1st debugging to know how files stored in proc

3 cases no redirection / out / out & err

```
ubuntu@ubuntu-virtual-machine:~/Code_Area/C/SPL_Course_Codes$ ./print_args a ba cac caca
argc=5
argv[0]=./print_args
argv[1]=a
argv[2]=ba
argv[3]=cac
argv[4]=caca
^C
ubuntu@ubuntu-virtual-machine:~/Code_Area/C/SPL_Course_Codes$ ./print_args a ba cac caca > out.txt

ubuntu@ubuntu-virtual-machine:~/Code_Area/C/SPL_Course_Codes$ ./print_args a ba cac caca >out.txt 2> err.txt
```

```
ubuntu@ubuntu-virtual-machine:~/Code_Area/C/SPL_Course_Codes$ ls -la /proc/`pidof ./print_args`/fd
total 0
dr-x------ 2 ubuntu ubuntu  0 Sep  3 21:20 .
dr-xr-xr-x 9 ubuntu ubuntu  0 Sep  3 21:20 ..
lrwx------ 1 ubuntu ubuntu 64 Sep  3 21:20 0 -> /dev/pts/0
lrwx------ 1 ubuntu ubuntu 64 Sep  3 21:20 1 -> /dev/pts/0
lrwx------ 1 ubuntu ubuntu 64 Sep  3 21:20 2 -> /dev/pts/0
ubuntu@ubuntu-virtual-machine:~/Code_Area/C/SPL_Course_Codes$ ls -la /proc/`pidof ./print_args`/fd
total 0
dr-x------ 2 ubuntu ubuntu  0 Sep  3 21:21 .
dr-xr-xr-x 9 ubuntu ubuntu  0 Sep  3 21:21 ..
lrwx------ 1 ubuntu ubuntu 64 Sep  3 21:21 0 -> /dev/pts/0
l-wx------ 1 ubuntu ubuntu 64 Sep  3 21:21 1 -> /home/ubuntu/Code_Area/C/SPL_Course_Codes/out.txt
lrwx------ 1 ubuntu ubuntu 64 Sep  3 21:21 2 -> /dev/pts/0
ubuntu@ubuntu-virtual-machine:~/Code_Area/C/SPL_Course_Codes$ ls -la /proc/`pidof ./print_args`/fd
total 0
dr-x------ 2 ubuntu ubuntu  0 Sep  3 21:21 .
dr-xr-xr-x 9 ubuntu ubuntu  0 Sep  3 21:21 ..
```

```
lrwx------ 1 ubuntu ubuntu 64 Sep  3 21:21 0 -> /dev/pts/0
l-wx------ 1 ubuntu ubuntu 64 Sep  3 21:21 1 ->
/home/ubuntu/Code_Area/C/SPL_Course_Codes/out.txt
l-wx------ 1 ubuntu ubuntu 64 Sep  3 21:21 2 ->
/home/ubuntu/Code_Area/C/SPL_Course_Codes/err.txt
```

Now Code to redirect errors

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>

int main(int argc, char ** argv)
{
    int err_fd;
    if (argc < 2)
    {
        printf("Usage: %s path-to-elf\n", argv[0]);
        exit(-1);
    }

    getchar();

    char* newargv[] = {argv[1], NULL};
    char* newenvp[] = {NULL};

    //create err_fd to the process
    if ((err_fd = open("err.txt", O_WRONLY | O_CREAT | O_TRUNC, 0644)) < 0){
        fprintf(stderr, "Could not open the file for error redirection\n");
        exit(-1);
    }

    dup2(err_fd, 2); // move err_fd to 2 which is default
    close(err_fd); // deletes err_fd
    getchar();

    execve(argv[1], newargv, newenvp);
    printf("Exec failed, kernel is not the mode of executing programs\n");
    return -2;
}
```