

COMPUTER NETWORKS

Name: Vinayak Sethi

Roll No: COE18B061

Assignment 6

Date: 9th September 2020

BITMAP Image Transfer using TCP

The general idea of sending the bitmap file is opening the image in 'rb' mode in client side and storing the content of it in sendbuffer using fread and then send the **sendbuffer** using **send() function** and then in server side opening the file in 'wb' mode and receiving the file using **recv() function** and write the contents in file using fwrite function.

Filename: TCP_Client_bmp.c

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<sys/time.h>
#include<netinet/in.h>
#include<unistd.h>
#include<string.h>
#include<libgen.h>
#include<time.h>

#define buffsize 1024

int main()
{
    int client_socket, sin_size, check, size = 0;
    char sendbuffer[buffsize];
    struct timeval start, end;
    struct sockaddr_in server_address;
```

```

//create a socket
client_socket = socket(AF_INET,SOCK_STREAM,0);
if(client_socket == -1)
{
    printf("\nSocket Creation Failure\n");
    exit(EXIT_FAILURE);
}

//specify an address for the socket
server_address.sin_family = AF_INET;
server_address.sin_port = htons(9009);
server_address.sin_addr.s_addr = INADDR_ANY;

sin_size = sizeof(struct sockaddr_in);

//connect to server
if(connect(client_socket,(struct sockaddr *)&server_address,
sin_size) == 0)
    printf("Connect Successful\n");

char path[bufsize];
printf("Enter the complete path of the filename you wish to send :
");
scanf("%s",path);
char *filename = basename(path); //return the last component of a
pathname
printf("%s\n", filename);

send(client_socket, filename, strlen(filename), 0);

FILE *fp = fopen(path,"rb");
if (fp == NULL)
{
    printf("Cannot open the file\n");
    exit(EXIT_FAILURE);
}

gettimeofday(&start, NULL);

while((check = fread(sendbuffer, 1, sizeof(sendbuffer), fp)) > 0)
{

```

```

        size = size + check;
        send(client_socket, sendbuffer, check, 0);
    }

    gettimeofday(&end, NULL);
    double duration = (double)(end.tv_usec - start.tv_usec) / 1000000 +
(double)(end.tv_sec - start.tv_sec);

    fclose(fp);

    printf("File sent successfully ...\n");
    printf("Transferred file size: %f MiB\n", (float) size / 1000000);
    printf("Time taken to transfer the file: %f seconds.\n", duration);

    close(client_socket);
    return 0;
}

```

The ***gettimeofday()*** function gets the system's clock time.

```

int gettimeofday ( struct timeval *tp , struct timezone *tz )

```

The ***gettimeofday()*** function is defined in **<sys/time.h>** header file.

Filename: TCP_Server_bmp.c

```

#include<stdio.h>
#include<stdlib.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<sys/time.h>
#include<netinet/in.h>
#include<unistd.h>
#include<string.h>
#include<time.h>

#define bufsize 1024

```

```

int main()
{
    int server_socket, client_socket, sin_size, check;
    int size = 0;
    char buffer[buffsize];
    struct timeval start, end;
    struct sockaddr_in server_address, client_address;

    //create a socket
    server_socket = socket(AF_INET, SOCK_STREAM, 0);
    if(server_socket == -1)
    {
        printf("\nSocket Creation Failure\n");
        exit(EXIT_FAILURE);
    }

    //specify an address for the socket
    server_address.sin_family = AF_INET;
    server_address.sin_port = htons(9009);
    server_address.sin_addr.s_addr = INADDR_ANY;

    //bind with the client
    if( bind(server_socket, (const struct sockaddr *)&server_address,
sizeof(server_address)) < 0)
    {
        printf("Could not bind to Client\n");
        exit(EXIT_FAILURE);
    }

    //listen to the incoming client request
    if(listen(server_socket, 10) == 0)
        printf("Listen successful\n");

