**ASSIGNMENT 1**

**Name : Meghana Dhananjay Pagare**

**Roll No : 1135**

**Sub : concepts of programming with OS**

1. In your home directory, create sets of empty practice files

Create 6 files with names of the forms on songsX.mp3.

Create 6 files with names of the form snapX.jpg.

Create 6 files with names of the form filmX.avi.

In each set, replace X with the numbers 1 through 6.

[user1@localhost ~]$ touch songs{1..6}.mp3

[user1@localhost ~]$ ls

Desktop family Music snap1.jpg songs2.mp3 songs5.mp3 Videos

Documents film1.avi Pictures songs1 songs3.mp3 songs6.mp3 work

Downloads friends Public songs1.mp3 songs4.mp3 Templates

[user1@localhost ~]$ touch snap{1..6}.jpg

[user1@localhost ~]$ ls

Desktop film1.avi Public snap4.jpg songs1.mp3 songs5.mp3 work

Documents friends snap1.jpg snap5.jpg songs2.mp3 songs6.mp3

Downloads Music snap2.jpg snap6.jpg songs3.mp3 Templates

family Pictures snap3.jpg songs1 songs4.mp3 Videos

[user1@localhost ~]$ touch film{1..6}.avi

[user1@localhost ~]$ ls

Desktop film2.avi friends snap2.jpg songs1 songs5.mp3

Documents film3.avi Music snap3.jpg songs1.mp3 songs6.mp3

Downloads film4.avi Pictures snap4.jpg songs2.mp3 Templates

family film5.avi Public snap5.jpg songs3.mp3 Videos

film1.avi film6.avi snap1.jpg snap6.jpg songs4.mp3 work

[user1@localhost ~]$

-----------------------------------------------------------------------------------------------------------------

2. From your home directory,

Move songs file into your Music subdirectory.

Move snap file into your Pictures subdirectory.

Move your movie files into Videos subdirectory

[user1@localhost ~]$ mv songs{1..6}.mp3 Music

[user1@localhost ~]$ ls

Desktop film1.avi film5.avi Pictures snap3.jpg songs1

Documents film2.avi film6.avi Public snap4.jpg Templates

Downloads film3.avi friends snap1.jpg snap5.jpg Videos

family film4.avi Music snap2.jpg snap6.jpg work

[user1@localhost ~]$ mv snap{1..6}.jpg Pictures

[user1@localhost ~]$ ls

Desktop family film3.avi film6.avi Pictures Templates

Documents film1.avi film4.avi friends Public Videos

Downloads film2.avi film5.avi Music songs1 work

[user1@localhost ~]$ mv film{1..6}.avi Videos

[user1@localhost ~]$ ls

Desktop Downloads friends Pictures songs1 Videos

Documents family Music Public Templates work

[user1@localhost ~]$

-------------------------------------------------------------------------------------------------------------

3. Create 3 subdirectories for organizing your files named friends,family,work

[user1@localhost ~]$ mkdir friends family work

[user1@localhost ~]$

---------------------------------------------------------------------------------------------------------------

4. Copy files (all types ) containing numbers 1 and 2 to the friends folder.

Copy files (all types) containing numbers 3 and 4 to the family folder.

Copy files (all types) containing numbers 5 and 6 to the work folder.

