**實驗環境：於Linux系統中，並使用127.0.0.1於本機利用c語言進行socket單機互傳。**

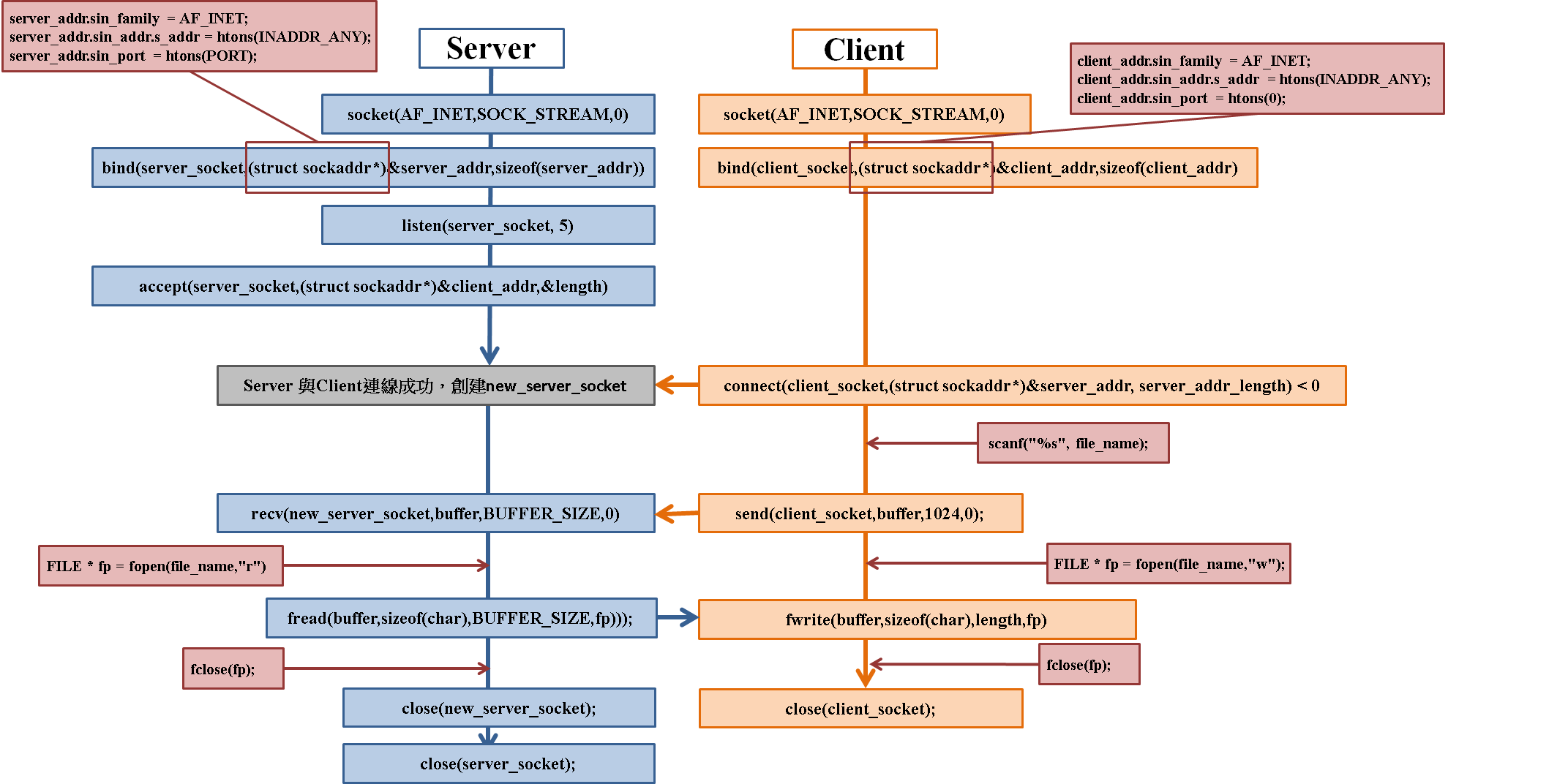
1. **Client / Srever程式(1個Client對1個Server)(30%)**
2. **程式碼**
3. **Server端**

