

```

/* HTTP Client - socket in c code */
#define LINESIZE 4096
#define SERVER_PORT 80 /*default WEB port */
char* Buffer[] = {
    /*
    "GET / HTTP/1.0\n",
    "\n\n",
    "Connection: Keep-Alive\n",
    "User-Agent: Moxilla/2.0 (WinNT; I)\n",
    "Host: chilly:2512\n",
    "Accept: image/gif\n",
    "\n\n",
    */
    "POST /cgi-bin/getset.exe HTTP/1.0\n",
    "Connection: Keep-Alive\n",
    "User-Agent: Mozilla/2.0\n",
    "Accept: text/plain\n",
    "Accept: text/html\n",
    "Content-type: application/x-www-form-urlencoded\n",
    "Content-length: 11\n",
    "\n",
    "name=nobody\n\n",
    NULL
};

int main(argc, argv)
int argc; char* argv[];
{
    struct sockaddr_in server;
    struct hostent *host_info;
    int sock, count;
    int i;
    char inline[LINESIZE];
    char outline[LINESIZE];
    char* server_name;
    /* Get server name from command line,if none use localhost*/
    server_name = (argc > 1) ? argv[1] : "localhost";
    /* Create the socket */
    sock = socket(AF_INET, SOCK_STREAM,0);
    if (sock < 0 ) {
        printf("%s\n", "Socket create problem\n");
        exit(1);
    }

    host_info = gethostbyname(server_name);
    if (host_info == NULL) {
        fprintf(stderr, "%s: unknown host: %s\n", argv[0],
            server_name);
        exit(2);
    }
}

```

Programmers Society Inc

```

/* Set up the server's socket addr */
server.sin_family = host_info->h_addrtype;
memcpy((char*)&server.sin_addr, host_info->h_addr,
        host_info->h_length);
server.sin_port=(argc >2) ? htons( atoi(argv[2]) ) :
htons(SERVER_PORT);

if (connect(sock,(struct sockaddr *)&server, sizeof(server) ) < 0 ) {
    printf("%s", "Connecting to server has problem");
    exit(3);
}

/* OK ! Now send and receive to/from the server */

printf("Connected to the server-> %s\n", server_name);

/* Write the Req to the socket */
i =0;
while ( Buffer[i]) {
    if ( (count = write (sock,Buffer[i], strlen(Buffer[i]))) > 0 )
        i++;
};

sleep(1);

read(sock, inline, 4096);

printf("%s\n", " Now what we got\n");
printf("%s\n", inline);

close(sock);
}

```

