ASSIGNMENT 6

Client.c

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <string.h>
#include <netinet/in.h>
#include <netdb.h>
#define PORT 999
#define BUF_SIZE 1000
int main(int argc, char **argv)
{
	struct sockaddr_in addr, cl_addr;
	int sockfd, ret;
	char buffer[BUF_SIZE];
	struct hostent * server;
	char * serverAddr;

	if (argc < 2)
	{
		printf("usage: client < ip address >\n");
		exit(1);
	}

	serverAddr = argv[1];
	sockfd = socket(AF_INET, SOCK_STREAM, 0);

	if (sockfd < 0)
	{
		printf("Error creating socket!\n");
		exit(1);
	}

	printf("Socket created...\n");
	memset(&addr, 0, sizeof(addr));
	addr.sin_family = AF_INET;
	addr.sin_addr.s_addr = inet_addr(serverAddr);
	addr.sin_port = PORT;
	ret = connect(sockfd, (struct sockaddr *) &addr, sizeof(addr));

	if (ret < 0)
	{
		printf("Error connecting to the server! : : %d\n",ret);
		exit(1);
	}

	printf("Connected to the server @ %s\n",serverAddr);
	memset(buffer, 0, BUF_SIZE);
```

```
	printf("Enter your message(s): ");

	while (fgets(buffer, BUF_SIZE, stdin) != NULL)
	{
		ret = sendto(sockfd, buffer, BUF_SIZE, 0, (struct sockaddr *) &addr,
		sizeof(addr));
		if (ret < 0)
		{
			printf("Error sending data!\n\t-%s", buffer);
		}
		ret = recvfrom(sockfd, buffer, BUF_SIZE, 0, NULL, NULL);
		if (ret < 0)
		{
			printf("Error receiving data!\n");
		}
		else
		{
			printf("Received: ");
			fputs(buffer, stdout);
			printf("\n");
		}
	}
	return 0;
}
```

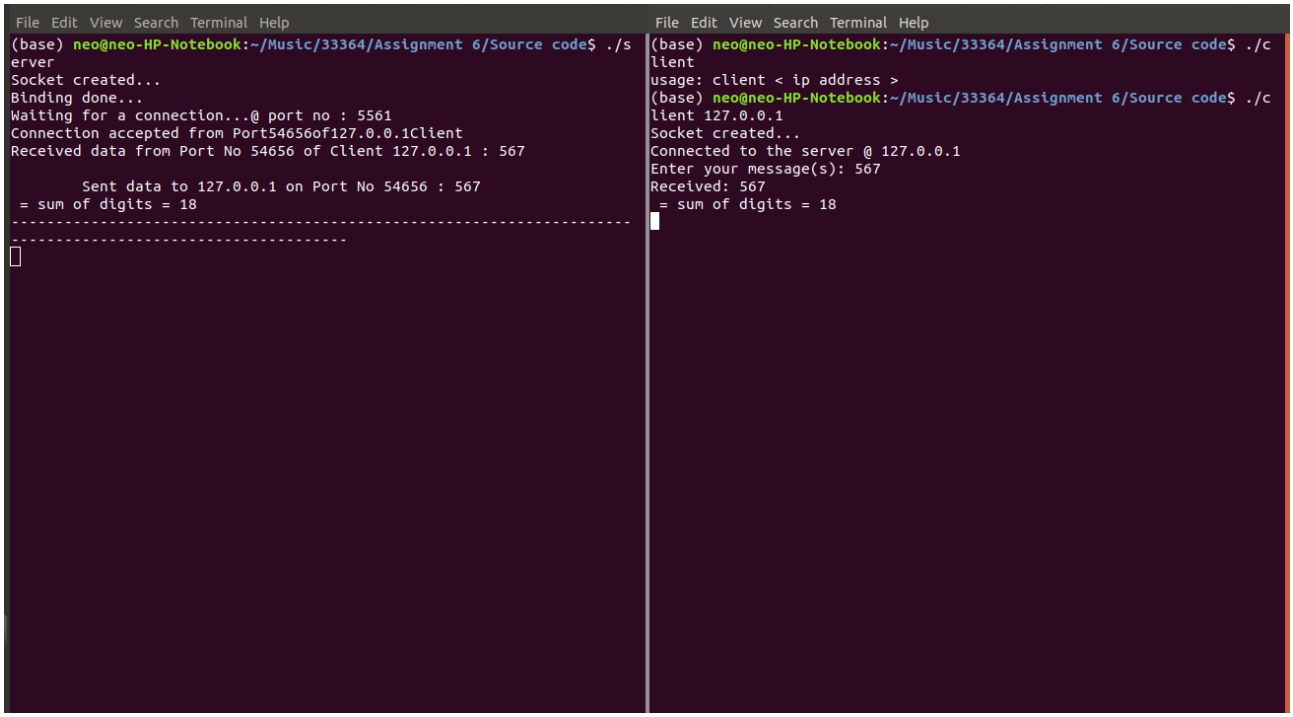
Server.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <errno.h>
#define PORT 9999
#define BUF_SIZE 1000
#define CLADDR_LEN 200
char *itoaa(int val, int base);
int main()
{
	struct sockaddr_in addr, cl_addr;
	int sockfd, len, ret, newsockfd;
	char buffer[BUF_SIZE];
	pid_t childpid;
	char clientAddr[CLADDR_LEN];
	int num, rem, sum;
	char *str;
```

```
sockfd = socket(AF_INET, SOCK_STREAM, 0);

if (sockfd < 0)
{
	printf("Error creating socket!\n");
	exit(1);
}

printf("Socket created...\n");

memset(&addr, 0, sizeof(addr));
addr.sin_family = AF_INET;
addr.sin_addr.s_addr = INADDR_ANY;
addr.sin_port = PORT;
ret = bind(sockfd, (struct sockaddr *) &addr, sizeof(addr));

if (ret < 0)
{
	printf("Error binding!\n");
	exit(1);
}
else
	printf("Binding done...\n");

printf("Waiting for a connection...@ port no : %d\n",PORT );listen(sockfd, 5);

for (;;) //infinite loop
{
	len = sizeof(struct sockaddr_in);
	newsockfd = accept(sockfd, (struct sockaddr*)&cl_addr,(socklen_t *)&len);

	if (newsockfd < 0)
	{
		printf("Error accepting connection!\n");
		exit(1);
	}

	else
		printf("Connection accepted from ");

	inet_ntop(AF_INET, &(cl_addr.sin_addr), clientAddr,CLADDR_LEN);
	printf("Port%dof%sClient\n",ntohs(cl_addr.sin_port),inet_ntoa(cl_addr.sin_addr));
	if ((childpid = fork()) == 0) //creating a child process
	{
		close(sockfd);