|  |
| --- |
| 1. **#include <netinet/in.h>** 2. **#include <sys/types.h>** 3. **#include <sys/socket.h>** 4. **#include <stdio.h>** 5. **#include <stdlib.h>** 6. **#include <string.h>** 7. **#include <unistd.h>** 8. **#include <arpa/inet.h>** 9. **#define PORT 7777** 10. **#define BUFFER\_SIZE 1024** 11. **int main(int argc, char \*\*argv)** 12. **{** 13. **// 宣告一個server\_addr來設定server socket的資訊** 14. **struct sockaddr\_in server\_addr;** 15. **// 把一段記憶體區的內容先全部設置為0** 16. **bzero(&server\_addr,sizeof(server\_addr));** 17. **// 設置成IPv4** 18. **server\_addr.sin\_family = AF\_INET;** 19. **// 地址設置成0.0.0.0，即任意的地址** 20. **server\_addr.sin\_addr.s\_addr = htons(INADDR\_ANY);** 21. **// 設置port number為7777** 22. **server\_addr.sin\_port = htons(PORT);** 24. **// 創建一個socket，使用IPv4、TCP** 25. **int server\_socket = socket(AF\_INET,SOCK\_STREAM,0);** 26. **// 檢查 socket 是否有創建失敗** 27. **if( server\_socket < 0)** 28. **{** 29. **printf("[-]Create socket failed!");** 30. **exit(1);** 31. **}** 33. **// 將設定好的address及port number設定置創建好的socket，錯誤回傳-1** 34. **if( bind(server\_socket,(struct sockaddr\*)&server\_addr,sizeof(server\_addr)))** 35. **{** 36. **printf("[-]Server bind port : %d failed!", PORT);** 37. **exit(1);** 38. **}** 39. **// 監聽server socket有無連線請求，最多5個client能連線，錯誤回傳-1** 40. **if ( listen(server\_socket, 5) )** 41. **{** 42. **printf("[-]Server listen failed!");** 43. **exit(1);** 44. **}** 45. **// 無限迴圈，使server socket能一直運作** 46. **while (1)** 47. **{** 48. **// 宣告一個client\_addr，來存放client端的資訊** 49. **struct sockaddr\_in client\_addr;** 50. **socklen\_t length = sizeof(client\_addr);** 51. **// 接收client端的連線，並產出新的socket，專與此client溝通** 52. **int new\_server\_socket = accept(server\_socket,(struct sockaddr\*)&client\_addr,&length);** 53. **// 若accept返回-1即代表錯誤** 54. **if ( new\_server\_socket < 0)** 55. **{** 56. **printf("[-]Server accept failed!\n");** 57. **break;** 58. **}** 59. **printf("[+]Client connect successfully!");** 60. **// 創建一個1024格空間的buffer array，並將其清成0** 61. **char buffer[1024];** 62. **bzero(buffer, 1024);** 63. **// server socket將收到的data放進buffer裡面，並回傳收到的位元組數** 64. **length = recv(new\_server\_socket,buffer,BUFFER\_SIZE,0);** 65. **// 若回傳的位元組數小於0則錯誤** 66. **if (length < 0)** 67. **{** 68. **printf("Server recieve data failed!\n");** 69. **break;** 70. **}** 71. **// 創建一個1024大小的array，用來存放client想要下載的檔案名稱** 72. **char file\_name[1024];** 73. **// 將array清空** 74. **bzero(file\_name, 1024);** 75. **// 將剛剛收到後放在buffer的檔案名稱複製進入file\_name array** 76. **strncpy(file\_name, buffer, 1024);** 77. **// 創建指向要讓client端下載的檔案指標** 78. **FILE \* fp = fopen(file\_name,"r");** 79. **// 若該指標為NULL，即代表該檔案不存在** 80. **if(fp == NULL)** 81. **{** 82. **printf("File: %s not found\n", file\_name);** 83. **}** 84. **else** 85. **{** 86. **// 若檔案存在，就將buffer清空，準備傳送給client端** 87. **bzero(buffer, BUFFER\_SIZE);** 88. **// 宣告一個整數型態的變數，用來接收fread回傳值** 89. **int file\_block\_length = 0;** 90. **// fread的回傳值為成功有效的讀取fp指向的檔案的元素個數** 91. **while( (file\_block\_length = fread(buffer,sizeof(char),BUFFER\_SIZE,fp))>0)** 92. **{** 93. **// 若元素個數至少有一個即開始傳送** 94. **// 發送buffer中的字串至new\_server\_socket** 95. **if(send(new\_server\_socket,buffer,file\_block\_length,0)<0)** 96. **{** 97. **printf("Send file: %s failed\n", file\_name);** 98. **break;** 99. **}** 100. **// 將buffer清空** 101. **bzero(buffer, BUFFER\_SIZE);** 102. **}** 103. **// 關閉指向要傳輸檔案的指標** 104. **fclose(fp);** 105. **printf("File: %s transfer finished\n",file\_name);** 106. **}** 107. **// 關閉與client 端socket的連接** 108. **close(new\_server\_socket);** 109. **}** 110. **// 關閉server端監聽所使用的socket** 111. **close(server\_socket);** 112. **return 0;** 113. **}** |

