Unix Socket

## 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>

## 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);

}

## 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;

}