		//stop listening for new connections by the main process.
		//the child will continue to listen.
		//the main process now handles the connected client.
		for (;;)
		{
			memset(buffer, 0, BUF_SIZE);
```

```
                                ret = recvfrom(newsockfd, buffer, BUF_SIZE, 0,(struct sockaddr *)
&cl_addr, (socklen_t *)&len);
                                if(ret < 0)
                                {
                                        printf("Error receiving data!\n");
                                        exit(1);
                                }
                                else
                                printf("Received data from Port No %d of Client %s : %s\n ",
ntohs(cl_addr.sin_port),clientAddr, buffer);
                                num=atoi(buffer);
                                sum=0;
                                while(num>0)
                                {
                                        sum = sum + (num % 10);
                                        num = num / 10;
                                }
                                strcat(buffer," = sum of digits = ");
                                str=itoaa(sum,10);
                                strcat(buffer,str);
                                ret = sendto(newsockfd, buffer, BUF_SIZE, 0,(struct sockaddr *)
&cl_addr, len);
                                if (ret < 0)
                                {
                                        printf("Error sending data!\n");
                                        exit(1);
                                }
                                else
                                        printf("\tSent data to %s on Port No %d : %s\n",
clientAddr,ntohs(cl_addr.sin_port), buffer);

                                }
                        }
                        close(newsockfd);
                }
                return(0);
}

char *itoaa(int val, int base)
{
        static char buf[32] = {0};
        int i = 30;
        for( ; val && i ; --i, val /= base)
        buf[i]="0123456789abcdef"[val % base];
        return &buf[i+1];
}
```

OUTPUT:-


File Edit View Search Terminal Help
(base) neo@neo-HP-Notebook:~/Music/33364/Assignment 6/Source code$ ./s
erver
Socket created...
Binding done...
Waiting for a connection...@ port no : 5561
Connection accepted from Port54656of127.0.0.1Client
Received data from Port No 54656 of Client 127.0.0.1 : 567
Sent data to 127.0.0.1 on Port No 54656 : 567
= sum of digits = 18
File Edit View Search Terminal Help
(base) neo@neo-HP-Notebook:~/Music/33364/Assignment 6/Source code$ ./c
lient
usage: client < ip address >
(base) neo@neo-HP-Notebook:~/Music/33364/Assignment 6/Source code$ ./c
lient 127.0.0.1
Socket created...
Connected to the server @ 127.0.0.1
Enter your message(s): 567
Received: 567
= sum of digits = 18