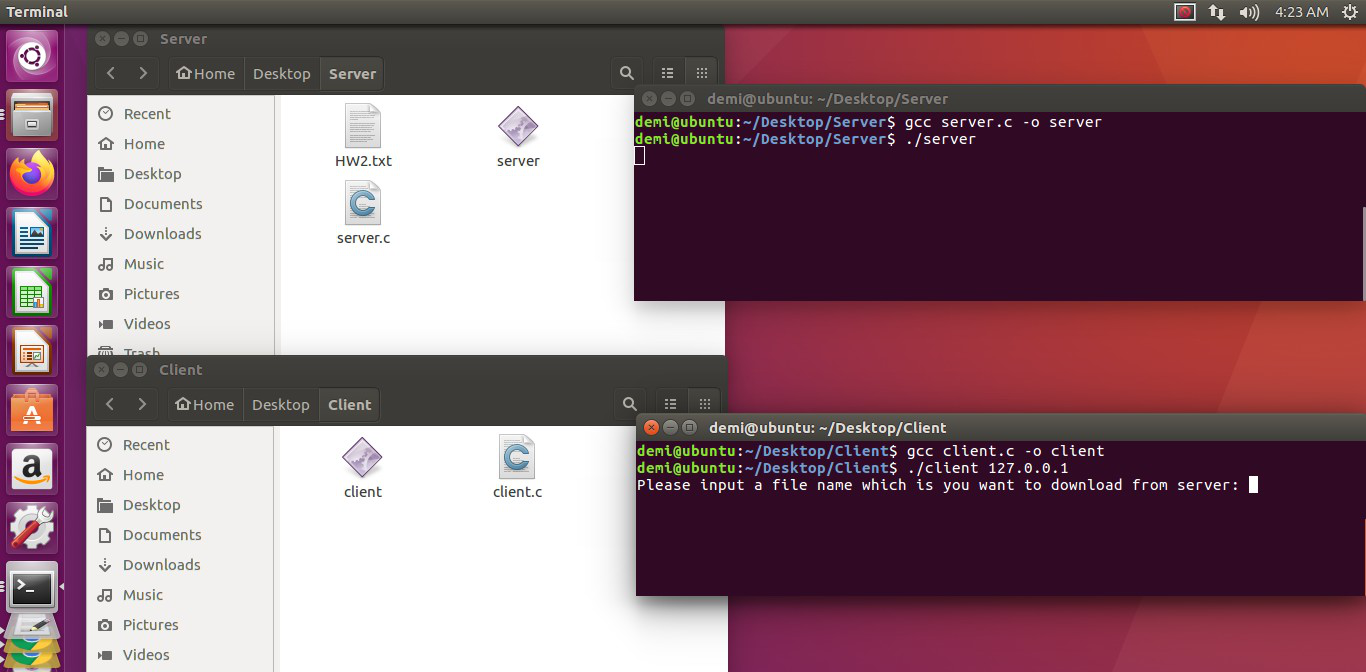
1. **Client端**

|  |
| --- |
| * + - 1. **#include <netinet/in.h>**       2. **#include <sys/types.h>**       3. **#include <sys/socket.h>**       4. **#include <stdio.h>**       5. **#include <stdlib.h>**       6. **#include <string.h>**       7. **#include <unistd.h>**       8. **#include <arpa/inet.h>**       9. **#define PORT 7777**       10. **#define BUFFER\_SIZE 1024**       11. **int main(int argc, char \*\*argv)**       12. **{**       13. **// 宣告一個client\_addr來設定client socket的資訊**       14. **struct sockaddr\_in client\_addr;**       15. **// 把一段記憶體區的內容先全部設置為0**       16. **bzero(&client\_addr,sizeof(client\_addr));**       17. **// 設置成IPv4**       18. **client\_addr.sin\_family = AF\_INET;**       19. **// 將地址設置為0.0.0.0，即任意位置**       20. **client\_addr.sin\_addr.s\_addr = htons(INADDR\_ANY);**       21. **// 0表示讓系統自動分配一個port number**       22. **client\_addr.sin\_port = htons(0);**       23. **// 創建一個socket，使用IPv4、TCP**       24. **int client\_socket = socket(AF\_INET,SOCK\_STREAM,0);**       25. **// 檢查 socket 是否有創建失敗**       26. **if( client\_socket < 0)**       27. **{**       28. **printf("[-]Create Socket Failed!\n");**       29. **exit(1);**       30. **}**       31. **// 將設定好的address及port number設定置創建好的socket，錯誤回傳-1**       32. **if( bind(client\_socket,(struct sockaddr\*)&client\_addr,sizeof(client\_addr)))**       33. **{**       34. **printf("[-]Client Bind Port Failed!\n");**       35. **exit(1);**       36. **}**       37. **// 宣告一個server\_addr來設定server socket的資訊**       38. **struct sockaddr\_in server\_addr;**       39. **// 把一段記憶體區的內容先全部設置為0**       40. **bzero(&server\_addr,sizeof(server\_addr));**       41. **// 設置成IPv4**       42. **server\_addr.sin\_family = AF\_INET;**       43. **// 將輸入在終端機的字串轉成數字後，將之設為伺服器的address**       44. **if(inet\_aton(argv[1],&server\_addr.sin\_addr) == 0)**       45. **{**       46. **printf("[-]Server IP Address Error!\n");**       47. **exit(1);**       48. **}**       49. **// 將server port number設置成7777**       50. **server\_addr.sin\_port = htons(PORT);**       51. **// 取得server\_addr的大小**       52. **socklen\_t server\_addr\_length = sizeof(server\_addr);**       53. **//向server發起連接請求，若錯誤則回傳-1**       54. **if(connect(client\_socket,(struct sockaddr\*)&server\_addr, server\_addr\_length) < 0)**       55. **{**       56. **printf("[-]Can Not Connect To %s!\n",argv[1]);**       57. **exit(1);**       58. **}**       59. **// 創建放置檔案名稱的array**       60. **char file\_name[1024];**       61. **bzero(file\_name, 1024);**       62. **// 要求client端輸入要從server端要載的檔案名稱**       63. **printf("Please input a file name which is you want to download from server: ");**       64. **scanf("%s", file\_name);**       65. **// 創建一個大小為1024的client端buffer，並將其清空**       66. **char buffer[1024];**       67. **bzero(buffer,1024);**       68. **// 將client端輸入的file\_name放進buffer中**       69. **strncpy(buffer, file\_name, 1024);**       70. **//將buffer中的資料送出去給server**       71. **send(client\_socket,buffer,1024,0);**       72. **// 創建一個指向下載的檔案名稱的指標，並創建此檔案**       73. **FILE \* fp = fopen(file\_name,"w");**       74. **// 若該指標為NULL，則無法打開此檔寫入，即錯誤**       75. **if(fp == NULL)**       76. **{**       77. **printf("File: %s Can Not Open To Write\n", file\_name);**       78. **exit(1);**       79. **}**       80. **// 清空buffer**       81. **bzero(buffer,BUFFER\_SIZE);**       82. **int length = 0;**       83. **// client socket將收到的data放進buffer裡面，並回傳收到的位元組數**       84. **while( length = recv(client\_socket,buffer,BUFFER\_SIZE,0))**       85. **{**       86. **// 若回傳的位元數組小於0，則代表錯誤**       87. **if(length < 0)**       88. **{**       89. **printf("Recieve Data From Server %s Failed!\n", argv[1]);**       90. **break;**       91. **}**       92. **// fwrite的回傳值為成功有效的讀取fp指向的檔案的元素個數**       93. **int write\_length = fwrite(buffer,sizeof(char),length,fp);**       94. **// 若fwrite取得的回傳的元素個數小於recv得到的即錯誤**       95. **if (write\_length<length)**       96. **{**       97. **printf("File:\t%s Write Failed\n", file\_name);**       98. **break;**       99. **}**       100. **// 清空buffer**       101. **bzero(buffer,BUFFER\_SIZE);**       102. **}**       103. **printf("Recieve file: %s from server[%s] finished\n",file\_name, argv[1]);**       104. **// 完成將從server端下載的資料寫入檔案中後，關閉指向檔案的指標**       105. **fclose(fp);**       106. **//關閉client端socket**       107. **close(client\_socket);**       108. **return 0;**       109. **}** |