[user1@localhost ~]$ cp songs{1..2}.\* friends

[user1@localhost ~]$ cd friends

[user1@localhost friends]$ ls

songs1.mp3 songs2.mp3

[user1@localhost Videos]$ cd

[user1@localhost ~]$ ls

Desktop film2.avi friends snap2.jpg songs1 songs5.mp3

Documents film3.avi Music snap3.jpg songs1.mp3 songs6.mp3

Downloads film4.avi Pictures snap4.jpg songs2.mp3 Templates

family film5.avi Public snap5.jpg songs3.mp3 Videos

film1.avi film6.avi snap1.jpg snap6.jpg songs4.mp3 work

[user1@localhost ~]$ cp snap{1..2}.\* friends

[user1@localhost ~]$ cd friends

[user1@localhost friends]$ ls

snap1.jpg snap2.jpg songs1.mp3 songs2.mp3

[user1@localhost ~]$ cp film{1..2}.\* friends

[user1@localhost ~]$ ls

Desktop film2.avi friends snap2.jpg songs1 songs5.mp3

Documents film3.avi Music snap3.jpg songs1.mp3 songs6.mp3

Downloads film4.avi Pictures snap4.jpg songs2.mp3 Templates

family film5.avi Public snap5.jpg songs3.mp3 Videos

film1.avi film6.avi snap1.jpg snap6.jpg songs4.mp3 work

[user1@localhost ~]$ cd friends

[user1@localhost friends]$ ls

film1.avi film2.avi snap1.jpg snap2.jpg songs1.mp3 songs2.mp3

[user1@localhost ~]$ cp songs{3..4}.\* family

[user1@localhost ~]$ ls

Desktop film2.avi friends snap2.jpg songs1 songs5.mp3

Documents film3.avi Music snap3.jpg songs1.mp3 songs6.mp3

Downloads film4.avi Pictures snap4.jpg songs2.mp3 Templates

family film5.avi Public snap5.jpg songs3.mp3 Videos

film1.avi film6.avi snap1.jpg snap6.jpg songs4.mp3 work

[user1@localhost ~]$ cd family

[user1@localhost family]$ ls

songs3.mp3 songs4.mp3

[user1@localhost family]$ cd

[user1@localhost ~]$ ls

Desktop film2.avi friends snap2.jpg songs1 songs5.mp3

Documents film3.avi Music snap3.jpg songs1.mp3 songs6.mp3

Downloads film4.avi Pictures snap4.jpg songs2.mp3 Templates

family film5.avi Public snap5.jpg songs3.mp3 Videos

film1.avi film6.avi snap1.jpg snap6.jpg songs4.mp3 work

[user1@localhost ~]$ cp snap{3..4}.\* family

[user1@localhost ~]$ cd family

[user1@localhost family]$ ls

snap3.jpg snap4.jpg songs3.mp3 songs4.mp3

[user1@localhost family]$ cd

[user1@localhost ~]$ ls

Desktop film2.avi friends snap2.jpg songs1 songs5.mp3

Documents film3.avi Music snap3.jpg songs1.mp3 songs6.mp3

Downloads film4.avi Pictures snap4.jpg songs2.mp3 Templates

family film5.avi Public snap5.jpg songs3.mp3 Videos

film1.avi film6.avi snap1.jpg snap6.jpg songs4.mp3 work

[user1@localhost ~]$ cp film{3..4}.\* family

[user1@localhost ~]$ cd family

[user1@localhost family]$ ls

film3.avi film4.avi snap3.jpg snap4.jpg songs3.mp3 songs4.mp3

[user1@localhost ~]$ cp songs{5..6}.\* work

[user1@localhost ~]$ cd work

[user1@localhost work]$ ls

songs5.mp3 songs6.mp3

[user1@localhost work]$

[user1@localhost ~]$ cp snap{5..6}.\* work

[user1@localhost ~]$ cd work

[user1@localhost work]$ ls

snap5.jpg snap6.jpg songs5.mp3 songs6.mp3

[user1@localhost work]$

[user1@localhost ~]$ cp film{5..6}.\* work

[user1@localhost ~]$ cd work

[user1@localhost work]$ ls

film5.avi film6.avi snap5.jpg snap6.jpg songs5.mp3 songs6.mp3

[user1@localhost work]$

-----------------------------------------------------------------------------------------------------------

6. Delete friends subdirectory

[user1@localhost ~]$ rm -r friends

[user1@localhost ~]$ cd friends

bash: cd: friends: No such file or directory

----------------------------------------------------------------------------------------------------------------------

7. rename all snap files in work directory to new name photoX.jpg

X is a number

[user1@localhost ~]$ cd

[user1@localhost ~]$ ls

Desktop film2.avi Music snap3.jpg songs1.mp3 songs6.mp3

Documents film3.avi Pictures snap4.jpg songs2.mp3 Templates

Downloads film4.avi Public snap5.jpg songs3.mp3 Videos

family film5.avi snap1.jpg snap6.jpg songs4.mp3 work

film1.avi film6.avi snap2.jpg songs1 songs5.mp3

[user1@localhost ~]$ mv snap1.jpg photo1.jpg

[user1@localhost ~]$ mv snap2.jpg photo2.jpg

[user1@localhost ~]$ mv snap3.jpg photo3.jpg

[user1@localhost ~]$ mv snap4.jpg photo4.jpg

[user1@localhost ~]$ mv snap5.jpg photo5.jpg

[user1@localhost ~]$ mv snap6.jpg photo6.jpg

[user1@localhost ~]$ ls

Desktop film2.avi Music photo5.jpg songs1.mp3 songs6.mp3

Documents film3.avi photo1.jpg photo6.jpg songs2.mp3 Templates

Downloads film4.avi photo2.jpg Pictures songs3.mp3 Videos

family film5.avi photo3.jpg Public songs4.mp3 work

film1.avi film6.avi photo4.jpg songs1 songs5.mp3

[user1@localhost ~]$

----------------------------------------------------------------------------------------------------------------

**ASSIGNMENT 2**

**Name : Meghana Dhananjay Pagare**

**Roll No : 1135**

**Sub : concepts of programming with OS**

1. Create user tom , bob , sam , prince

[user1@localhost ~]$ pwd

/home/user1

[user1@localhost ~]$ su -

Password:

[root@localhost ~]# useradd tom

[root@localhost ~]# useradd bob

[root@localhost ~]# useradd sam

[root@localhost ~]# useradd prince

[root@localhost ~]# tail /etc/passwd

sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin

avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin

postfix:x:89:89::/var/spool/postfix:/sbin/nologin

ntp:x:38:38::/etc/ntp:/sbin/nologin

tcpdump:x:72:72::/:/sbin/nologin

user1:x:1000:1000:user1:/home/user1:/bin/bash

tom:x:1001:1001::/home/tom:/bin/bash

bob:x:1002:1002::/home/bob:/bin/bash

sam:x:1003:1003::/home/sam:/bin/bash

prince:x:1004:1004::/home/prince:/bin/bash

--------------------------------------------------------------------------------------------------------------------

1. Create Group dac , dbda ,ditiss

[root@localhost ~]# sudo groupadd dac

[root@localhost ~]# sudo groupadd dbda

[root@localhost ~]# sudo groupadd ditiss

[root@localhost ~]# tail /etc/group

ntp:x:38:

tcpdump:x:72:

user1:x:1000:user1

tom:x:1001:

bob:x:1002:

sam:x:1003:

prince:x:1004:

dac:x:1005:

dbda:x:1006:

ditiss:x:1007:

----------------------------------------------------------------------------------------------------------------------

3. add user

Tom in dac

Bob in dbda

Sam in ditiss

[root@localhost ~]# sudo usermod -aG dac tom

[root@localhost ~]# sudo usermod -aG dbda bob

[root@localhost ~]# sudo usermod -aG ditiss sam

[root@localhost ~]# tail /etc/group

ntp:x:38:

tcpdump:x:72:

user1:x:1000:user1

tom:x:1001:

bob:x:1002:

sam:x:1003:

prince:x:1004:

dac:x:1005:tom

dbda:x:1006:bob

ditiss:x:1007:sam

-----------------------------------------------------------------------------------------------------------------

