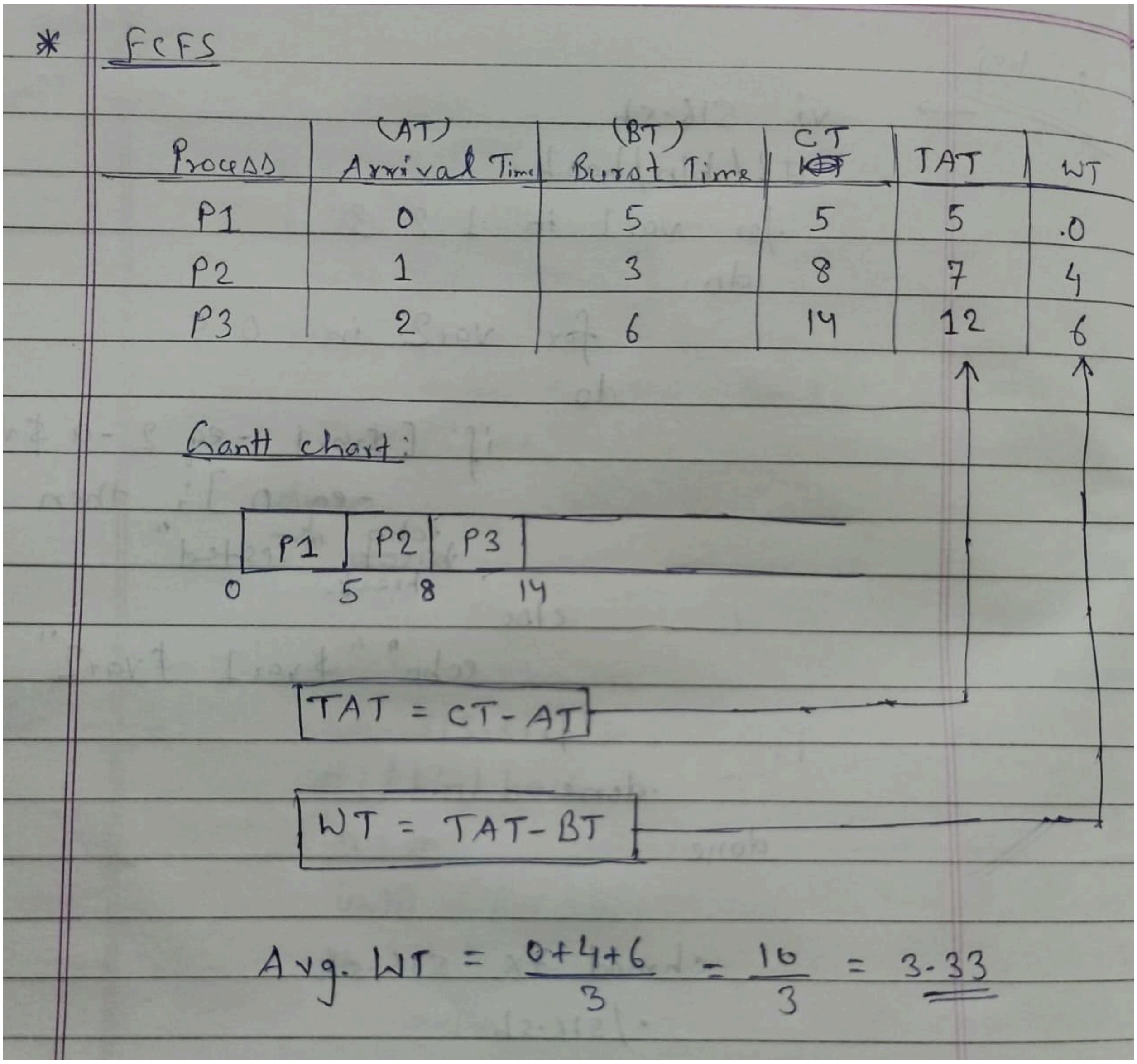
Part E

1. Consider the following processes with arrival times and burst times:

Process	Arrival Time	Burst Time
P1	0	5
P2	1	3
P3	2	6

Calculate the average waiting time using First-Come, First-Served (FCFS) scheduling.