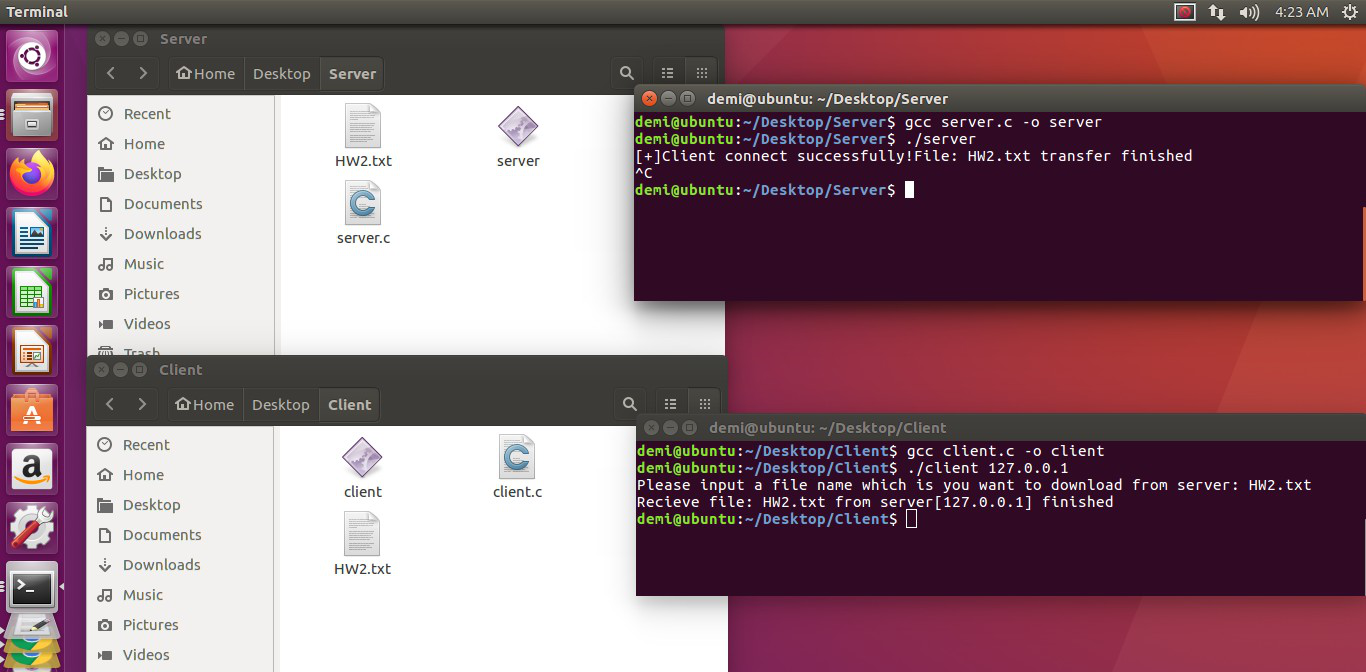
1. **架構流程圖**



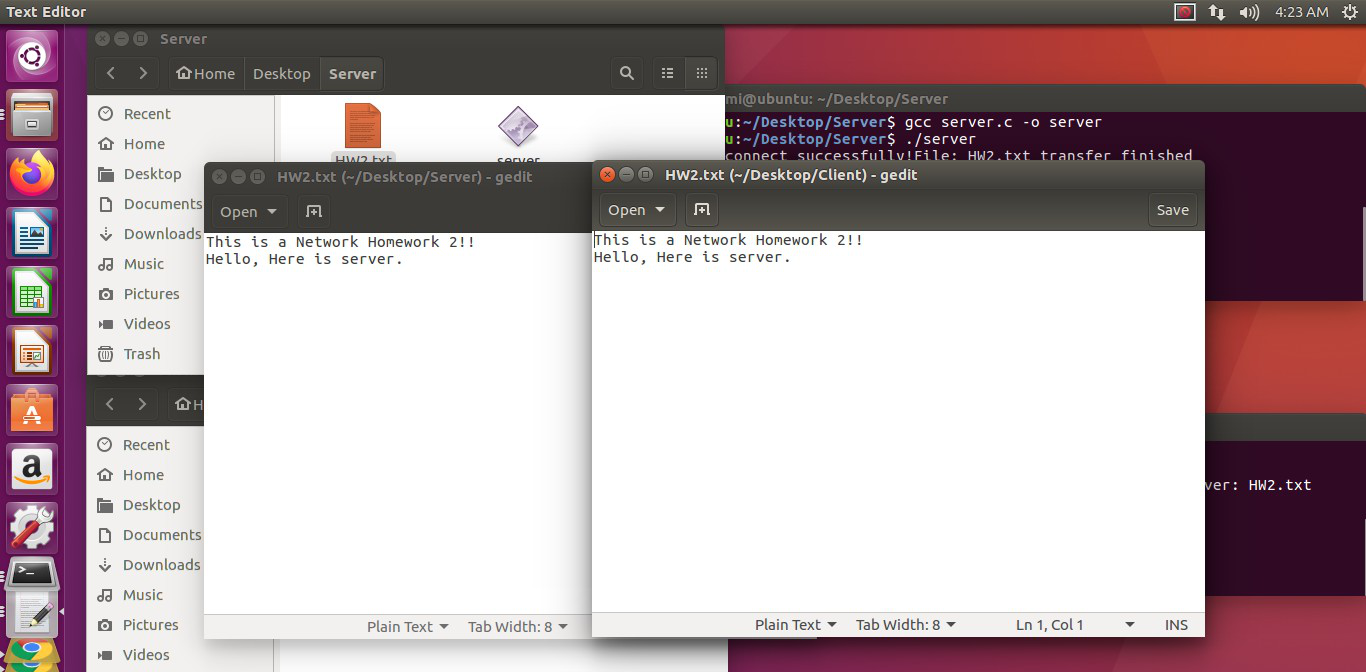
1. **實驗過程、內容及結果**



**圖(1)：我將server及client分至於兩個資料夾，如同兩台電腦般。並開啟兩個終端機各自編譯後執行。另外在server的資料夾中有準備讓client端下載的檔案HW2.txt，且在client端還未下載的時候其並未有此檔。**



**圖(2)：client端輸入要下載的檔案名稱HW2.txt，即在client的資料夾中，多出了一個從server端下載的HW2.txt檔案，並在server端顯示傳輸成功，在client端顯示接收成功。接收完檔案之後client端即關閉socket，server端也輸入Ctrl+C關閉。**