4. login as prince and create iacsd directory in /tmp and

create 4 files in iacsd with name project-1 project-2 upto 4

[root@localhost ~]# nano /etc/sudoers

[root@localhost ~]# tail -n 3 /etc/passwd

bob:x:1002:1002::/home/bob:/bin/bash

sam:x:1003:1003::/home/sam:/bin/bash

prince:x:1004:1004::/home/prince:/bin/bash

[root@localhost ~]# mkdir iacsd

[root@localhost ~]# cd iacsd

[root@localhost iacsd]# touch project1

[root@localhost iacsd]# touch project2

[root@localhost iacsd]# touch project3

[root@localhost iacsd]# touch project4

[root@localhost iacsd]# ls

prince project1 project2 project3 project4

-----------------------------------------------------------------------------------------------------------

5. assign permissions to project files as below

Project-1 – tom should be owner of this

Project-2 – dac should be owner of this

Project-3 --- others should not have any permission but tom should have rw access

Project-4 – dbda group should have rwx permissions.

[root@localhost iacsd]# sudo chown tom project1

[root@localhost ~]# cd

[root@localhost ~]# cd iacsd

[root@localhost iacsd]# sudo chgrp dac project2

[root@localhost iacsd]# sudo chmod 600 project3

[root@localhost iacsd]# ls -l

total 0

drwxr-xr-x. 2 root root 6 May 13 12:10 prince

-rw-r--r--. 1 tom root 0 May 13 12:16 project1

-rw-r--r--. 1 root dac 0 May 13 12:16 project2

-rw-------. 1 root root 0 May 13 12:16 project3

-rw-r--r--. 1 root root 0 May 13 12:16 project4

[root@localhost iacsd]# sudo chown tom project3

[root@localhost iacsd]# ls -l

total 0

drwxr-xr-x. 2 root root 6 May 13 12:10 prince

-rw-r--r--. 1 tom root 0 May 13 12:16 project1

-rw-r--r--. 1 root dac 0 May 13 12:16 project2

-rw-------. 1 tom root 0 May 13 12:16 project3

-rw-r--r--. 1 root root 0 May 13 12:16 project4

[root@localhost iacsd]# sudo chgrp dbda project4

[root@localhost iacsd]# sudo chmod 670 project4

[root@localhost iacsd]# ls -l

total 0

drwxr-xr-x. 2 root root 6 May 13 12:10 prince

-rw-r--r--. 1 tom root 0 May 13 12:16 project1

-rw-r--r--. 1 root dac 0 May 13 12:16 project2

-rw-------. 1 tom root 0 May 13 12:16 project3

-rw-rwx---. 1 root dbda 0 May 13 12:16 project4

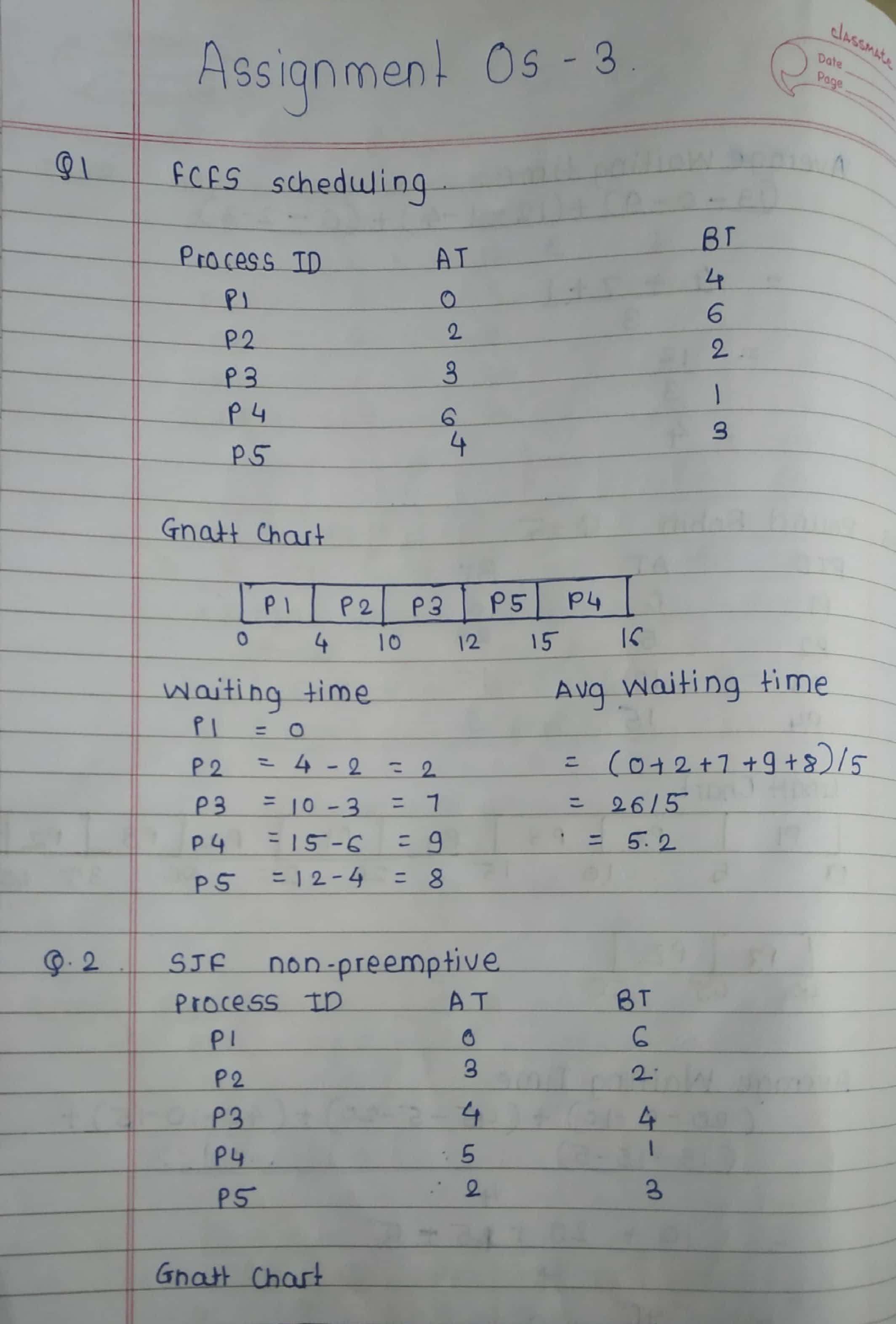
-----------------------------------------------------------------------------------------------------------------------

