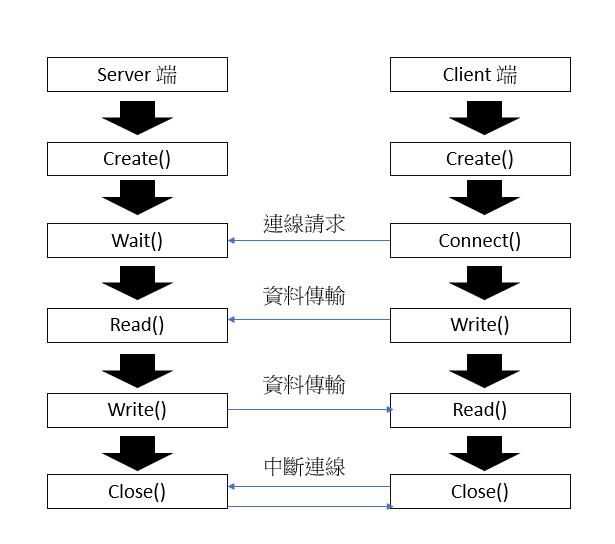
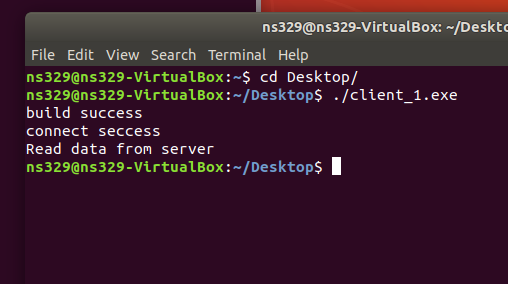
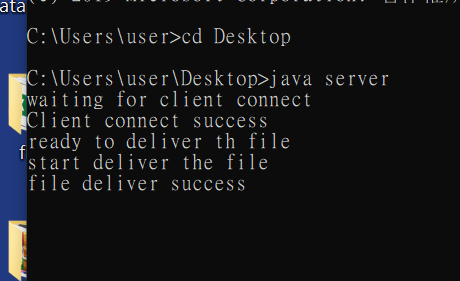
系級:資工三乙

學號；406262333

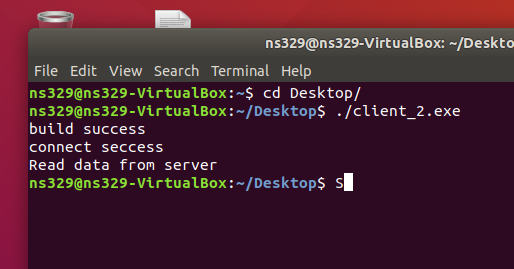
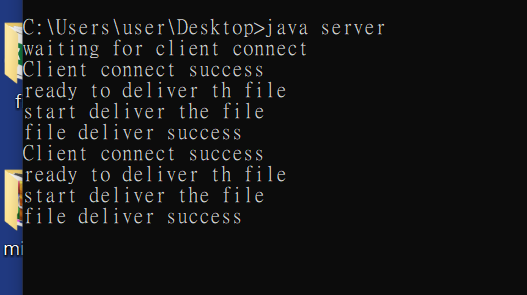
姓名；吳佩臻

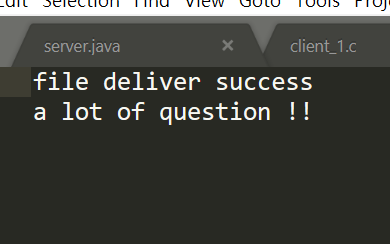
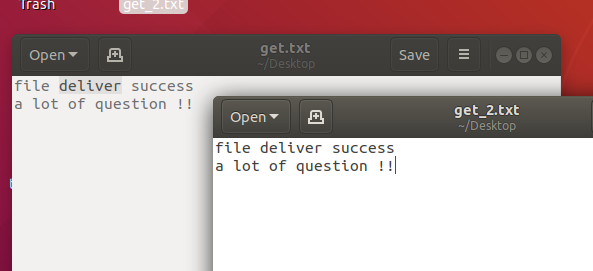
1. 實驗環境:
   * Server 端
     + Windows
     + 使用的語言是Java
     + IPV4 :192.168.56.1
   * Client 端
     + Ubuntu
     + 使用的語言是C
2. 程式碼結構
   * Server 建立後等待Client的連線。
   * Client 建立後對Server發出連線請求
   * 連線成功後，資料傳輸
   * 傳輸完成後，中斷連線