**圖(3)：打開server端和client端各自的HW2.txt檔案驗證，傳輸內容相同。下載成功！**

1. **Client/Srever程式(多個Client對1個Server)(30%)**

**程式碼**

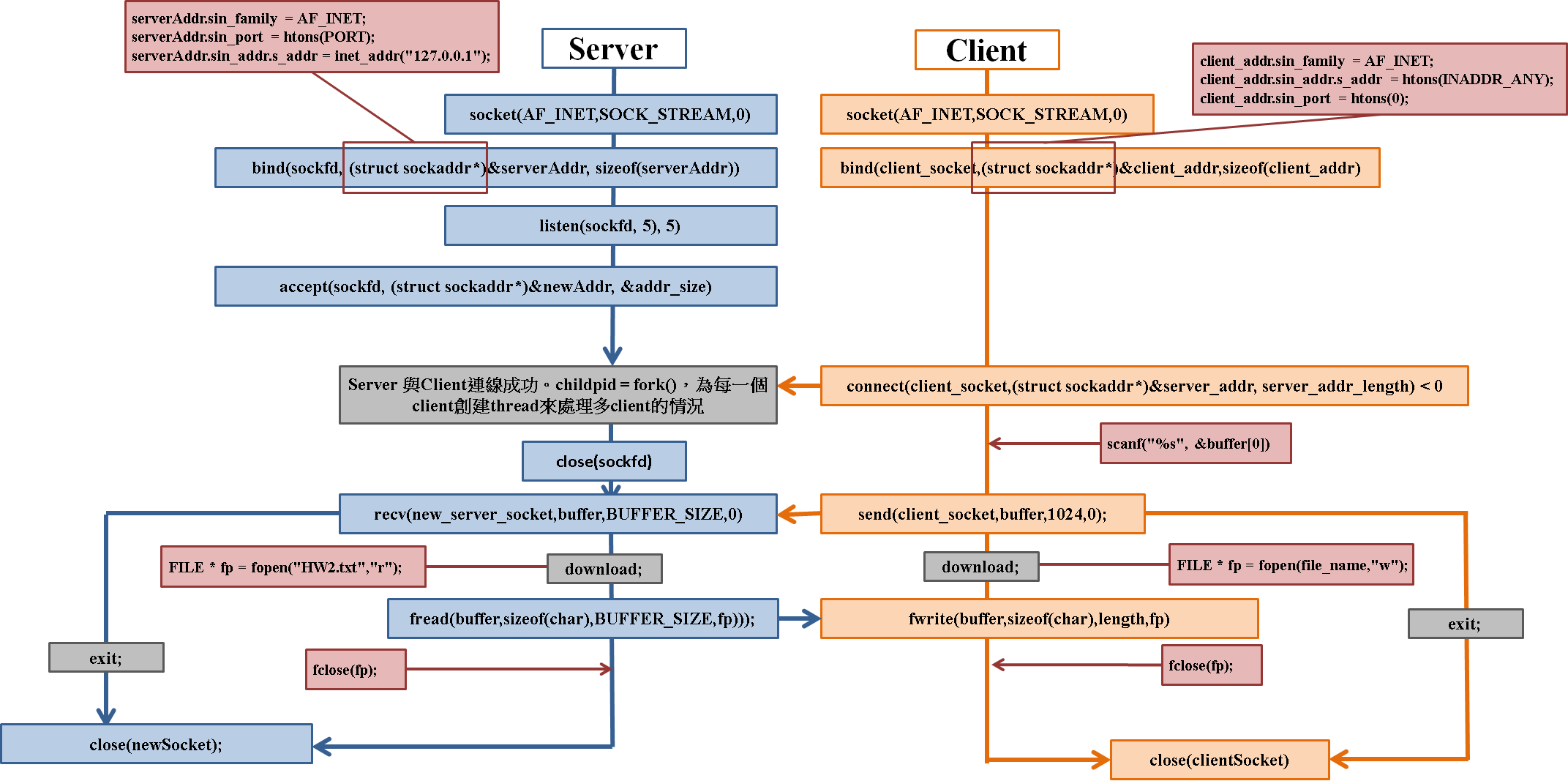
**Server端**

|  |
| --- |
| * 1. **#include <stdio.h>**   2. **#include <stdlib.h>**   3. **#include <string.h>**   4. **#include <unistd.h>**   5. **#include <sys/socket.h>**   6. **#include <sys/types.h>**   7. **#include <netinet/in.h>**   8. **#include <arpa/inet.h>**   9. **#define PORT 4444**   10. **#define HELLO\_WORLD\_SERVER\_PORT 6666**   11. **#define LENGTH\_OF\_LISTEN\_QUEUE 20**   12. **#define BUFFER\_SIZE 1024**   13. **#define FILE\_NAME\_MAX\_SIZE 512**   14. **int main()**   15. **{**   16. **int sockfd, ret;**   17. **// 宣告一個serverAddr來設定server socket的資訊**   18. **struct sockaddr\_in serverAddr;**   19. **int newSocket;**   20. **struct sockaddr\_in newAddr;**   21. **socklen\_t addr\_size;**   22. **char buffer[1024];**   23. **pid\_t childpid;**   24. **// 創建一個socket，使用IPv4、TCP**   25. **sockfd = socket(AF\_INET, SOCK\_STREAM, 0);**   26. **// 若回傳值小於0，即創建錯誤**   27. **if(sockfd < 0){**   28. **printf("[-]Error in connection.\n");**   29. **exit(1);**   30. **}**   31. **printf("[+]Server Socket is created.\n");**   32. **// 先將serverAddr清空**   33. **memset(&serverAddr, '\0', sizeof(serverAddr));**   34. **// 設定成IPv4**   35. **serverAddr.sin\_family = AF\_INET;**   36. **// 將server port number設為4444**   37. **serverAddr.sin\_port = htons(PORT);**   38. **// 將server address設置成loopback address 127.0.0.1**   39. **serverAddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");**   40. **// 將設定好的server address和port number等資訊綁在server socket上**   41. **ret = bind(sockfd, (struct sockaddr\*)&serverAddr, sizeof(serverAddr));**   42. **// 若bind回傳值小於0的時候即代表發生錯誤**   43. **if(ret < 0){**   44. **printf("[-]Error in binding.\n");**   45. **exit(1);**   46. **}**   47. **printf("[+]Bind to port %d\n", 4444);**   48. **// server socket監聽是否有連線請求，且規定最多有五個client連至server**   49. **if(listen(sockfd, 5) == 0)**   50. **{**   51. **printf("[+]Listening....\n");**   52. **}**   53. **else**   54. **{**   55. **printf("[-]Error in binding.