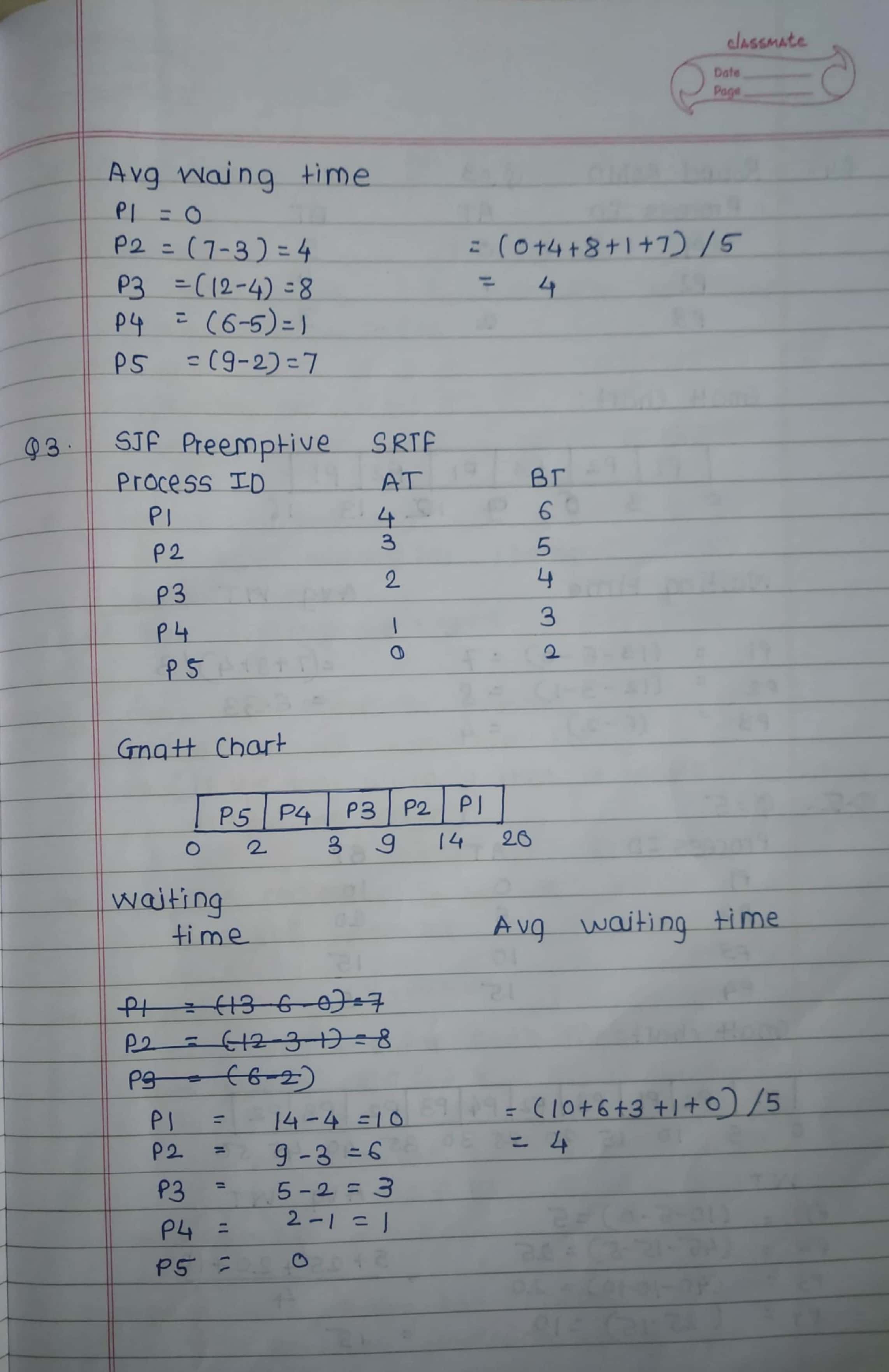
**ASSIGNMENT 3**

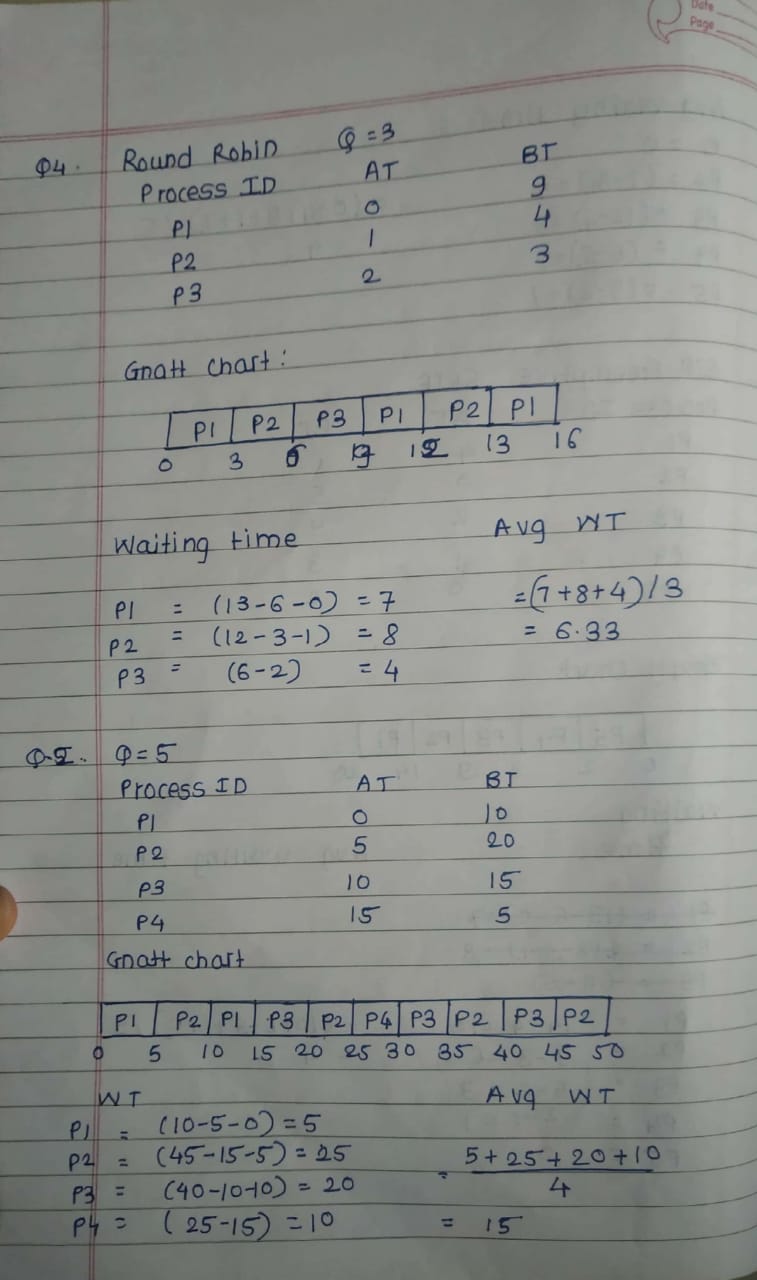
**Name : Meghana Dhananjay Pagare**

**Roll No : 1135**

**Sub : concepts of programming with OS**





**ASSIGNMENT 4**

**Name : Meghana Dhananjay Pagare**

**Roll No : 1135**

**Sub : concepts of programming with OS**

1. Write a shell script to calculate simple interest.

[user1@localhost ~]$ nano s.sh

GNU nano 2.3.1 File: s.sh

#!/bin/bash

read -p " enter principle = " p

read -p " enter rate = " r

read -p " enter years = " n

s=` expr $p \\* $r \\* $n `

s=` expr $s / 100 `

echo "simple interest is : $s"

//output

[user1@localhost ~]$ ./s.sh

enter principle = 5000

enter rate = 10

enter years = 1

simple interest is : 500

----------------------------------------------------------------------------------------------------------------------------------

2) Write a shell script to calculate salary from given basic.

Salary = basic + dp + da +hra +ma –pf

basic – to be taken as input

dp - 50 % of basic

da - 35 % of (basic + dp)

hra - 8 % of (basic + dp)

ma - 3 % of (basic + dp)

pf - 10% of (basic + dp)

#!/bin/bash

read -p " enter basic = " basic

dp=` echo "0.5\*$basic" | bc`

echo "dp=$dp"

da=` echo "0.35\*($basic+$dp) " | bc`

echo "da=$da"

hra=` echo " 0.08\*($basic+$da) " | bc`

echo "hra=$hra"

ma=` echo " 0.03\*($basic+$dp) "| bc`

echo "ma=$ma"

pf=` echo "0.1\*($basic+$dp) " | bc`

echo "pf=$pf"

salary=` echo "$basic+$dp+$da+$hra+$ma-$pf" |bc `

echo "total salary =$salary"

//output

[user1@localhost ~]$ ./s.sh

enter basic = 20000

dp=10000.0

da=10500.00

hra=2440.00