1. 作業截圖
   * 作業一: server 與 Client 連線成功，並傳輸檔案

* 作業二:server 與 多個 Client 連線成功，並且傳輸檔案



* Server 傳輸檔案(deliver\_file.txt) 以及 Client 接收檔案

1. 程式碼
   * Server (JAVA)
2. import java.io.\*;
3. import java.net.\*;
4. public class server{
5. public static void main(String[] args) throws Exception{
6. File file = new File("deliver\_file.txt");
7. int port = 4444;
8. System.out.println("waiting for client connect");
9. // create server
10. ServerSocket ss = new ServerSocket(port);
11. // connect client
12. Socket client = ss.accept();
13. System.out.println("Client connect success");
14. // 數據封裝
15. System.out.println("ready to deliver th file");
16. OutputStream netOut = client.getOutputStream();
17. OutputStream temp = new DataOutputStream(new BufferedOutputStream(netOut));
18. // 建立緩衝區
19. System.out.println("start deliver the file");
20. byte[] buf = new byte[2048];
21. FileInputStream fos = new FileInputStream(file);
22. int num = fos.read(buf);
23. while(num != -1)
24. {
25. temp.write(buf,0,num);// 把檔案寫到緩衝區內
26. temp.flush();// 寫往客戶端
27. num = fos.read(buf);
28. }
29. System.out.println("file deliver success");
30. fos.close();
31. temp.close();
32. }
33. }

* Client\_1 (C)

1. #include<stdio.h>
2. #include<string.h>
3. #include<stdlib.h>
4. #include<sys/types.h>
5. #include<netinet/in.h>
6. #include<arpa/inet.h>
7. #include<unistd.h>
8. #include<sys/socket.h>
9. #define server\_IP "192.168.56.1"
10. #define target\_PORT 4444
11. void connect\_to\_server(char \*ip,int port,int socketfd)
12. {
13. struct sockaddr\_in info;
14. //初始化，將struct涵蓋的bits設為0
15. bzero(&info,sizeof(info));
16. //BSD是AF, POSIX是PF
17. //sockaddr\_in為Ipv4結構
18. info.sin\_family = PF\_INET;
19. //IP address setting
20. info.sin\_addr.s\_addr = inet\_addr(ip);
21. //將port(int)轉換成port(network\_short\_int
22. info.sin\_port = htons(port);
23. int error = connect(socketfd,(struct sockaddr \*)&info,sizeof(info));
25. if(error == -1)
26. {
27. printf("connect error\n");
28. }
29. else
30. {
31. printf("connect seccess\n");
32. }
33. }
34. void get\_file(char \*FILENAME, int socketid)
35. {
36. int len = 0;
37. //接收資訊buffer
38. char text[500];
39. printf("Read data from server\n");
40. recv(socketid,text,sizeof(text),0);
41. FILE \*fp = fopen(FILENAME,"w");
42. fprintf(fp,"%s",text);
43. close(socketid);
44. fclose(fp);
46. }
47. int main(){
48. //create a client
49. int socket\_ = 0;
50. socket\_ = socket(AF\_INET,SOCK\_STREAM,0);
52. if(socket\_==-1)
53. {
54. printf("build failed\n");
55. }
56. else
57. {
58. printf("build success\n");
59. }
60. //socket的連線
61. connect\_to\_server(server\_IP,target\_PORT,socket\_);
62. // 從server載檔案
63. get\_file("get\_1.txt",socket\_);
64. return 0 ;
65. }

* Client\_2 (C)

1. #include<stdio.h>
2. #include<string.h>
3. #include<stdlib.h>
4. #include<sys/types.h>
5. #include<netinet/in.h>
6. #include<arpa/inet.h>
7. #include<unistd.h>
8. #include<sys/socket.h>
9. #define server\_IP "192.168.56.1"
10. #define target\_PORT 4444
11. void connect\_to\_server(char \*ip,int port,int socketfd)
12. {
13. struct sockaddr\_in info;
14. //初始化，將struct涵蓋的bits設為0
15. bzero(&info,sizeof(info));
16. //BSD是AF, POSIX是PF
17. //sockaddr\_in為Ipv4結構
18. info.sin\_family = PF\_INET;
19. //IP address setting
20. info.sin\_addr.s\_addr = inet\_addr(ip);
21. //將port(int)轉換成port(network\_short\_int
22. info.sin\_port = htons(port);
23. int error = connect(socketfd,(struct sockaddr \*)&info,sizeof(info));
25. if(error == -1)
26. {
27. printf("connect error\n");
28. }
29. else
30. {
31. printf("connect seccess\n");
32. }
33. }
34. void get\_file(char \*FILENAME, int socketid)
35. {
36. int len = 0;
37. //接收資訊buffer
38. char text[500];
39. printf("Read data from server\n");
40. recv(socketid,text,sizeof(text),0);
41. FILE \*fp = fopen(FILENAME,"w");
42. fprintf(fp,"%s",text);
43. close(socketid);
44. fclose(fp);
46. }
47. int main(){
48. //create a client
49. int socket\_ = 0;
50. socket\_ = socket(AF\_INET,SOCK\_STREAM,0);
52. if(socket\_==-1)
53. {
54. printf("build failed\n");
55. }
56. else
57. {
58. printf("build success\n");
59. }
60. //socket的連線
61. connect\_to\_server(server\_IP,target\_PORT,socket\_);
62. // 從server載檔案
63. get\_file("get\_2.txt",socket\_);
64. return 0 ;
65. }