Ans:



* FCFS

Process	(AT) Arrival Time	(BT) Burst Time	CT	TAT	WT
P1	0	5	5	5	.0
P2	1	3	8	7	4
P3	2	6	14	12	6

Gantt chart:

P1	P2	P3

0 5 8 14

TAT = CT - AT

WT = TAT - BT

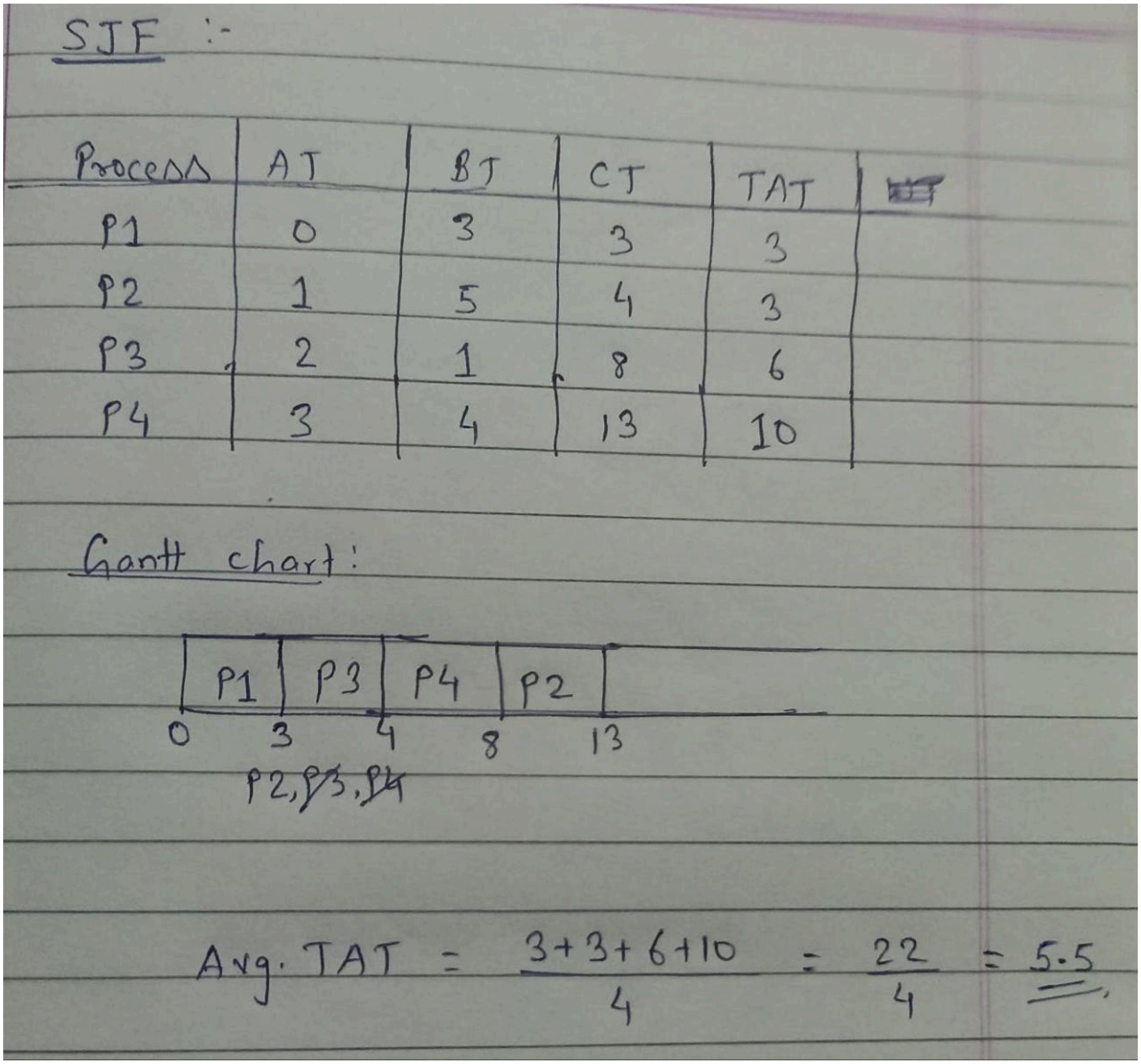
$$\text{Avg. WT} = \frac{0+4+6}{3} = \frac{10}{3} = \underline{\underline{3.33}}$$

2. Consider the following processes with arrival times and burst times:

Process	Arrival Time	Burst Time
P1	0	3
P2	1	5
P3	2	1
P4	3	4

Calculate the average turnaround time using Shortest Job First (SJF) scheduling.

Ans:



SJF :-

Process	AT	BT	CT	TAT	~~WT~~
P1	0	3	3	3	
P2	1	5	4	3	
P3	2	1	8	6	
P4	3	4	13	10	

Gantt chart:

