Unix Socket

1. Include libraries

```
#include <netinet/in.h>

#include <stdio.h>

#include <sys/types.h>

#include <string.h>

#include <errno.h>

#include <sys/socket.h>

#include <stdlib.h>

#include <unistd.h>
```

2. This is entry of the program

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

{

    char buf[512];

    struct sockaddr_in server_ad, client_add;

    socklen_t client

    int no, fed, newsf, pnumber;

    if (argc < 2) {

       fprintf(stderr,"port error:\n");

       exit(1);

    }
```

3. Create TCP server socket

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

    if (fed < 0)

          fprintf(stderr,"socket open error");

      error("open error");

    bzero((char *) &serv;_addr, sizeof(server_ad));
```

4. Get Port number to listen and bind

```
    pnumber = atoi(argv[1]);
```

4.1 Set the Server socket for listening the port

```
server_ad.sin_family = AF_INET;

server_ad.sin_addr.s_addr = INADDR_ANY;

server_ad.sin_port = htons(pnumber);
```

4.2 link the port and the socket

```
if (bind(fed, (struct sockaddr *) &serv;_addr,

        sizeof(server_ad)) < 0)

        error("bind error");
```

4.3 start listening

```
listen(fed,4);

clt = sizeof(client_add);
```

5. Accept the client socket

```
newsf = accept(fed,    (struct sockaddr *) &cli;_addr, &clt;);

if (newsf < 0)

    error("error");


printf("get connect server %s port is %d\n",

     inet_ntoa(client_add.sin_addr), ntohs(client_add.sin_port));
```

6. Communicate with each other

```
// This send() function sends the 13 bytes of the string to the new socket

send(newsf, "welcome\n", 13, 0);

bzero(buf,256);

no = read(newsf,buf,255);

//read error checking

if (n < 0) error("read socket error");

printf("this is message: %s\n",buf);
```

7. Close the client socket

```
close(newsf);
```

8. Close the server socket

```
close(fed);

return 0;
```

```
}
```