NAME – NISHANT PRATAP SINGH

ROLL NO. - AE-1233

SEMESTER- III

ASSIGNMENT – OPERATING SYSTEM

SUBMITTED TO – DR. VIBHA GAUR

Execute various LINUX commands for:
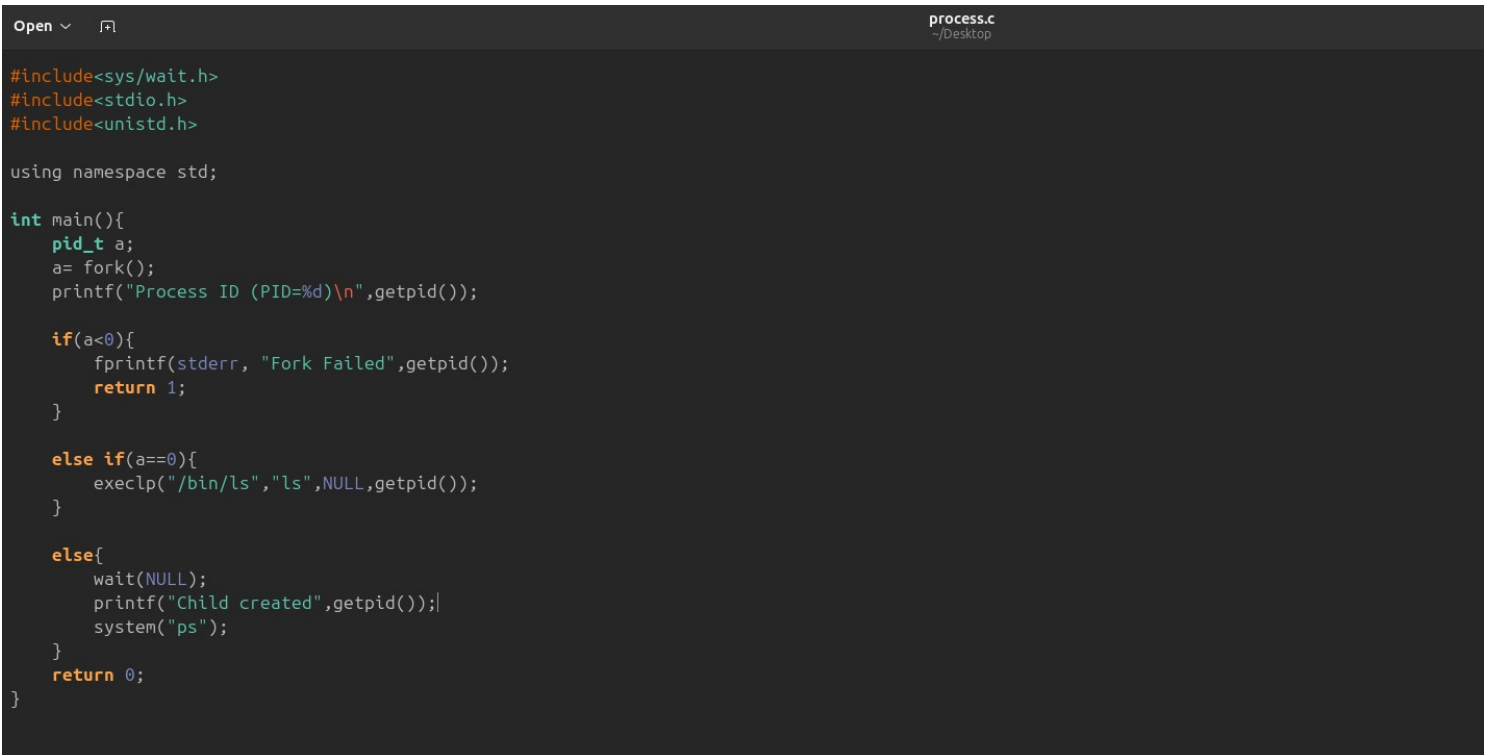
i. Process Control: fork, getpid, ps, kill, sleep
ii. Communication: Input-output redirection, Pipe
iii. Protection Management: chmod, chown, chgrp

fork():

The Fork system call is used for creating a new process Unix systems, which is called the(child process), which runs concurrently with the process that makes the fork() call (parent process).

Different values returned by fork()**:**

- **Negative Value**: The creation of a child process was unsuccessful.
- **Zero**: Returned to the newly created child process.
- **Positive value**: Returned to parent or caller.