P1	P3	P4	P2

0 3 4 8 13

P2, ~~P3~~, ~~P4~~

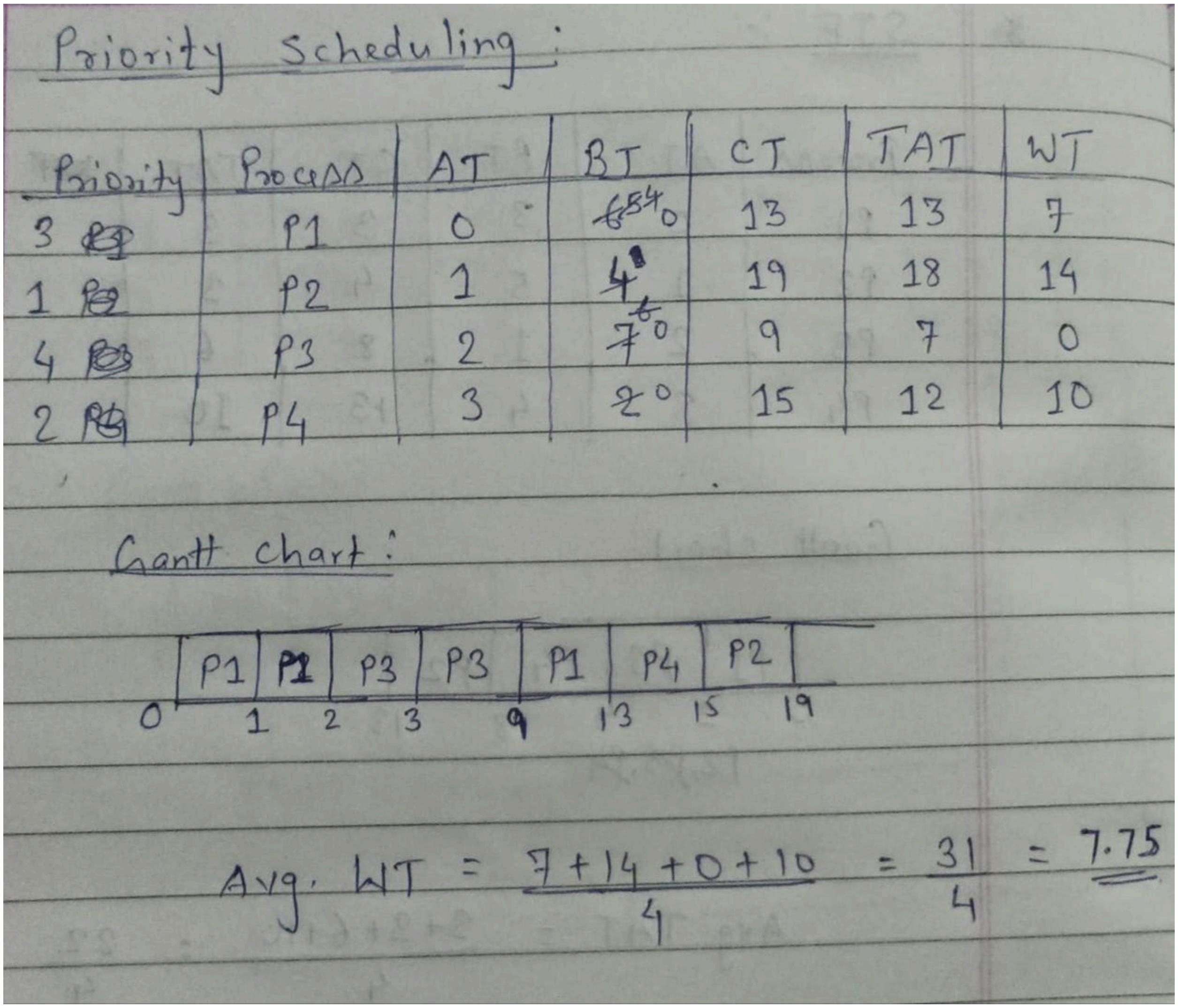
$$\text{Avg. TAT} = \frac{3+3+6+10}{4} = \frac{22}{4} = \underline{5.5}$$

3. Consider the following processes with arrival times, burst times, and priorities (lower number indicates higher priority):

Process	Arrival Time	Burst Time	Priority
P1	0	6	3
P2	1	4	1
P3	2	7	4
P4	3	2	2

Calculate the average waiting time using Priority Scheduling.

Ans:



Priority Scheduling :

Priority	Process	AT	BT	CT	TAT	WT
3 ~~P1~~	P1	0	~~6~~ 5 4 0	13	13	7
1 ~~P2~~	P2	1	~~4~~	19	18	14
4 ~~P3~~	P3	2	~~7~~ 6 0	9	7	0
2 ~~P4~~	P4	3	~~2~~ 0	15	12	10

Gantt chart :

