**Assignment – 2**

2. Write a file transfer program for TCP client and server (Mininet is must). Total 40 marks

* A topology with at least three hosts h1, h2, h3. h1 is the server, h2, h3 are clients. 5 marks
* A file should be uploaded by h2 to h1 and downloaded by the client h3 from h1 20 marks
* Calculate the time required to upload and download 10 marks
* Change the download/upload rate of a given file (change the buffer size) 5 marks

**Solution:**

**Server.c Program**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/time.h>

#include <arpa/inet.h>

#include <netinet/in.h>

#include <sys/socket.h>

#define MAX\_LINE 50

#define LINSTENPORT 7788

#define SERVERPORT 8877

#define BUFFSIZE 50

void writefile(int sockfd, FILE \*fp);

void sendfile(FILE \*fp, int sockfd);

ssize\_t total=0;

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

{

struct timeval start\_time;

struct timeval end\_time;

int sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

if (sockfd == -1)

{

perror("Can't allocate sockfd");

exit(1);

}

struct sockaddr\_in clientaddr, serveraddr;

memset(&serveraddr, 0, sizeof(serveraddr));

serveraddr.sin\_family = AF\_INET;

serveraddr.sin\_addr.s\_addr = inet\_addr("10.0.0.1");

serveraddr.sin\_port = htons(SERVERPORT);

if (bind(sockfd, (const struct sockaddr \*) &serveraddr, sizeof(serveraddr)) == -1)

{

perror("Bind Error");

exit(1);

}

if (listen(sockfd, LINSTENPORT) == -1)

{

perror("Listen Error");

exit(1);

}

printf("\nServer Listening on port %d \n", SERVERPORT);

while(1){

socklen\_t addrlen = sizeof(clientaddr);

int connfd = accept(sockfd, (struct sockaddr \*) &clientaddr, &addrlen);

if (connfd == -1)

{

perror("Connect Error");

exit(1);

}

// close(sockfd);

char operation[BUFFSIZE] = {0};

recv(connfd, operation, BUFFSIZE, 0);

char filename[BUFFSIZE] = {0};

if(strcmp(operation, "upload") == 0){

if (recv(connfd, filename, BUFFSIZE, 0) == -1) {

perror("Can't receive filename");

exit(1);

}

FILE \*fp = fopen(filename, "wb");

if (fp == NULL)

{

perror("Can't open file");

exit(1);

}

char addr[INET\_ADDRSTRLEN];

printf("\n-------------------------------------------------------------\n");

printf("Uploading file: %s by %s\n", filename, inet\_ntop(AF\_INET, &clientaddr.sin\_addr, addr, INET\_ADDRSTRLEN));

gettimeofday(&start\_time, NULL);

writefile(connfd, fp);

gettimeofday(&end\_time, NULL);

fclose(fp);

printf("Upload Success, Total Bytes = %ld\n", total);

printf("Time taken: %ld secs \n\n", end\_time.tv\_sec - start\_time.tv\_sec);

}

if(strcmp(operation, "download") == 0){

if (recv(connfd, filename, BUFFSIZE, 0) == -1) {

perror("Can't receive filename");

exit(1);

}

FILE \*fp = fopen(filename, "rb");

if (fp == NULL)

{

perror("Can't open file");

exit(1);

}

char addr[INET\_ADDRSTRLEN];

printf("\n-------------------------------------------------------------\n");

printf("Downloading file: %s by %s\n", filename, inet\_ntop(AF\_INET, &clientaddr.sin\_addr, addr, INET\_ADDRSTRLEN));

gettimeofday(&start\_time, NULL);

sendfile(fp, connfd);

gettimeofday(&end\_time, NULL);

fclose(fp);

printf("File download success, Total Bytes = %ld\n", total);

printf("Time taken: %ld secs \n\n", end\_time.tv\_sec - start\_time.tv\_sec);

}

close(connfd);

}

return 0;

}

// write file fn def

void writefile(int sockfd, FILE \*fp)

{

ssize\_t n;

char buff[MAX\_LINE] = {0};

while ((n = recv(sockfd, buff, MAX\_LINE, 0)) > 0)

{

total+=n;

if (n == -1)

{

perror("Receive File Error");

exit(1);

}

if (fwrite(buff, sizeof(char), n, fp) != n)

{

perror("Write File Error");

exit(1);

}

memset(buff, 0, MAX\_LINE);

}

}

// send file for download

void sendfile(FILE \*fp, int sockfd)

{

int n;

char sendline[MAX\_LINE] = {0};

while ((n = fread(sendline, sizeof(char), MAX\_LINE, fp)) > 0)

{

total+=n;

if (n != MAX\_LINE && ferror(fp))

{

perror("Read File Error");

exit(1);

}

if (send(sockfd, sendline, n, 0) == -1)

{

perror("Can't send file");

exit(1);

}

memset(sendline, 0, MAX\_LINE);

}

}

Client.c program - the program to upload file to server

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <libgen.h>

#include <unistd.h>

#include <arpa/inet.h>

#include <time.h>

#include <sys/time.h>

#include <netinet/in.h>

#include <sys/socket.h>

#define MAX\_LINE 50

#define LINSTENPORT 7788

#define SERVERPORT 8877

#define BUFFSIZE 50

void sendfile(FILE \*fp, int sockfd);

ssize\_t total=0;