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

using namespace std;

int main(){
    pid_t a;
    a= fork();
    printf("Process ID (PID=%d)\n",getpid());

    if(a<0){
        fprintf(stderr, "Fork Failed",getpid());
        return 1;
    }

    else if(a==0){
        execlp("/bin/ls","ls",NULL,getpid());
    }

    else{
        wait(NULL);
        printf("Child created",getpid());
        system("ps");
    }
    return 0;
}
```

After running the code in terminal:

```
nishant@nishant-VirtualBox:~/Desktop$ ./a.out
Process ID (PID=8971)
Process ID (PID=8972)
a.out  copy.cpp  dest_file.txt  fork.cpp  process.c  src_file.txt
    PID TTY          TIME CMD
   8909 pts/0    00:00:00 bash
   8971 pts/0    00:00:00 a.out
   8973 pts/0    00:00:00 sh
   8974 pts/0    00:00:00 ps
Child creatednishant@nishant-VirtualBox:~/Desktop$
```

Kill():
Built-in command which is used to terminate processes manually.

Syntax:
kill [signal] PID

Parameters:

- PID =The `kill` command requires the process ID (PID) of the process we want to terminate.

- [signal] =Specify the signal.


```c
#include<stdio.h>
#include<unistd.h>
#include<sys/types.h>
#include<signal.h>

int main(int argc, int *argv[]){
    printf("The pid is = %d\n",getpid());
    sleep(5);
    kill(getpid(), SIGSEGV);
return 0;
}
```

After running the code in terminal:

```
nishant@nishant-VirtualBox:~/Desktop$ gcc kill.c
nishant@nishant-VirtualBox:~/Desktop$ ./a.out
The pid is = 10225
Segmentation fault (core dumped)
nishant@nishant-VirtualBox:~/Desktop$
```

Sleep():

Sets the process to wait until the specified amount of time proceeds. The sleep system call is used to take a time value as a parameter, specifying the minimum amount of time that the process is to sleep before resuming execution.

Syntax:
     sleep(time);

getpid():

Returns the process ID of the calling process.

Syntax:
     pid_t getpid(void);

Parameters:

Return type: getpid() returns the process ID of the current process. It never throws any error therefore is always successful.

```
nishant@nishant-VirtualBox:~/Desktop$ gcc kill.c
nishant@nishant-VirtualBox:~/Desktop$ ./a.out
The pid is = 10225
Segmentation fault (core dumped)
nishant@nishant-VirtualBox:~/Desktop$
```

Ps:

Viewing information related with the processes on a system which stands as abbreviation for "Process Status".
ps command is used to list the currently running processes and their PIDs.

Syntax:

ps [options]

parameters:

- **PID –** The unique process ID
- **TTY –** Terminal type that the user is logged into
- **TIME –**Amount of CPU in minutes and seconds that the process has been running.
- **CMD –** Name of the command that launched the process.

```
PID TTY          TIME CMD
8909 pts/0    00:00:00 bash
8971 pts/0    00:00:00 a.out
8973 pts/0    00:00:00 sh
8974 pts/0    00:00:00 ps
```

Input-output redirection:

Whenever an individual runs a command, it can take input, give output, or do both. Redirection helps us redirect these input and output functionalities to the files or folders we want, and we can use special commands or characters to do so.

Syntax:

cat > file.txt - Used to write in the file
cat < file.txt - Use to read the file
cat >> file.txt - Used to write in the existing file

```
nishant@nishant-VirtualBox:~/Desktop$ ls -l
total 40
-rwxrwxr-x 1 nishant nishant 16088 Sep 19 19:09 a.out
-rw-rw-r-- 1 nishant nishant  1119 Sep 19 17:34 copy.cpp
-rw-rw-r-- 1 nishant nishant     0 Sep 19 17:39 dest_file.txt
-rw-rw-r-- 1 nishant nishant   616 Sep 19 18:22 fork.cpp
-rw-rw-r-- 1 nishant nishant    18 Sep 19 20:20 input.txt
-rw-rw-r-- 1 nishant nishant   209 Sep 19 19:09 kill.c
--w-rw-r-- 1 nishant nishant   455 Sep 19 18:46 process.c
-rw-rw-r-- 1 nishant nishant    66 Sep 19 17:20 src_file.txt
nishant@nishant-VirtualBox:~/Desktop$ cat >> input.txt
I also want to tell something