P1	P1	P3	P3	P1	P4	P2

0 1 2 3 9 13 15 19

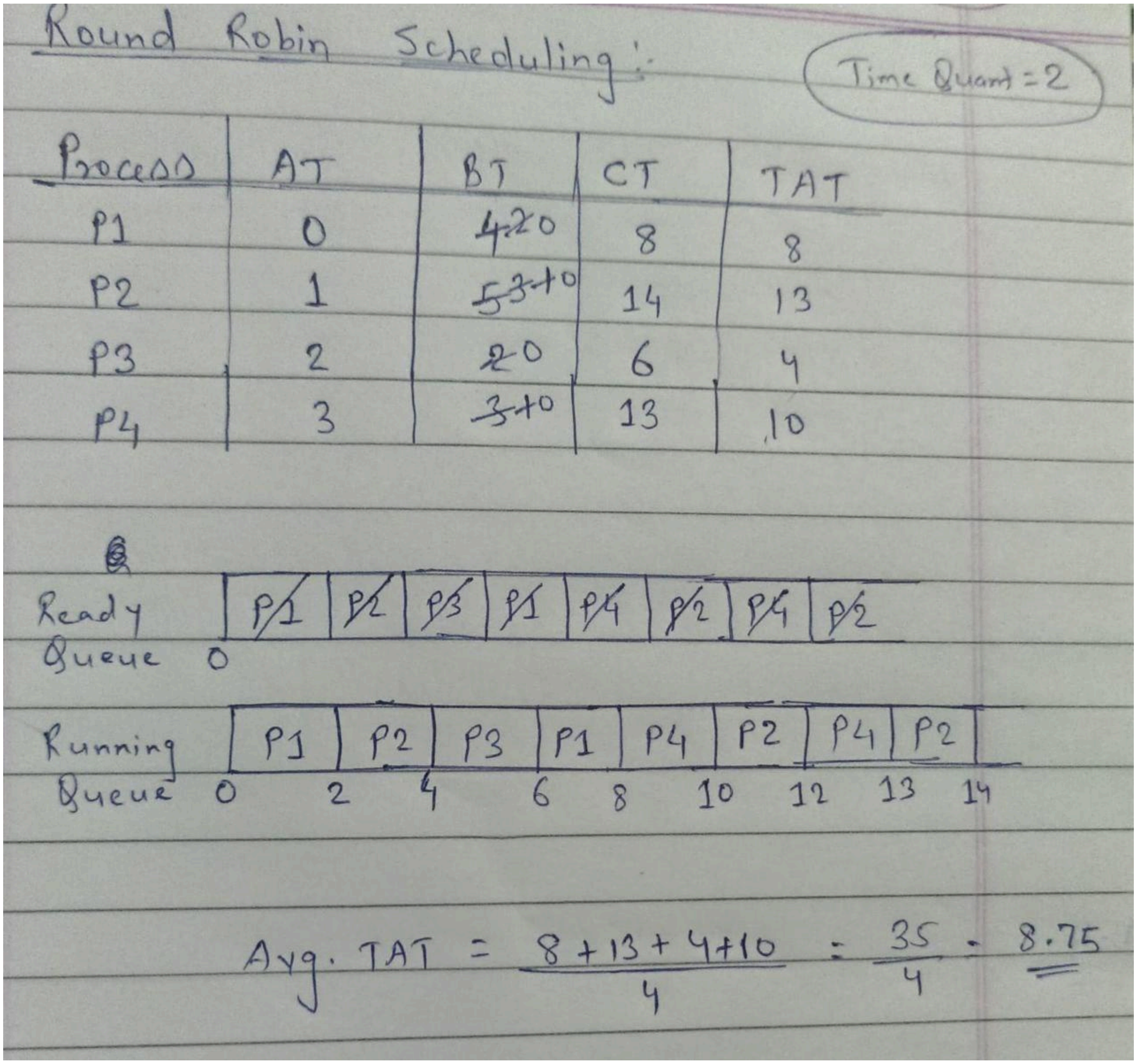
$$\text{Avg. WT} = \frac{7+14+0+10}{4} = \frac{31}{4} = \underline{\underline{7.75}}$$

4. Consider the following processes with arrival times and burst times, and the time quantum for Round Robin scheduling is 2 units:

Process	Arrival Time	Burst Time
P1	0	4
P2	1	5
P3	2	2
P4	3	3

Calculate the average turnaround time using Round Robin scheduling.

Ans:



Round Robin Scheduling :

Time Quant = 2

Process	AT	BT	CT	TAT
P1	0	~~4~~ ~~2~~ 0	8	8
P2	1	~~5~~ ~~3~~ ~~1~~ 0	14	13
P3	2	~~2~~ 0	6	4
P4	3	~~3~~ ~~1~~ 0	13	10

Ready Queue

~~P1~~	~~P2~~	~~P3~~	~~P1~~	~~P4~~	~~P2~~	~~P4~~	~~P2~~

0

Running Queue

P1	P2	P3	P1	P4	P2	P4	P2

0 2 4 6 8 10 12 13 14

Avg. TAT = $\frac{8 + 13 + 4 + 10}{4} = \frac{35}{4} = \underline{\underline{8.75}}$