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

{

char buff[BUFFSIZE] = {0};

struct timeval start\_time;

struct timeval end\_time;

if (argc != 3)

{

perror("usage:./client u <filepath>");

exit(1);

}

int sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

if (sockfd < 0)

{

perror("Can't allocate sockfd");

exit(1);

}

struct sockaddr\_in serveraddr;

memset(&serveraddr, 0, sizeof(serveraddr));

serveraddr.sin\_family = AF\_INET;

serveraddr.sin\_port = htons(SERVERPORT);

serveraddr.sin\_addr.s\_addr = inet\_addr("10.0.0.1");

if (connect(sockfd, (const struct sockaddr \*) &serveraddr, sizeof(serveraddr)) < 0)

{

perror("Connect Error");

exit(1);

}

// getting the user operation

strcpy(buff, argv[1]);

send(sockfd,buff, BUFFSIZE, 0);

char \*filename = basename(argv[2]);

if (filename == NULL)

{

perror("Can't get filename");

exit(1);

}

strncpy(buff, filename, strlen(filename));

if (send(sockfd, buff, BUFFSIZE, 0) == -1)

{

perror("Can't send filename");

exit(1);

}

printf("\nUploading File... \"%s\"\n", filename);

FILE \*fp = fopen(argv[2], "rb");

if (fp == NULL)

{

perror("Can't open file");

exit(1);

}

gettimeofday(&start\_time, NULL);

sendfile(fp, sockfd);

gettimeofday(&end\_time, NULL);

//puts("Send Success");

printf("Upload Success, Total Bytes = %ld\n", total);

printf("Time taken: %ld sec\n", end\_time.tv\_sec - start\_time.tv\_sec);

printf("Upload rate: %ld B/s (%.2lf MB/s)\n\n", total/(end\_time.tv\_sec - start\_time.tv\_sec),(double)(total/(end\_time.tv\_sec - start\_time.tv\_sec))/1000000);

fclose(fp);

close(sockfd);

return 0;

}

void sendfile(FILE \*fp, int sockfd)

{

int n;

char sendline[MAX\_LINE] = {0};

while ((n = fread(sendline, sizeof(char), MAX\_LINE, fp)) > 0)

{

total+=n;

if (n != MAX\_LINE && ferror(fp))

{

perror("Read File Error");

exit(1);

}

if (send(sockfd, sendline, n, 0) == -1)

{

perror("Can't send file");

exit(1);

}

memset(sendline, 0, MAX\_LINE);

}

}

dclient.c - the program to download file

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <libgen.h>

#include <unistd.h>

#include <arpa/inet.h>

#include <time.h>

#include <sys/time.h>

#include <netinet/in.h>

#include <sys/socket.h>

#define MAX\_LINE 50

#define LINSTENPORT 7788

#define SERVERPORT 8877

#define BUFFSIZE 50

void writefile(int sockfd, FILE \*fp);

ssize\_t total=0;

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

{

char buff[BUFFSIZE] = {0};

struct timeval start\_time;

struct timeval end\_time;

if (argc != 3)

{

perror("usage:./client u <filepath>");

exit(1);

}

int sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

if (sockfd < 0)

{

perror("Can't allocate sockfd");

exit(1);

}

struct sockaddr\_in serveraddr;

memset(&serveraddr, 0, sizeof(serveraddr));

serveraddr.sin\_family = AF\_INET;

serveraddr.sin\_port = htons(SERVERPORT);

serveraddr.sin\_addr.s\_addr = inet\_addr("10.0.0.1");

if (connect(sockfd, (const struct sockaddr \*) &serveraddr, sizeof(serveraddr)) < 0)

{

perror("Connect Error");

exit(1);

}

// getting the user operation

strcpy(buff, argv[1]);

send(sockfd,buff, BUFFSIZE, 0);

char \*filename = basename(argv[2]);

if (filename == NULL)

{

perror("Can't get filename");

exit(1);

}

strncpy(buff, filename, strlen(filename));

if (send(sockfd, buff, BUFFSIZE, 0) == -1)

{

perror("Can't send filename");

exit(1);

}

printf("\nDownloading File... \"%s\" \n", filename);

FILE \*fp = fopen(filename, "wb");

if (fp == NULL)

{

perror("Can't open file");

exit(1);

}

gettimeofday(&start\_time, NULL);

writefile(sockfd, fp);

gettimeofday(&end\_time, NULL);

printf("Download Success, Total Bytes = %ld\n", total);

printf("Time taken: %ld sec\n", end\_time.tv\_sec - start\_time.tv\_sec);

printf("Download rate: %ld B/s (%.2lf MB/s)\n\n", total/(end\_time.tv\_sec - start\_time.tv\_sec),(double)(total/(end\_time.tv\_sec - start\_time.tv\_sec))/1000000);

fclose(fp);

close(sockfd);

return 0;

}

// write file fn def

void writefile(int sockfd, FILE \*fp)

{

ssize\_t n;

char buff[MAX\_LINE] = {0};

while ((n = recv(sockfd, buff, MAX\_LINE, 0)) > 0)

{

total+=n;

if (n == -1)

{

perror("Receive File Error");

exit(1);

}

if (fwrite(buff, sizeof(char), n, fp) != n)

{

perror("Write File Error");

exit(1);

}

memset(buff, 0, MAX\_LINE);

}

}

**Output:**

