COOCHBEHAR GOVERMENT ENGINEERING COLLEGE

COMPUTER NETWORKS LAB

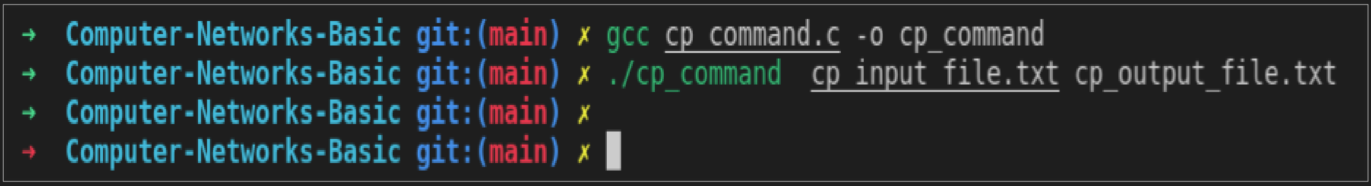
Nikti Paul

Roll: 34900119032
6th Semester, CSE

1. Write a C program to simulate the "cp" command.

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

int main(int argc, char* argv[]){

FILE *from_copy_file, *to_copy_file;

from_copy_file = fopen(argv[1],"r");
to_copy_file = fopen(argv[2],"w");

char current;
do{
        current = fgetc(from_copy_file);

        if (current != EOF)
                fputc(current,to_copy_file);

}while(current != EOF);


fclose(from_copy_file);
fclose(to_copy_file);
return 0;

}
```

Terminal Commands:



```
➜  Computer-Networks-Basic git:(main) ✗ gcc cp_command.c -o cp_command
➜  Computer-Networks-Basic git:(main) ✗ ./cp_command  cp_input_file.txt cp_output_file.txt
➜  Computer-Networks-Basic git:(main) ✗
➜  Computer-Networks-Basic git:(main) ✗ █
```

Output: (cp_output_file.txt):

```
Hello World!!
This is a cp command simulation
written by Nikti
```

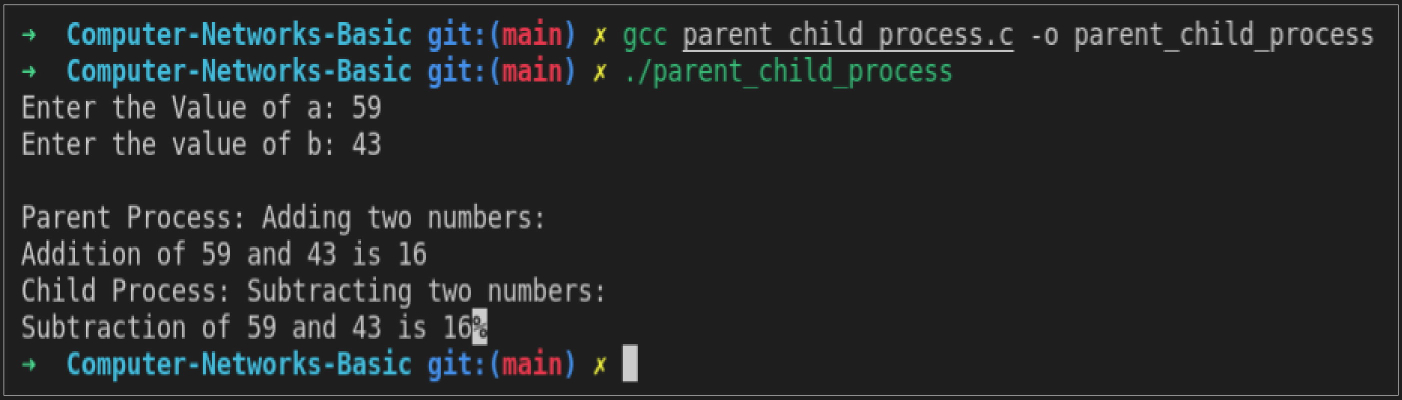
Codes and Files:

- ➔ *cp_command.c* in github
- ➔ *cp_input_file.txt* in github
- ➔ *cp_output_file.txt* in github

2. Write a C program to create a child process. The parent process will display the addition and the child process will display the subtraction of two numbers.

```
#include <stdio.h>

#include <stdlib.h>
#include <unistd.h>

int main(){


int a,b,prs,result;
printf("Enter the Value of a: ");
scanf("%d",&a);
printf("Enter the value of b: ");
scanf("%d", &b);

prs = fork();

if (prs == -1)
        printf("Process Failed");

else if (prs == 0){
        printf("\nChild Process: Subtracting two numbers: ");
        result = a - b;
        printf("\nSubtraction of %d and %d is %d",a,b,result);
        }

else if (prs > 0){
        printf("\nParent Process: Adding two numbers: ");
        result = a - b;
        printf("\nAddition of %d and %d is %d",a,b,result);
        }

}
```

Terminal Code and Output:



```
➜  Computer-Networks-Basic git:(main) ✗ gcc parent_child_process.c -o parent_child_process
➜  Computer-Networks-Basic git:(main) ✗ ./parent_child_process
Enter the Value of a: 59
Enter the value of b: 43