Some other daynishant@nishant-VirtualBox:~/Desktop$ ls -l
total 40
-rwxrwxr-x 1 nishant nishant 16088 Sep 19 19:09 a.out
-rw-rw-r-- 1 nishant nishant  1119 Sep 19 17:34 copy.cpp
-rw-rw-r-- 1 nishant nishant     0 Sep 19 17:39 dest_file.txt
-rw-rw-r-- 1 nishant nishant   616 Sep 19 18:22 fork.cpp
-rw-rw-r-- 1 nishant nishant    63 Sep 19 20:23 input.txt
-rw-rw-r-- 1 nishant nishant   209 Sep 19 19:09 kill.c
--w-rw-r-- 1 nishant nishant   455 Sep 19 18:46 process.c
-rw-rw-r-- 1 nishant nishant    66 Sep 19 17:20 src_file.txt
nishant@nishant-VirtualBox:~/Desktop$ cat < input.txt
My name is NPS
ByeI also want to tell something

Some other daynishant@nishant-VirtualBox:~/Desktop$
```

```
nishant@nishant-VirtualBox:~/Desktop$ ls -l
total 36
-rwxrwxr-x 1 nishant nishant 16088 Sep 19 19:09 a.out
-rw-rw-r-- 1 nishant nishant  1119 Sep 19 17:34 copy.cpp
-rw-rw-r-- 1 nishant nishant     0 Sep 19 17:39 dest_file.txt
-rw-rw-r-- 1 nishant nishant   616 Sep 19 18:22 fork.cpp
-rw-rw-r-- 1 nishant nishant   209 Sep 19 19:09 kill.c
--w-rw-r-- 1 nishant nishant   455 Sep 19 18:46 process.c
-rw-rw-r-- 1 nishant nishant    66 Sep 19 17:20 src_file.txt
nishant@nishant-VirtualBox:~/Desktop$ cat > input.txt
My name is NPS
Byenishant@nishant-VirtualBox:~/Desktop$ ls -l
total 40
-rwxrwxr-x 1 nishant nishant 16088 Sep 19 19:09 a.out
-rw-rw-r-- 1 nishant nishant  1119 Sep 19 17:34 copy.cpp
-rw-rw-r-- 1 nishant nishant     0 Sep 19 17:39 dest_file.txt
-rw-rw-r-- 1 nishant nishant   616 Sep 19 18:22 fork.cpp
-rw-rw-r-- 1 nishant nishant    18 Sep 19 20:20 input.txt
-rw-rw-r-- 1 nishant nishant   209 Sep 19 19:09 kill.c
--w-rw-r-- 1 nishant nishant   455 Sep 19 18:46 process.c
-rw-rw-r-- 1 nishant nishant    66 Sep 19 17:20 src_file.txt
nishant@nishant-VirtualBox:~/Desktop$ cat < input.txt
My name is NPS
Byenishant@nishant-VirtualBox:~/Desktop$
```

Pipe():

A pipe is a form of redirection (transfer of standard output to some other destination) to send the output of one command/program/process to another command/program/process for further processing. You can make it do so by using the pipe character '|'.

Syntax:

command_1 | command_2 | command_3 |.... | command_N

ls | grep filename:

The output of the ls command is sent to the input of the grep command, and the result is a list of files that match the search term.

cat filename| head -n | tail -m:

This command selects the first n lines through (head -n) command and that will be input to (tail -m) command which will finally print last m lines from those n lines.

sort filename | uniq:

This will sort the given file and print the unique values only.

```
nishant@nishant-VirtualBox:~/Desktop$ ls -l
total 40
-rwxrwxr-x 1 nishant nishant 16088 Sep 19 19:09 a.out
-rw-rw-r-- 1 nishant nishant  1119 Sep 19 17:34 copy.cpp
-rw-rw-r-- 1 nishant nishant     0 Sep 19 17:39 dest_file.txt
-rw-rw-r-- 1 nishant nishant   616 Sep 19 18:22 fork.cpp
-rw-rw-r-- 1 nishant nishant    63 Sep 19 20:23 input.txt
-rw-rw-r-- 1 nishant nishant   209 Sep 19 19:09 kill.c
--w-rw-r-- 1 nishant nishant   455 Sep 19 18:46 process.c
-rw-rw-r-- 1 nishant nishant    66 Sep 19 17:20 src_file.txt
nishant@nishant-VirtualBox:~/Desktop$ cat input.txt
My name is NPS
ByeI also want to tell something

Some other daynishant@nishant-VirtualBox:~/Desktop$ sort input.txt | uniq

ByeI also want to tell something
My name is NPS
Some other day
nishant@nishant-VirtualBox:~/Desktop$
```

Chmod:

The chmod command is used to change the access mode of a file. The name is an abbreviation of change mode: read, write, and execute simultaneously represented by `r`, `w` and `x`.