    //accept a connection request from client
    sin_size = sizeof(struct sockaddr_in);
    if((client_socket = accept(server_socket, (struct sockaddr
*)&client_address, &sin_size)) > 0)
        printf("Accept Successful\n");

    char filename[buffsize];
    memset (filename, '\0', sizeof(filename));

```

```

recv(client_socket, filename, sizeof(filename), 0);
printf("File received is : ");
printf("%s\n", filename);

FILE *fp = fopen(filename, "wb");
gettimeofday(&start, NULL);
if(fp!=NULL)
{
    while((check = recv(client_socket,buffer, sizeof(buffer), 0)) >
0)
    {
        size = size + check;
        fwrite(buffer, 1, check, fp);
    }
fclose(fp);
}

gettimeofday(&end, NULL);
double duration = (double)(end.tv_usec - start.tv_usec) / 1000000 +
(double)(end.tv_sec - start.tv_sec);

printf("File received successfully...\n");
printf("Received file size: %f MiB\n", (float) size / 1000000);
printf("Time taken to receive the file the file: %f seconds.\n",
duration);

//close the socket
close(client_socket);
close(server_socket);

return 0;
}

```

Output:

```
vinayak@vinayak-Swift-SF315-52G: ~/Documents/CN/Lab/Bitmap/TCP/Client - 1 x
File Edit View Search Terminal Help
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/TCP/Client$ make TCP_Client_bmp
make: 'TCP_Client_bmp' is up to date.
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/TCP/Client$ ./TCP_Client_bmp
Connect Successful
Enter the complete path of the filename you wish to send : Land
Land
File sent successfully ...
Transferred file size: 0.787510 MiB
Time taken to transfer the file: 0.004658 seconds.
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/TCP/Client$ ls -l
total 4960
-rw-rw-r-- 1 vinayak vinayak 787510 Sep 23 16:51 Land
-rw-rw-r-- 1 vinayak vinayak 4264316 Sep 23 16:54 Marbles
-rwxr-xr-x 1 vinayak vinayak 13064 Sep 23 17:09 TCP_Client_bmp
-rwxrwxrwx 1 vinayak vinayak 1798 Sep 23 17:08 TCP_Client_bmp.c
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/TCP/Client$ ./TCP_Client_bmp
Connect Successful
Enter the complete path of the filename you wish to send : Marbles
Marbles
File sent successfully ...
Transferred file size: 4.264316 MiB
Time taken to transfer the file: 0.014536 seconds.
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/TCP/Client$ ls -l
total 4960
-rw-rw-r-- 1 vinayak vinayak 787510 Sep 23 16:51 Land
-rw-rw-r-- 1 vinayak vinayak 4264316 Sep 23 16:54 Marbles
-rwxr-xr-x 1 vinayak vinayak 13064 Sep 23 17:09 TCP_Client_bmp
-rwxrwxrwx 1 vinayak vinayak 1798 Sep 23 17:08 TCP_Client_bmp.c
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/TCP/Client$