Parent Process: Adding two numbers:
Addition of 59 and 43 is 16
Child Process: Subtracting two numbers:
Subtraction of 59 and 43 is 16%
➜  Computer-Networks-Basic git:(main) ✗
```

Codes and Files:

➔ *parent_child_process c.* in github

3. Write a client server program to communicate between them (one way).

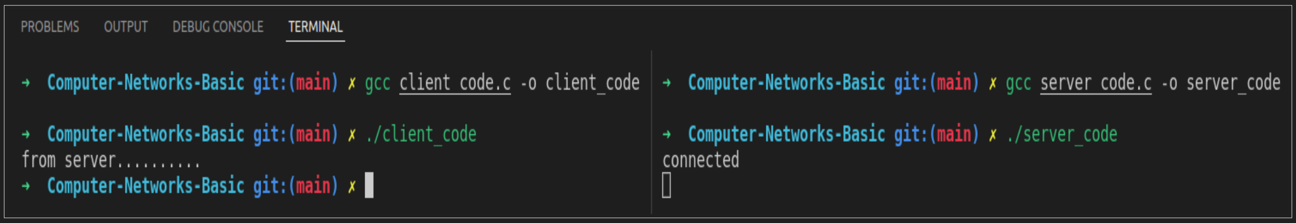
// client_code

```
#include<stdio.h>
#include<unistd.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include<sys/socket.h>
#include<sys/types.h>

int main()
{
        struct sockaddr_in c_addr;
        int c_fd,c_len;
        char buff[100];

        if((c_fd=socket(AF_INET,SOCK_STREAM,0))==-1)
                printf("error..socket\n");
        c_addr.sin_family=AF_INET;
        c_addr.sin_addr.s_addr=INADDR_ANY;
        c_addr.sin_port=3452;
        c_len=sizeof(c_addr);

        if(connect(c_fd,(struct sockaddr*)& c_addr,c_len)==-1)
                printf("error connect\n");

        read(c_fd,buff,100);
        printf("%s\n",buff);
        close(c_fd);
        return 0;
}
```

// server_code

```c
#include<stdio.h>

#include<unistd.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include<sys/socket.h>
#include<sys/types.h>
int main()
{
	struct sockaddr_in s_addr,c_addr;
	int s_fd,c_fd,s_len,c_len;
	if((s_fd=socket(AF_INET,SOCK_STREAM,0))==-1)
		printf("error..socket\n");
	s_addr.sin_family=AF_INET;
	s_addr.sin_addr.s_addr=INADDR_ANY;
	s_addr.sin_port=3452;
	s_len=sizeof(s_addr);
	if(bind(s_fd,(struct sockaddr*)& s_addr,s_len)==-1)
		printf("error bind\n");
	if(listen(s_fd,5)==-1)
		printf("error listen\n");
	while(1)
	{
		c_len=sizeof(c_addr);
		char buff[100];
		if((c_fd=accept(s_fd,(struct sockaddr*)&c_addr,&c_len))==-1)
			printf("erroe accept\n");
		printf("connected\n");
		write(c_fd,"from server..........",100);
		close(c_fd);
	}
	return 0;
}
```

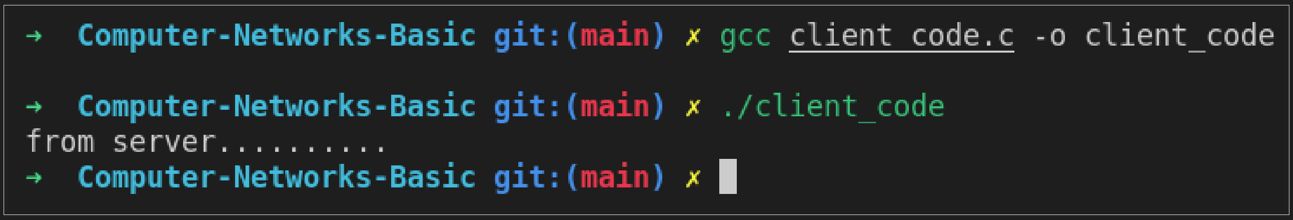
Terminal code and output:



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

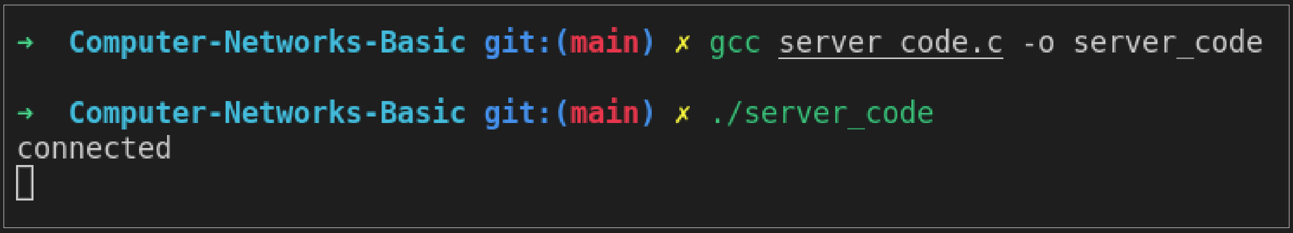
```
➜  Computer-Networks-Basic git:(main) ✗ gcc client_code.c -o client_code

➜  Computer-Networks-Basic git:(main) ✗ ./client_code
from server..........
➜  Computer-Networks-Basic git:(main) ✗ 
```

```
➜  Computer-Networks-Basic git:(main) ✗ gcc server_code.c -o server_code

➜  Computer-Networks-Basic git:(main) ✗ ./server_code
connected
```



```
➜  Computer-Networks-Basic git:(main) ✗ gcc client_code.c -o client_code

➜  Computer-Networks-Basic git:(main) ✗ ./client_code
from server..........
➜  Computer-Networks-Basic git:(main) ✗ 
```

1.client side



```
➜  Computer-Networks-Basic git:(main) ✗ gcc server_code.c -o server_code

➜  Computer-Networks-Basic git:(main) ✗ ./server_code
connected
```

2.server side

Codes and Files;

- ➔ *server_code.c* in github
- ➔ *client_code.c* in github