\n");**   56. **}**   57. **// 無限迴圈，使server socket能一直運作**   58. **int num = 1;**   59. **while(1){**   60. **// 接收client端的連線，並產出新的socket，專與此client溝通**   61. **newSocket = accept(sockfd, (struct sockaddr\*)&newAddr, &addr\_size);**   62. **// 若 accept 回傳的數值小於0，即server socket 接收連線請求發生錯誤**   63. **if(newSocket < 0){**   64. **exit(1);**   65. **}**   66. **printf("Client %d connection accepted from %s:%d\n", num++, inet\_ntoa(newAddr.sin\_addr), ntohs(newAddr.sin\_port));**   67. **// fork 將當前的 process分支出另一個 process，並回傳子程序的pid 0**   68. **if((childpid = fork()) == 0){**   69. **close(sockfd);**   70. **while(1){**   71. **// 從newSocket中取得client端輸入的字串並放置於buffer**   72. **recv(newSocket, buffer, 1024, 0);**   73. **// 比較buffer和字串exit;，若回傳值為0，則兩者字串相同**   74. **if(strcmp(buffer, "exit;") == 0){**   75. **// 若client輸入的是exit;，則break結束與該client的連線**   76. **printf("Disconnected from %s:%d\n", inet\_ntoa(newAddr.sin\_addr), ntohs(newAddr.sin\_port));**   77. **break;**   78. **}**   79. **// 比較buffer和字串download;，若回傳值為0則兩者字串相同**   80. **else if(strcmp(buffer, "download;") == 0){**   81. **// 創建一個指向要下載檔案的指標**   82. **FILE \* fp = fopen("HW2.txt","r");**   83. **// 若fp回傳為等於NULL，即為找不到此檔**   84. **if(fp == NULL)**   85. **{**   86. **printf("File:HW2.txt Not Found\n");**   87. **}**   88. **else**   89. **{**   90. **// 將buffer清空**   91. **bzero(buffer, 1024);**   92. **// 宣告一個整數型態的變數，用來接收fread回傳值**   93. **int file\_block\_length = 0;**   94. **// fread的回傳值為成功有效的讀取fp指向的檔案的元素個數**   95. **while( (file\_block\_length = fread(buffer,sizeof(char),1024,fp))>0)**   96. **{**   97. **if(send(newSocket,buffer,file\_block\_length,0)<0)**   98. **{**   99. **// 若元素個數至少有一個即開始傳送**   100. **// 發送buffer中的字串至newSocket**   101. **// 若將buffer的東西傳送出去回傳的數值小於 0，即發生錯誤**   102. **printf("Send File:HW2.txt Failed\n");**   103. **break;**   104. **}**   105. **bzero(buffer, 1024);**   106. **}**   107. **// 關閉指向要寫入的檔案的檔案指標**   108. **fclose(fp);**   109. **printf("File:HW2 Transfer Finished\n");**   110. **break;**   111. **}**   112. **}**   113. **}**   114. **}**   115. **}**   116. **// 關閉用來與該client溝通及處理的socket**   117. **close(newSocket);**   118. **return 0;**   119. **}** |