vinayak@vinayak-Swift-SF315-52G: ~/Documents/CN/Lab/Bitmap/TCP/Server - 1 x
File Edit View Search Terminal Help
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/TCP/Server$ make TCP_Server_bmp
make: 'TCP_Server_bmp' is up to date.
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/TCP/Server$ ./TCP_Server_bmp
Listen successful
Accept Successful
File received is : Land
File received successfully...
Received file size: 0.787510 MiB
Time taken to receive the file: 0.004847 seconds.
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/TCP/Server$ ls -l
total 792
-rw-r--r-- 1 vinayak vinayak 787510 Sep 23 20:16 Land
-rwxr-xr-x 1 vinayak vinayak 13048 Sep 23 17:09 TCP_Server_bmp
-rwxrwxrwx 1 vinayak vinayak 2045 Sep 23 17:08 TCP_Server_bmp.c
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/TCP/Server$ ./TCP_Server_bmp
Listen successful
Accept Successful
File received is : Marbles
File received successfully...
Received file size: 4.264316 MiB
Time taken to receive the file: 0.016634 seconds.
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/TCP/Server$ ls -l
total 4960
-rw-r--r-- 1 vinayak vinayak 787510 Sep 23 20:16 Land
-rw-r--r-- 1 vinayak vinayak 4264316 Sep 23 20:16 Marbles
-rwxr-xr-x 1 vinayak vinayak 13048 Sep 23 17:09 TCP_Server_bmp
-rwxrwxrwx 1 vinayak vinayak 2045 Sep 23 17:08 TCP_Server_bmp.c
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/TCP/Server$
```

BITMAP Image Transfer using UDP

Same concept is used like TCP bitmap image transfer logic but in UDP there is chances of data loss, so to reduce data loss we have used sleep command because client keep sending data to server and server take time to write to file so to slow down this process sleep command is used. Also some functions are changed as per UDP protocol, like **sendto()** in client side and **recvfrom()** in server side is used.

Also fgetc command is used to read the content byte by byte and fputc command is used to write byte by byte to file on the server side.

Filename: UDP_Client_bmp.c

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<netinet/in.h>
#include<unistd.h>
#include<string.h>
#include<libgen.h>
#include<sys/time.h>

#define buffsize 1024

int main()
{
    int client_socket,check,size = 0;
    struct timeval start, end;
    struct sockaddr_in server_address;

    //create a socket
    client_socket = socket(AF_INET,SOCK_DGRAM,0);
    if(client_socket == -1)
    {
        printf("\nSocket Creation Failure\n");
        exit(EXIT_FAILURE);
    }

    //specify an address for the socket
    server_address.sin_family = AF_INET;
    server_address.sin_port = htons(9009);
    server_address.sin_addr.s_addr = INADDR_ANY;

    socklen_t length = sizeof(server_address);

    sendto(client_socket,"Hello server", strlen("Hello server"),0,(struct
sockaddr *)&server_address, sizeof(server_address));

    char path[buffsize];
    printf("Enter the complete path of the filename you wish to send :
");
    scanf("%s",path);
```

```

    char *filename = basename(path); //return the last component of a
pathname
    printf("%s\n", filename);

    sendto(client_socket, filename, strlen(filename), 0, (struct sockaddr
*)&server_address, sizeof(server_address));

    FILE *fp = fopen(path, "rb");
    if (fp == NULL)
    {
        printf("Cannot open the file\n");
        exit(EXIT_FAILURE);
    }

    gettimeofday(&start, NULL);

    while((check = fgetc(fp)) != EOF)
    {
        size++;
        sendto(client_socket, &check, sizeof(check), 0, (struct sockaddr
*)&server_address, sizeof(server_address));
        sleep(0);
    }

    //send the EOF to signal file end
    size++;
    sendto(client_socket, &check, sizeof(check), 0, (struct sockaddr
*)&server_address, sizeof(server_address));

    gettimeofday(&end, NULL);
    double duration = (double)(end.tv_usec - start.tv_usec) / 1000000 +
(double)(end.tv_sec - start.tv_sec);

    fclose(fp);

    printf("File sent successfully ...\n");
    printf("Transferred file size: %f MiB\n", (float) size / 1000000);
    printf("Time taken to transfer the file: %f seconds.\n", duration);

    close(client_socket);
    return 0;
}

```


Filename: UDP_Server_bmp.c

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<netinet/in.h>
#include<unistd.h>
#include<string.h>
#include<sys/time.h>

#define buffsize 1024

int main()
{
    int server_socket,check,store, size = 0;
    char buffer[buffsize];
    struct timeval start, end;
    struct sockaddr_in server_address, client_address;

    //create a socket
    server_socket = socket(AF_INET,SOCK_DGRAM,0);
    if(server_socket == -1)
    {
        printf("\nSocket Creation Failure\n");
        exit(EXIT_FAILURE);
    }

    //specify an address for the socket
    server_address.sin_family = AF_INET;
    server_address.sin_port = htons(9009);
    server_address.sin_addr.s_addr = INADDR_ANY;

    //bind with the client
    if( bind(server_socket, (const struct sockaddr *)&server_address,
sizeof(server_address)) < 0)
    {
        printf("Could not bind to Client\n");
        exit(EXIT_FAILURE);
    }
}
```

