

2a. Write a C program to create a new process using Fork Operation. Make the parent Process to wait until child process Terminates. Once the child Terminates collect exit status and print the exit status of the child in the parent process using WEXITSTATUS macro.

Program:

```
#include<unistd.h>
#include<stdio.h>
#include<sys/wait.h>
#include<stdlib.h>
#include<fcntl.h>

int main(int argc, char *argv[])
{
    int fd, exitstatus;
    int exitval=10;
    fd=open(argv[1], O_WRONLY | O_CREAT | O_TRUNC, 0664);
    write(fd, "Original Process writes\n", 24);
    switch(fork())
    {
        case 0: write(fd, "Child Process writes\n", 21);
                close(fd);
                printf("Child : Terminating with exit value %d\n", exitval);
                exit(exitval);
                break;

        default: wait(&exitstatus);
                 printf("Parent : Child terminated with exit value %d\n", WEXITSTATUS(exitstatus));
                 write(fd, "Parent Process writes\n", 22);
                 exit(20);
    }
```

```
}  
  
}
```

output :

```
file1.txt  ×  
Labtest2 > file1.txt  
1 Original Process writes  
2 Child Process writes  
3 Parent Process writes  
4   
  
dhanrz@pop-os:~/LAB_WORK/UNIX_IBM18CS027/Labtest2$ g++ 2a.c -o 2a  
dhanrz@pop-os:~/LAB_WORK/UNIX_IBM18CS027/Labtest2$ ./2a file1.txt  
Child : Terminating with exit value 10  
Parent : Child terminated with exit value 10  
dhanrz@pop-os:~/LAB_WORK/UNIX_IBM18CS027/Labtest2$
```

2b .Write a Shell Script that accepts two filenames as arguments .Check if the permissions for these files are identical and if the permissions are identical output the common permissions,otherwise output each filename followed by its permission.

Program:

```

#!/bin/sh

if [ $# -ne 2 ]
then
echo "no arguments pls enter 2 arguments"
elif [ ! -e $1 -o ! -e $2 ]
then
echo "File does not exist"
else
per1=`ls -l $1 | cut -c 2-10`
per2=`ls -l $2 | cut -c 2-10`
if [ $per1 = $per2 ]
then
echo "Permissions are Identical:permission is $per1"
else
echo "Permissions are not Identical"
echo "permission of $1 is $per1"
echo "permission of $2 is $per2"
fi
fi

```

Output:

```

dhanrz@pop-os:~/LAB_WORK/UNIX_1BM18CS027/Labtest2$ sh 2b.sh 2a file1.txt
Permissions are not Identical
permission of 2a is rwxrwxr-x
permission of file1.txt is rw-rw-r--
dhanrz@pop-os:~/LAB_WORK/UNIX_1BM18CS027/Labtest2$ sh 2b.sh file2.txt file1.txt
Permissions are Identical:permission is rw-rw-r--

```