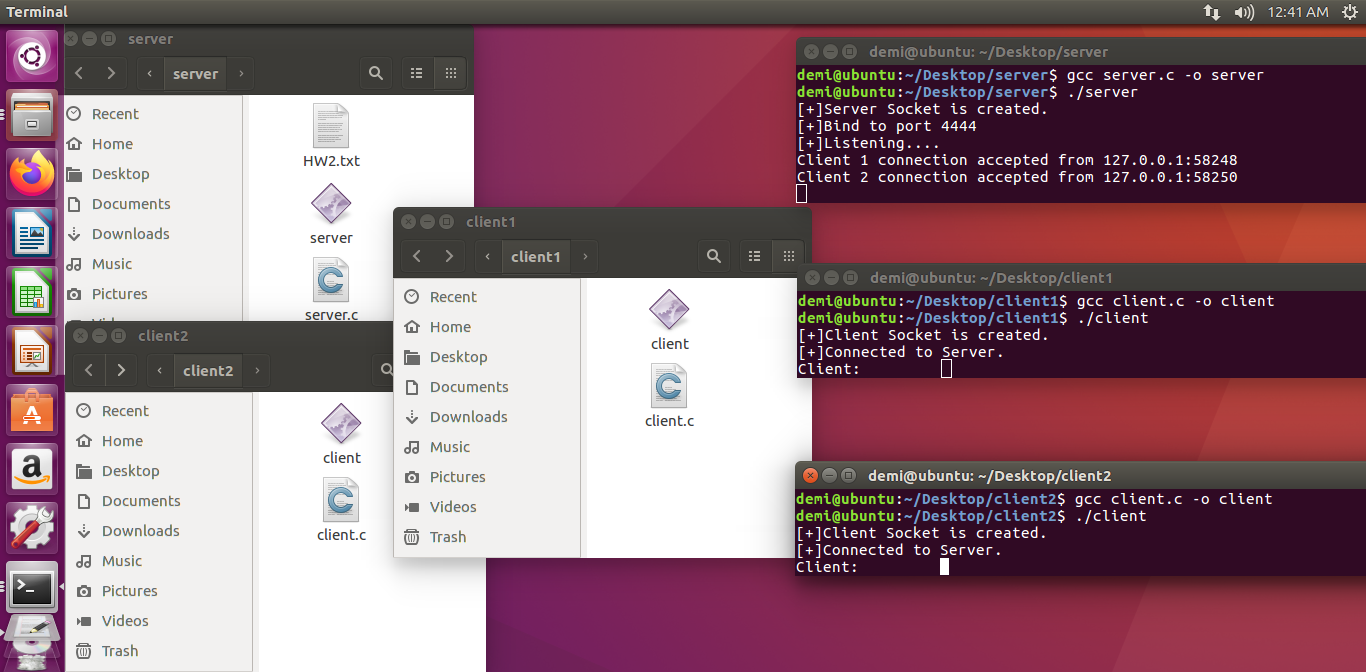
**Client端**

|  |
| --- |
| * + - 1. **#include <stdio.h>**       2. **#include <stdlib.h>**       3. **#include <string.h>**       4. **#include <unistd.h>**       5. **#include <sys/socket.h>**       6. **#include <sys/types.h>**       7. **#include <netinet/in.h>**       8. **#include <arpa/inet.h>**       9. **#define PORT 4444**       10. **#define HELLO\_WORLD\_SERVER\_PORT 6666**       11. **#define BUFFER\_SIZE 1024**       12. **#define FILE\_NAME\_MAX\_SIZE 512**       13. **int main(){**       14. **int clientSocket, ret;**       15. **// 宣告一個serverAddr來設定serversocket的資訊**       16. **struct sockaddr\_in serverAddr;**       17. **// 宣告一個大小為 1024 的陣列buffer**       18. **char buffer[1024];**       19. **// Client端創建一個socket**       20. **clientSocket = socket(AF\_INET, SOCK\_STREAM, 0);**       21. **// 若創建的回傳值小於0即發生錯誤**       22. **if(clientSocket < 0){**       23. **printf("[-]Error in connection.\n");**       24. **exit(1);**       25. **}**       26. **printf("[+]Client Socket is created.\n");**       27. **// 清空serverAddr**       28. **memset(&serverAddr, '\0', sizeof(serverAddr));**       29. **// 設定成IPv4**       30. **serverAddr.sin\_family = AF\_INET;**       31. **// 將要連線的server port number設定成4444**       32. **serverAddr.sin\_port = htons(PORT);**       33. **// 將要連線的server IP address設定成loopback address 127.0.0.1**       34. **serverAddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");**       35. **// Client端發起連線至Server端**       36. **ret = connect(clientSocket, (struct sockaddr\*)&serverAddr, sizeof(serverAddr));**       37. **// 若connect的回傳值小於0即發生錯誤**       38. **if(ret < 0){**       39. **printf("[-]Error in connection.\n");**       40. **exit(1);**       41. **}**       42. **printf("[+]Connected to Server.\n");**       43. **printf("Client: \t");**       44. **// Client端輸入要執行的動作，此輸入放進buffer中**       45. **scanf("%s", &buffer[0]);**       46. **// 將buffer的內容透過clientSocket送出去**       47. **send(clientSocket, buffer, strlen(buffer), 0);**       48. **// 若使用者輸入的是exit的話就直接斷掉與server的連線**       49. **if(strcmp(buffer, "exit;") == 0){**       50. **// 關閉client端的socket**       51. **close(clientSocket);**       52. **printf("[-]Disconnected from server.\n");**       53. **// 結束應用程式**       54. **exit(1);**       55. **}**       56. **// 若使用者輸入的是download;，則開始下載檔案**       57. **if(strcmp(buffer, "download;") == 0){**       58. **printf("Downloading.....\n");**       59. **// 創建指向要寫入的檔案的指標，並創建此檔案**       60. **FILE \* fp = fopen("HW2.txt","w");**       61. **// 若此fp為NULL，即發生錯誤，無法打開檔案寫入**       62. **if(fp == NULL)**       63. **{**       64. **printf("File:HW2.txt Can Not Open To Write\n");**       65. **exit(1);**       66. **}**       67. **// 將buffer清空**       68. **bzero(buffer,1024);**       69. **// 創建存放recv回傳值的變數**       70. **int length = 0;**       71. **// 透過clientSocket從獲取資料放進buffer中，並回傳收到的位元組數**       72. **while( length = recv(clientSocket,buffer,1024,0))**       73. **{**       74. **// 若收到的位元組數小於0即發生錯誤**       75. **if(length < 0)**       76. **{**       77. **printf("Recieve Data From Server Failed!\n");**       78. **break;**       79. **}**       80. **// fwrite的回傳值為成功有效的讀取fp指向的檔案的元素個數**       81. **int write\_length = fwrite(buffer,sizeof(char),length,fp);**       82. **// 若fwrite回傳值小於從socket取得的資料數量即發生錯誤**       83. **if (write\_length<length)**       84. **{**       85. **printf("File:HW2.txt Write Failed\n");**       86. **break;**       87. **}**       88. **// 清空buffer**       89. **bzero(buffer,1024);**       90. **}**       91. **printf("Recieve File:HW2.txt From Server Finished\n");**       92. **// 關閉指向要寫入的檔案的指標**       93. **fclose(fp);**       94. **printf("[-]Disconnected from server.\n");**       95. **// 關閉client端socket**       96. **close(clientSocket);**       97. **// 結束應用程式**       98. **exit(1);**       99. **}**       100. **return 0;**       101. **}** |