ma=900.00

pf=3000.0

total salary =40840.00

---------------------------------------------------------------------------------------------------------------

3) Write a shell script to calculate the average of a 3 number.

#!/bin/bash

read -p " enter 3 numbers to find avg = " a b c

avg=` echo "($a+$b+$c)/3" | bc`

echo "avg=$avg"

//output

[user1@localhost ~]$ ./s.sh

enter 3 numbers to find avg = 10 20 30

avg=20

------------------------------------------------------------------------------------------------------------

4). Write a shell script to create a command line calculator.

e.g. input : mycal 5 + 5 Result : 10 , input : mycal 5 / 5 result : 1

#!/bin/bash

echo " enter 1 for add ,2 for sub ,3 for div , 4 for mult : "

read -p "enter 2 num " a b

read -p " enter choice =" c

case $c in

1) add=` echo "$a+$b" | bc`

echo "add=$add"

;;

2) sub=` echo "$a-$b" | bc`

echo "sub=$sub"

;;

3) div=` echo "$a/$b" | bc`

echo "div=$div"

;;

4) mult=` echo "$a\*$b" | bc`

echo "mult=$mult"

;;

Esac

//output

[user1@localhost ~]$ ./s.sh

enter 1 for add ,2 for sub ,3 for div , 4 for mult :

enter 2 num 5 5

enter choice =1

add=10

[user1@localhost ~]$ ./s.sh

enter 1 for add ,2 for sub ,3 for div , 4 for mult :

enter 2 num 5 5

enter choice =2

sub=0

[user1@localhost ~]$ ./s.sh

enter 1 for add ,2 for sub ,3 for div , 4 for mult :

enter 2 num 5 5

enter choice =3

div=1

[user1@localhost ~]$ ./s.sh

enter 1 for add ,2 for sub ,3 for div , 4 for mult :

enter 2 num 5 5

enter choice =4

mult=25

---------------------------------------------------------------------------------------------------