```
nishant@nishant-VirtualBox:~/Desktop$ ls -l
total 36
-rwxrwxr-x 1 nishant nishant 16088 Sep 19 19:09 a.out
-rw-rw-r-- 1 nishant nishant  1119 Sep 19 17:34 copy.cpp
-rw-rw-r-- 1 nishant nishant     0 Sep 19 17:39 dest_file.txt
-rw-rw-r-- 1 nishant nishant   616 Sep 19 18:22 fork.cpp
-rw-rw-r-- 1 nishant nishant   209 Sep 19 19:09 kill.c
-rw-rw-r-- 1 nishant nishant   455 Sep 19 18:46 process.c
-rw-rw-r-- 1 nishant nishant    66 Sep 19 17:20 src_file.txt
nishant@nishant-VirtualBox:~/Desktop$ chmod u-r process.c
nishant@nishant-VirtualBox:~/Desktop$ ls -l
total 36
-rwxrwxr-x 1 nishant nishant 16088 Sep 19 19:09 a.out
-rw-rw-r-- 1 nishant nishant  1119 Sep 19 17:34 copy.cpp
-rw-rw-r-- 1 nishant nishant     0 Sep 19 17:39 dest_file.txt
-rw-rw-r-- 1 nishant nishant   616 Sep 19 18:22 fork.cpp
-rw-rw-r-- 1 nishant nishant   209 Sep 19 19:09 kill.c
--w-rw-r-- 1 nishant nishant   455 Sep 19 18:46 process.c
-rw-rw-r-- 1 nishant nishant    66 Sep 19 17:20 src_file.txt
nishant@nishant-VirtualBox:~/Desktop$
```

chgrp:

The chgrp command in Linux is used to change the group ownership of a file or directory.

Syntax:

chgrp [OPTION]… GROUP FILE…


```
nishant@nishant-VirtualBox:~/Desktop$ ls -l
total 40
-rwxrwxr-x 1 nishant nishant 16088 Sep 19 19:09 a.out
-rw-rw-r-- 1 nishant nishant  1119 Sep 19 17:34 copy.cpp
-rw-rw-r-- 1 nishant nishant     0 Sep 19 17:39 dest_file.txt
-rw-rw-r-- 1 nishant nishant   616 Sep 19 18:22 fork.cpp
-rw-rw-r-- 1 nishant nishant    63 Sep 19 20:23 input.txt
-rw-rw-r-- 1 nishant nishant   209 Sep 19 19:09 kill.c
--w-rw-r-- 1 nishant nishant   455 Sep 19 18:46 process.c
-rw-rw-r-- 1 nishant nishant    66 Sep 19 17:20 src_file.txt
nishant@nishant-VirtualBox:~/Desktop$ sudo chgrp yash fork.cpp
[sudo] password for nishant:
nishant@nishant-VirtualBox:~/Desktop$ ls -l
total 40
-rwxrwxr-x 1 nishant nishant 16088 Sep 19 19:09 a.out
-rw-rw-r-- 1 nishant nishant  1119 Sep 19 17:34 copy.cpp
-rw-rw-r-- 1 nishant nishant     0 Sep 19 17:39 dest_file.txt
-rw-rw-r-- 1 nishant yash      616 Sep 19 18:22 fork.cpp
-rw-rw-r-- 1 nishant nishant    63 Sep 19 20:23 input.txt
-rw-rw-r-- 1 nishant nishant   209 Sep 19 19:09 kill.c
--w-rw-r-- 1 nishant nishant   455 Sep 19 18:46 process.c
-rw-rw-r-- 1 nishant nishant    66 Sep 19 17:20 src_file.txt
nishant@nishant-VirtualBox:~/Desktop$
```

chown:

To change owner, change the user and/or group ownership of each given File to a new Owner. Chown can also change the ownership of a file to match the user/group of an existing reference file.

Syntax:

chown owner_name file_name

```
nishant@nishant-VirtualBox:~/Desktop$ ls -l
total 40
-rwxrwxr-x 1 nishant nishant 16088 Sep 19 19:09 a.out
-rw-rw-r-- 1 nishant nishant  1119 Sep 19 17:34 copy.cpp
-rw-rw-r-- 1 nishant nishant     0 Sep 19 17:39 dest_file.txt
-rw-rw-r-- 1 nishant yash       616 Sep 19 18:22 fork.cpp
-rw-rw-r-- 1 nishant nishant    63 Sep 19 20:23 input.txt
-rw-rw-r-- 1 nishant nishant   209 Sep 19 19:09 kill.c
--w-rw-r-- 1 nishant nishant   455 Sep 19 18:46 process.c
-rw-rw-r-- 1 nishant nishant    66 Sep 19 17:20 src_file.txt
nishant@nishant-VirtualBox:~/Desktop$ chown yash process.c
chown: changing ownership of 'process.c': Operation not permitted
nishant@nishant-VirtualBox:~/Desktop$
```