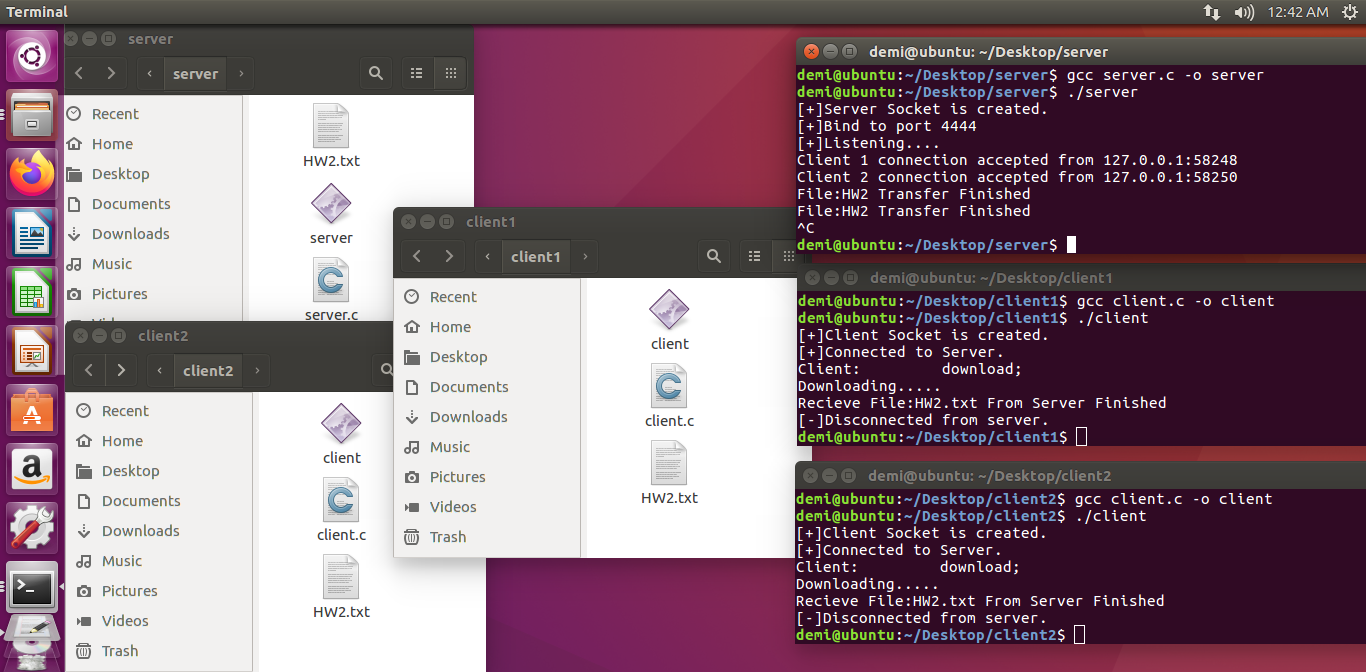
**架構流程圖**



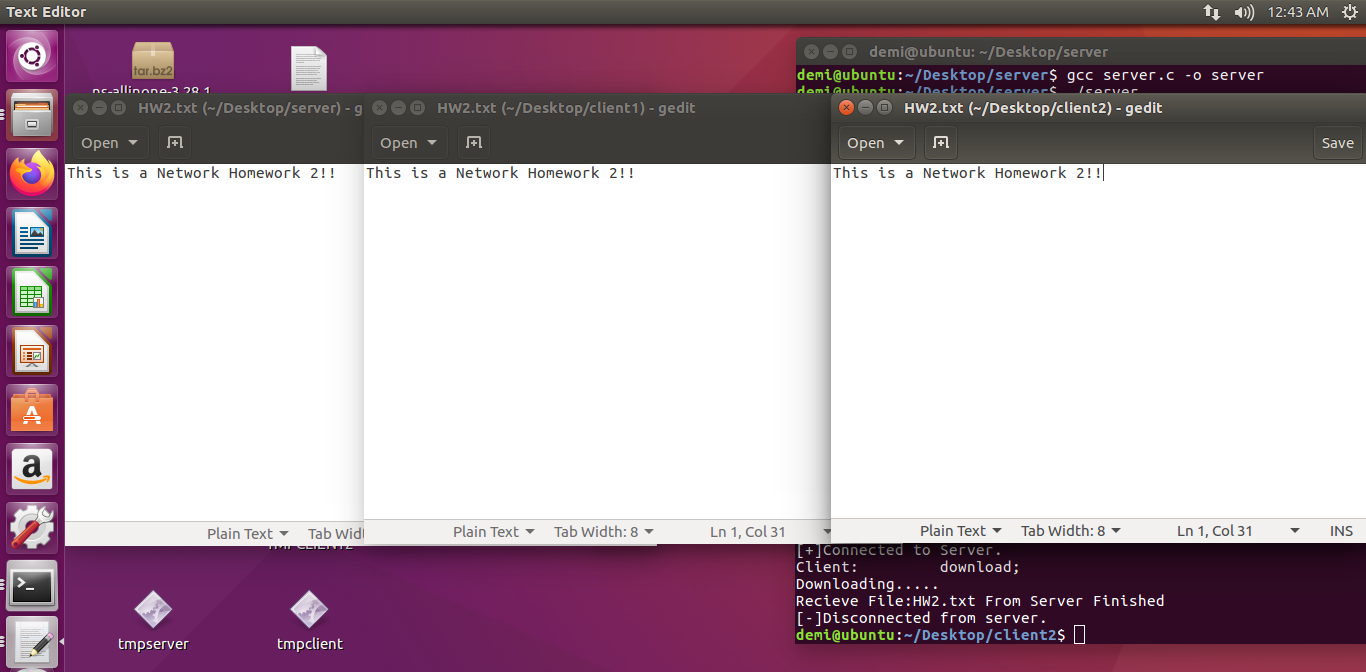
**實驗過程、內容及結果**



**圖(1)：我將server及client1、client2分至於三個資料夾，模擬一個server和多個client。並開啟三個終端機各自編譯後執行。另外在server的資料夾中有準備讓client端下載的檔案HW2.txt，且在client端還未下載的時候其並未有此檔。**



**圖(2)：clinet1及client2連至server，此server利用不同的port number可以辨識不同的client端。而client1和client2分別輸入download;來下載檔案，將server按ctrl+c關閉之後，兩個client端成功下載檔案。下載完畢後可以發現client1和client2的資料夾內各自多了一個下載下來的HW2.txt**



**圖(3)：分別打開server、client1、client2的HW2.txt檔案，檔案內容相同，下載成功！**