5) Write a shell script to accept 2 numbers and display which number is greater

#!/bin/bash

read -p "enter 2 num " a b

if [ $a -gt $b ]

then

echo "$a is greater "

else

echo "$b is greater"

fi

//output

[user1@localhost ~]$ ./s.sh

enter 2 num 34 5

34 is greater

[user1@localhost ~]$ ./s.sh

enter 2 num 34 10000

10000 is greater

------------------------------------------------------------------------------------

6) Create a script to

Create user , Delete user , Create group , delete Group using case

#!/bin/bash

echo "type 1 for create user ,2 for delete user ,3 for create group, 4 for dele$

read -p " enter choice : " c

case $c in

1) echo "enter username : " name

read name

sudo useradd $name

tail -n 3 /etc/passwd

;;

2) echo "enter user to del : " del

read del

sudo userdel $del

tail -n 3 /etc/passwd

;;

3) echo "enter grp to add : " grp

read grp

sudo groupadd $grp

tail -n 3 /etc/group

;;

4) echo " enter grp to del :" grpdel

read grpdel

sudo groupdel $grpdel

tail -n 3 /etc/group

;;

5)exit 0 ;;

esac

//output

[user1@localhost ~]$ ./s.sh

type 1 for create user ,2 for delete user ,3 for create group, 4 for delete group

enter choice : 1

enter username : name

ashu

useradd: warning: the home directory already exists.

Not copying any file from skel directory into it.

Creating mailbox file: File exists

sam:x:1003:1003::/home/sam:/bin/bash

prince:x:1004:1004::/home/prince:/bin/bash

ashu:x:1005:1008::/home/ashu:/bin/bash

[user1@localhost ~]$ ./s.sh

type 1 for create user ,2 for delete user ,3 for create group, 4 for delete group

enter choice : 2

enter user to del : del

ashu

bob:x:1002:1002::/home/bob:/bin/bash

sam:x:1003:1003::/home/sam:/bin/bash

prince:x:1004:1004::/home/prince:/bin/bash

[user1@localhost ~]$ ./s.sh

type 1 for create user ,2 for delete user ,3 for create group, 4 for delete group

enter choice : 3

enter grp to add : grp

ashu

dbda:x:1006:bob

ditiss:x:1007:sam

ashu:x:1008:

[user1@localhost ~]$ ./s.sh

type 1 for create user ,2 for delete user ,3 for create group, 4 for delete group

enter choice : 4

enter grp to del : grpdel

ashu

dac:x:1005:tom

dbda:x:1006:bob

ditiss:x:1007:sam

-------------------------------------------------------------------------------------

**ASSIGNMENT 5**

**Name : Meghana Dhananjay Pagare**

**Roll No : 1135**

**Sub : concepts of programming with OS**

1. Write a script to find out String is palindrome or not.

#!/bin/bash

read -p "enter number " num

s=0

rev=""

temp=$num

while [ $num -gt 0 ];

do

s=$(( $num % 10 ))

num=$(( $num / 10 ))

rev=$( echo $rev$s )

done

if [ $temp = $rev ];

then

echo "number is palidrome"

else

echo "number is not palidrome"

fi

//output

[user1@localhost ~]$ ./s.sh

enter number 222

number is palidrome

[user1@localhost ~]$ nano s.sh

[user1@localhost ~]$ ./s.sh

enter number 3443

number is palidrome

[user1@localhost ~]$ ./s.sh

enter number 324

number is not palindrome

-----------------------------------------------------------------------------------------------------

1. Write a shell script to accept 10 numbers and tell how many are +tive, -tive and zero.

#!/bin/bash

read -p "enter numbers : " n

if [ $n -lt 0 ]

then

echo "negative"

elif [ $n -gt 0 ]

then

echo "positive"

else

echo "zero"

fi

//output

[user1@localhost ~]$ ./s.sh

enter numbers : 223

positive

[user1@localhost ~]$ nano s.sh

[user1@localhost ~]$ ./s.sh

enter numbers : -223

negative

------------------------------------------------------------------------------

3)Write a shell script to print given number’s sum of all digits (eg. If number is 123, then it’s sum of all digits will be 1+2+3=6)

#!/bin/bash

read -p "enter number : " n

sum=0

while [ $n -gt 0 ]

do

rem=`expr $n % 10`

sum=`expr $sum + $rem`

n=`expr $n / 10`

done

echo $sum

//output

[user1@localhost ~]$ ./s.sh

enter number : 22334

14

-----------------------------------------------------------------------------

4)Write a shell script to display the prime numbers from 1 to n ( n is a given number )

#!/bin/bash

echo "enter the range"

read n

echo "the prime no are:"

m=2

while [ $m -le $n ]

do

i=2

flag=0

while [ $i -le `expr $m / 2` ]

do

if [ ` expr $m % $i ` -eq 0 ]

then

flag=1

break

fi

i=` expr $i + 1`

done

if [ $flag -eq 0 ]

then

echo $m

fi

m=` expr $m + 1`

done

//output

[user1@localhost ~]$ ./s.sh

enter the range

12

the prime no are:

2

3

5

7

11

---------------------------------------------------------------------------------

5)Write a shell script to find whether a given year is leap year or not

#!/bin/bash

read -p "enter year : " y

a=`expr $y%4`

b=`expr $y%100`

c=`expr $y%400`

if [[ $a -eq 0 && $b -ne 0 || $c -eq 0 ]]

then

echo "$y is a leap year "

else

echo "$y is not a leap year "

fi

//output

[user1@localhost ~]$ ./s.sh

enter year : 2020

2020 is a leap year

[user1@localhost ~]$ ./s.sh

enter year : 2000

2000 is a leap year

[user1@localhost ~]$ ./s.sh

enter year : 2100

2100 is not a leap year

[user1@localhost ~]$ ./s.sh

enter year : 2019

2019 is not a leap year

----------------------------------------------------------------------------

**ASSIGNMENT 6**

**Name : Meghana Dhananjay Pagare**

**Roll No : 1135**

**Sub : concepts of programming with OS**

1.Write a shell script to create a menu driven program for adding, deletion or finding a record in a database. Database should have the field like rollno, name, semester and marks of three subjects. Last option of the menu should be to exit the menu.