```

    socklen_t length = sizeof(server_address);
    recvfrom(server_socket, buffer, sizeof(buffer), 0, (struct sockaddr
*)&client_address, &length);

    char filename[bufsize];
    memset (filename, '\0', sizeof(filename));
    recvfrom(server_socket, filename, sizeof(filename), 0, (struct
sockaddr *)&client_address, &length);
    printf("File received is : ");
    printf("%s\n", filename);

    FILE *fp = fopen(filename, "wb");
    gettimeofday(&start, NULL);

    if(fp!=NULL)
    {
        while(store != EOF)
        {
            recvfrom(server_socket, &store, sizeof(store), 0, (struct
sockaddr *)&client_address, &length);
            size++;
            fputc(store, fp);
        }
    }

    fclose(fp);

    gettimeofday(&end, NULL);
    double duration = (double)(end.tv_usec - start.tv_usec) / 1000000 +
(double)(end.tv_sec - start.tv_sec);

    printf("File received successfully...\n");
    printf("Received file size: %f MiB\n", (float) size / 1000000);
    printf("Time taken to receive the file the file: %f seconds.\n",
duration);

    //close the socket
    close(server_socket);
    return 0;
}

```

Output:

```
vinayak@vinayak-Swift-SF315-52G: ~/Documents/CN/Lab/Bitmap/UDP/Client$ make UDP_Client_bmp
make: 'UDP_Client_bmp' is up to date.
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/UDP/Client$ ./UDP_Client_bmp
Enter the complete path of the filename you wish to send : Land
Land
File sent successfully ...
Transferred file size: 0.787511 MiB
Time taken to transfer the file: 57.699979 seconds.
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/UDP/Client$ ls -l
total 4960
-rw-rw-r-- 1 vinayak vinayak 787510 Sep 23 16:51 Land
-rw-rw-r-- 1 vinayak vinayak 4264316 Sep 23 16:54 Marbles
-rwxr-xr-x 1 vinayak vinayak 13064 Sep 23 19:54 UDP_Client_bmp
-rwxrwxr-x 1 vinayak vinayak 1981 Sep 23 19:53 UDP_Client_bmp.c
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/UDP/Client$ make UDP_Client_bmp
make: 'UDP_Client_bmp' is up to date.
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/UDP/Client$ ./UDP_Client_bmp
Enter the complete path of the filename you wish to send : Marbles
Marbles
File sent successfully ...
Transferred file size: 4.264317 MiB
Time taken to transfer the file: 316.351543 seconds.
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/UDP/Client$
```

```
vinayak@vinayak-Swift-SF315-52G: ~/Documents/CN/Lab/Bitmap/UDP/Server$ make UDP_Server_bmp
make: 'UDP_Server_bmp' is up to date.
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/UDP/Server$ ./UDP_Server_bmp
File received is : Land
File received successfully...
Received file size: 0.787511 MiB
Time taken to receive the file the file: 57.699630 seconds.
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/UDP/Server$ ls -l
total 792
-rw-r--r-- 1 vinayak vinayak 787511 Sep 23 20:01 Land
-rwxr-xr-x 1 vinayak vinayak 12968 Sep 23 19:40 UDP_Server_bmp
-rwxrwxr-x 1 vinayak vinayak 1874 Sep 23 19:39 UDP_Server_bmp.c
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/UDP/Server$ make UDP_Server_bmp
make: 'UDP_Server_bmp' is up to date.
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/UDP/Server$ ./UDP_Server_bmp
File received is : Marbles
File received successfully...
Received file size: 4.264273 MiB
Time taken to receive the file the file: 316.351148 seconds.
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/UDP/Server$ ls -l
total 4960
-rw-r--r-- 1 vinayak vinayak 787511 Sep 23 20:01 Land
-rw-r--r-- 1 vinayak vinayak 4264273 Sep 23 20:08 Marbles
-rwxr-xr-x 1 vinayak vinayak 12968 Sep 23 19:40 UDP_Server_bmp
-rwxrwxr-x 1 vinayak vinayak 1874 Sep 23 19:39 UDP_Server_bmp.c
vinayak@vinayak-Swift-SF315-52G:~/Documents/CN/Lab/Bitmap/UDP/Server$
```