Experiment No. 11

Installing and configure DHCP server and write a program to install the software on remote Machine.

Program for DHCP:

// Program for Server: Server.c

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include <arpa/inet.h>
#include<string.h>
#include<stdlib.h>
int main(int argc, char* argv[])
{
	/*Variables*/
	int sock,i=0;
	struct sockaddr_in server;
	int mysock;
	char buffer[1024],command[1000];
	int rval;
	/*Create Sockets*/
	sock = socket(AF_INET, SOCK_STREAM, 0);
	if(sock<0)
	{
		perror("Failed to create Socket");
		exit(1);
	}
	server.sin_family = AF_INET;
	server.sin_addr.s_addr = INADDR_ANY;
	server.sin_port = htons(5000);
	/*Call Bind*/
	if(bind(sock, (struct sockaddr *)&server, sizeof(server)))
	{
			perror("Bind Failed");
			exit(1);
	}
	/*Listen*/
	listen(sock, 5);
	/*Accept*/
		mysock = accept(sock, (struct sockaddr *) 0, 0);
		if(mysock == -1)
		{
			perror("Accept Failed");
		}
		else
		{
		do
		{
```

```
                        memset(buffer, 0, sizeof(buffer));
                        //Receiving command character by character from the client
                        if((rval = recv(mysock, buffer, sizeof(buffer), 0))<0)
                        {
                                perror("Reading Stream Message error");
                        }
                        else if(rval == 0)
                        {
                                printf("Ending Connection\n");
                                //command[i] = '\0';
                                printf("\nCommand==%s\n",command);
                                //Executing the received command on the server
                                system(command);
                                break;
                        }
                        else
                        {
                                 system("clear");
                                 command[i] = buffer[0];
                                 command[i+1] = '\0';
                                printf("%s\n",command);
                                i++;
                        }
                }while(1);
        }
        close(mysock);
        return 0;
}
```

// Program for Client: Client.c

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <termios.h>
int mygetch( ) {
  struct termios oldt,newt;
  int ch;
  tcgetattr( STDIN_FILENO, &oldt );
  newt = oldt;
  newt.c_lflag &= ~( ICANON | ECHO );
  tcsetattr( STDIN_FILENO, TCSANOW, &newt );
  ch = getchar();
  tcsetattr( STDIN_FILENO, TCSANOW, &oldt );
  return ch;
```

```
}
int main(int argc, char *argv[]){
        int sock;
        struct sockaddr_in server;
        struct hostent *hp;
        char buffer[1024], cbuff[10];
        //Creating Socket
        sock= socket(AF_INET, SOCK_STREAM, 0);
        if(sock<0)
        {
                perror("Socket Failed");
                close(sock);
                exit(1);
        }
        server.sin_family = AF_INET;
        hp = gethostbyname(argv[1]);
        if(hp==0)
        {
                perror("gethostbynme Failed");
                close(sock);
                exit(1);
        }
        memcpy(&server.sin_addr, hp->h_addr, hp->h_length);
        server.sin_port = htons(5000);
if (connect(sock , (struct sockaddr *)&server , sizeof(server)) < 0)
   {
      perror("connect failed. Error");
      return 1;
   }
   puts("Connected\n");
        //Accepting command
        printf("Enter command: \t");
        cbuff[0] = 1;
        do {
                cbuff[0] = mygetch();
                cbuff[1] = '\0';
                if( send(sock , cbuff , strlen(cbuff) , 0) < 0)
      {
         puts("Send failed");
         return 1;
      }
         printf("%s",cbuff);
         }while(cbuff[0] != '\n');
         return 0;
}
```

Output:

```
root@Bhavesh:/home/bhavesh# gcc -o s server.c
root@Bhavesh:/home/bhavesh# ./s
abc.txt         Downloads         nwc.cc~               sc.py
```

```
a.out           examples.desktop  nwemu.cc              server.c


root@Bhavesh:/home/bhavesh# gcc -o c client.c
root@Bhavesh:/home/bhavesh# ./c
root@Bhavesh:/home/bhavesh# ./c 10.10.0.80
Connected

Enter command: ls
```