Socket收发文件

1. Client文件

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <arpa/inet.h>

#include <sys/socket.h>

void error\_handling(char \*message);

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

{

int sock;

struct sockaddr\_in serv\_addr;

char message[30];

int str\_len;

if (argc != 3)

{

printf("Usage : %s <IP> <port>\n", argv[0]);

exit(1);

}

//鍒涘缓濂楁帴瀛楋紝姝ゆ椂濂楁帴瀛楀苟涓嶉┈涓婂垎涓烘湇鍔＄鍜屽鎴风銆傚鏋滅揣鎺ョ潃璋冪敤 bind,listen 鍑芥暟锛屽皢鎴愪负鏈嶅姟鍣ㄥ鎺ュ瓧

//濡傛灉璋冪敤 connect 鍑芥暟锛屽皢鎴愪负瀹㈡埛绔鎺ュ瓧

sock = socket(PF\_INET, SOCK\_STREAM, 0);

if (sock == -1)

error\_handling("socket() error");

memset(&serv\_addr, 0, sizeof(serv\_addr));

serv\_addr.sin\_family = AF\_INET;

serv\_addr.sin\_addr.s\_addr = inet\_addr(argv[1]);

serv\_addr.sin\_port = htons(atoi(argv[2]));

//璋冪敤 connect 鍑芥暟鍚戞湇鍔″櫒鍙戦€佽繛鎺ヨ姹? if (connect(sock, (struct sockaddr \*)&serv\_addr, sizeof(serv\_addr)) == -1)

error\_handling("connect() error!");

str\_len = read(sock, message, sizeof(message) - 1);

if (str\_len == -1)

error\_handling("read() error!");

printf("Message from server : %s \n", message);

close(sock);

return 0;

}

void error\_handling(char \*message)

{

fputs(message, stderr);

fputc('\n', stderr);

exit(1);

}

二.Sever文件

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <arpa/inet.h>

#include <sys/socket.h>

void error\_handling(char \*message);

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

{

int serv\_sock;

int clnt\_sock;

struct sockaddr\_in serv\_addr;

struct sockaddr\_in clnt\_addr;

socklen\_t clnt\_addr\_size;

char message[] = "Hello World!";

if (argc != 2)

{

printf("Usage : %s <port>\n", argv[0]);

exit(1);

}

//璋冪敤 socket 鍑芥暟鍒涘缓濂楁帴瀛? serv\_sock = socket(PF\_INET, SOCK\_STREAM, 0);

if (serv\_sock == -1)

error\_handling("socket() error");

memset(&serv\_addr, 0, sizeof(serv\_addr));

serv\_addr.sin\_family = AF\_INET;

serv\_addr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

serv\_addr.sin\_port = htons(atoi(argv[1]));

//璋冪敤 bind 鍑芥暟鍒嗛厤ip鍦板潃鍜岀鍙ｅ彿

if (bind(serv\_sock, (struct sockaddr \*)&serv\_addr, sizeof(serv\_addr)) == -1)

error\_handling("bind() error");

//璋冪敤 listen 鍑芥暟灏嗗鎺ュ瓧杞负鍙帴鍙楄繛鎺ョ姸鎬? if (listen(serv\_sock, 5) == -1)

error\_handling("listen() error");

clnt\_addr\_size = sizeof(clnt\_addr);

//璋冪敤 accept 鍑芥暟鍙楃悊杩炴帴璇锋眰銆傚鏋滃湪娌℃湁杩炴帴璇锋眰鐨勬儏鍐典笅璋冪敤璇ュ嚱鏁帮紝鍒欎笉浼氳繑鍥烇紝鐩村埌鏈夎繛鎺ヨ姹備负姝? clnt\_sock = accept(serv\_sock, (struct sockaddr \*)&clnt\_addr, &clnt\_addr\_size);

if (clnt\_sock == -1)

error\_handling("accept() error");

//绋嶅悗瑕佸皢浠嬬粛鐨?write 鍑芥暟鐢ㄤ簬浼犺緭鏁版嵁锛岃嫢绋嬪簭缁忚繃 accept 杩欎竴琛屾墽琛屽埌鏈锛屽垯璇存槑宸茬粡鏈変簡杩炴帴璇锋眰

write(clnt\_sock, message, sizeof(message));

close(clnt\_sock);

close(serv\_sock);

return 0;

}

void error\_handling(char \*message)

{

fputs(message, stderr);

fputc('\n', stderr);

exit(1);

}

三.文件截图