#!/bin/bash

echo "1.Add records "

echo "2.Delet records "

echo "3.Find record "

echo "4.Exit "

read -p "enter your choice : " reply

case $reply in

1) read -p "enter first name: " fname

read -p "enter last name: " lname

read -p "enter roll no: " roll

read -p "enter course name: " course

read -p "enter marks in subject1: " S1

read -p "enter marks in subject2: " S2

read -p "enter marks in subject3: " S3

echo "$fname:$lname:$roll:$course:$S1:$S2:$S3" >> database

echo "Record is added "

cat database

;;

2) read -p "Enter roll no" roll

grep -v "$roll" database > tmpfile && mv tmpfile database

echo "Record is deleted"

cat database

;;

3) read -p "enter roll number " roll

echo "Name: "

grep "$roll" database | cut -d ':' -f 1,2 --output-delimiter=' '

echo "roll no : "

grep "$roll" database | cut -d ':' -f 3

echo "course: "

grep "$roll" database | cut -d ':' -f 4

echo "marks in subject1: "

grep "$roll" database | cut -d ':' -f 5

echo "marks in subject2: "

grep "$roll" database | cut -d ':' -f 6

echo "marks in subject3: "

grep "$roll" database | cut -d ':' -f 7

;;

4) echo "Exit"

exit

;;

\*) echo "Invalid choice "

;;

esac

//output

[user1@localhost ~]$ ./s.sh

1.Add records

2.Delet records

3.Find record

4.Exit

enter your choice : 1

enter first name: mrunali

enter last name: pagare

enter roll no: 1137

enter course name: dac

enter marks in subject1: 56

enter marks in subject2: 56

enter marks in subject3: 78

Record is added

meghana:pagare:1135:dac:40:40:40

meghana:pagare:1135:gahga:88:99:99

Meghana:Pagare:1135:dac:40:40:40

Ashutosh:pagare:1136:dac:34:45:45

mrunali:pagare:1137:dac:56:56:78

[user1@localhost ~]$ ./s.sh

1.Add records

2.Delet records

3.Find record

4.Exit

enter your choice : 2

Enter roll no1136

Record is deleted

meghana:pagare:1135:dac:40:40:40

meghana:pagare:1135:gahga:88:99:99

Meghana:Pagare:1135:dac:40:40:40

mrunali:pagare:1137:dac:56:56:78

[user1@localhost ~]$ ./s.sh

1.Add records

2.Delet records

3.Find record

4.Exit

enter your choice : 3

enter roll number 1137

Name:

mrunali pagare

roll no :

1137

course:

dac

marks in subject1:

56

marks in subject2:

56

marks in subject3:

78

[user1@localhost ~]$ ./s.sh

1.Add records

2.Delet records

3.Find record

4.Exit

enter your choice : 4

Exit

----------------------------------------------------------------------------------------------------------------

2.Write a unix shell to add records to a file called item.dat The fields being itemcode, qty, sold and rate

item\_code to be generated

qty\_sold should be greater than 0

1. rate between 100 to 10000

#!/bin/bash

read -p "enter a itemcode: " code

read -p "enter quantity: " qty

read -p "enter rate: " rate

num=` tail -n 1 item.dat | cut -d ' ' -f 1 `

num=$((num+1))

echo "$num $code $qty $rate " >> item.dat

cat item.dat

//output

[user1@localhost ~]$ ./s.sh

enter a itemcode: 15

enter quantity: 200

enter rate: 2000

1 15 200 2000

[user1@localhost ~]$ ./s.sh

enter a itemcode: 43

enter quantity: 2500

enter rate: 1200

1 15 200 2000

2 43 2500 1200

-------------------------------------------------------------------------------------------------------

3.Write a scripts which copies the content of file1 to file2 without using cp command

It should check If file has a read permissions if not it should print an error message. If file2 exits then it should ask the user whether he wants to overwrite it

#!/bin/bash

read -p "Enter file name" fname1

if [ -r $fname1 ]

then

read -p "Enter file name where you want to copy data: " fname2

if test -f $fname2

then

read -p "you want to overwrite(y/n): " ch

case $ch in

y)cat $fname1 > $fname2

;;

n)exit

;;

esac

fi

else

echo "File is not readable"

fi

//output

[user1@localhost ~]$ chmod +r file1

chmod: cannot access ‘file1’: No such file or directory

[user1@localhost ~]$ chmod +r file1.sh

[user1@localhost ~]$ ./s.sh

Enter file namefile1.sh

Enter file name where you want to copy data: file2,sh

-----------------------------------------------------------------------------------------------------------

4. Write a shell scripts that delete all files in current directory with 0 byte.

#!/bin/bash

echo "files in current directory with 0 byte are: "

find . -maxdepth 1 -type f -size 0

echo "enter Y to delete files with size 0 "

echo "enter q to quit "

read -p "enter youe choice: " reply

case $reply in

Y|y) find . -maxdepth 1 -type f -size 0 -delete

echo "all files with 0 byte are deleted successfully ..."

;;

Q|q) echo "operation cancelled "

exit

;;

\*) echo "Invalid choice "

;;

esac

//output

[user1@localhost ~]$ ./s.sh

files in current directory with 0 byte are:

./film5.avi

./photo3.jpg

./photo5.jpg

./songs1

./songs1.mp3

./songs2.mp3

./songs3.mp3

./songs4.mp3

./songs5.mp3

./songs6.mp3

./film1.avi

./film2.avi

./film3.avi

./film4.avi

./film6.avi

./photo1.jpg

./photo2.jpg

./photo4.jpg

./photo6.jpg

./file1

enter Y to delete files with size 0

enter q to quit

enter youe choice: y

all files with 0 byte are deleted successfully ...  
------------------------------------------------------------------------------------------------------

5. Write a shell script to display a directory listing as follows. Your home directory is <home directory name> 3

File name date time permission

------------- ------ ----- ---------------

Filename1 date time permission

Filename2 date time permission

Filename3 date time permission

………..

………..

Total no. of files : <total number>

Total no of normal file : <number>

Total no of directory : <number>

#!/bin/bash

echo

echo "your home directory file $HOME"

echo

echo "date time permissiomns file name"

echo

echo "---------------------------------------------------------------------"

ls -l | while read perm bsize month day time file; do echo "$month $day $time $$

echo "---------------------------------------------------------------------"

echo "Total Number of files: "

find . -maxdepth 1 | wc -l

echo "Total number of normal files: "

find . -maxdepth 1 -type f | wc -l

echo "Total number of directories : "

find . -maxdepth 1 -type d | wc -l

//output

[user1@localhost ~]$ ./a2.sh

your home directory file /home/user1

date time permissiomns file name

-----------------------------------------------------------

total

user1 user1 61 -rw-rw-r--. May 20 13:07 a1.c

user1 user1 555 -rwxrwxr-x. May 21 12:53 a2.sh

user1 user1 152 -rw-rw-r--. May 20 13:07 a.c

user1 user1 8496 -rwxrwxr-x. May 20 14:03 a.out

user1 user1 13 -rw-------. May 20 13:55 assign.sh.save

user1 user1 134 -rw-rw-r--. May 19 15:12 database

user1 user1 6 drwxr-xr-x. May 12 09:59 Desktop

user1 user1 6 drwxr-xr-x. May 12 09:59 Documents

user1 user1 6 drwxr-xr-x. May 12 09:59 Downloads

user1 user1 110 drwxrwxr-x. May 12 14:00 family

user1 user1 8 --w--w----. May 20 06:50 file1.sh

user1 user1 3 -rw-rw-r--. May 20 06:42 file2

user1 user1 32 -rw-rw-r--. May 20 07:41 file3

user1 user1 31 -rw-rw-r--. May 19 15:58 item.dat

user1 user1 6 drwxrwxr-x. May 12 13:08 Music

user1 user1 156 -rw-rw-r--. May 20 13:59 orphan.c

user1 user1 4096 drwxr-xr-x. May 21 10:48 Pictures

user1 user1 6 drwxr-xr-x. May 12 09:59 Public

user1 user1 1 -rw-------. May 18 15:44 q.save

user1 user1 432 -rwxrwxr-x. May 20 08:29 s.sh

user1 user1 6 drwxr-xr-x. May 12 09:59 Templates

user1 user1 6 drwxr-xr-x. May 12 13:22 Videos

user1 user1 110 drwxrwxr-x. May 12 14:05 work

user1 user1 147 -rw-rw-r--. May 20 13:55 zombie.c

----------------------------------------------------------------------

Total Number of files:

38

Total number of normal files:

22

Total number of directories :

16

-------------------------------------------------------------------------------------------------------------------

**ASSIGNMENT 7**

**Name : Meghana Dhananjay Pagare**

**Roll No : 1135**

**Sub : concepts of programming with OS**

1. Create Child process  using fork()

#include<stdio.h>

#include<sys/types.h>

#include<unistd.h>

int main()

{

fork();

printf ("using fork() system call..... \n");

return 0;

}

//output

[user1@localhost ~]$ gcc a.c

[user1@localhost ~]$ ./a.out

using fork() system call.....

[user1@localhost ~]$ using fork() system call.....

-------------------------------------------------------------------------------------------------------------------------

1. Create orphan process

#include<stdlib.h>

#include<sys/types.h>

#include<unistd.h>

int main()

{

int pid=fork();

if(pid>0)

{

exit(0);

}

else if(pid==0)

{

sleep(60);

}

return 0;

}

//output

[user1@localhost ~]$ gcc orphan.c

[user1@localhost ~]$ ./a.out &

[1] 13721

[user1@localhost ~]$ ps -aux | grep a.out

user1 13703 0.0 0.0 4212 84 pts/0 S 14:02 0:00 ./a.out

user1 13722 0.0 0.0 4212 84 pts/0 S 14:03 0:00 ./a.out

user1 13724 0.0 0.0 112808 976 pts/0 R+ 14:03 0:00 grep --color=auto a.out

[1] + Done ./a.out

[user1@localhost ~]$ ps -o ppid 13722

PPID

1

-------------------------------------------------------------------------------------------------------------

1. Create Zombie process

#include<stdlib.h>

#include<sys/types.h>

#include<unistd.h>

int main()

{

int pid=fork();

if(pid>0)

{

sleep(80);

}

else

{

exit(0);

}

return 0;

}

//output

[user1@localhost ~]$ nano zombie.c

[user1@localhost ~]$ ./a.out &

[1] 13458

[user1@localhost ~]$ pstree -p 13458

a.out(13458)───a.out(13459)

[user1@localhost ~]$ ps -aux | grep 13459

user1 13459 0.0 0.0 0 0 pts/1 Z 13:52 0:00 [a.out] <defunct>

user1 13462 0.0 0.0 112808 960 pts/1 R+ 13:52 0:00 grep --color=auto 13459

---------------------------------------------------------------------------------------------------------------