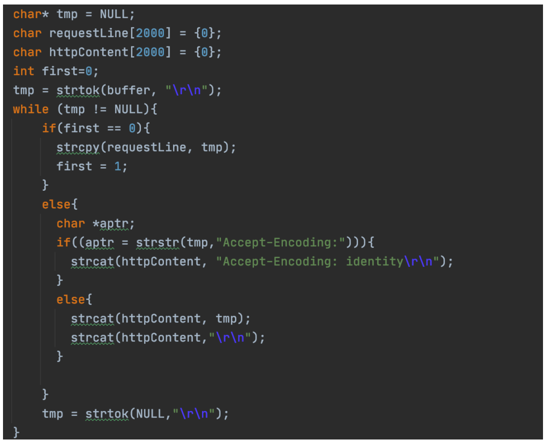
Project 3 보고서

2017-11621 전기정보공학부 배병욱

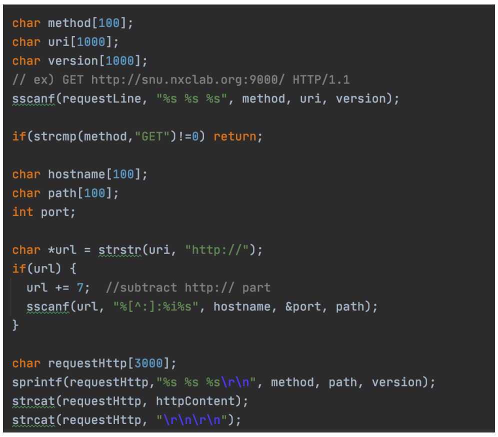
1. Proxy Server

1-a. Parsing http request

First, we must parse http request from client to send to host. By using strok() and use \r\n for token. By this we can parse first line for http request header and remainder for http contents. To make server's response readable, code changed Accept-Encoding to identity. First line of the http request is parsed and changed to <GET> <path> <version> using sprintf function.



```
char* tmp = NULL;
char requestLine[2000] = {0};
char httpContent[2000] = {0};
int first=0;
tmp = strtok(buffer, "\r\n");
while (tmp != NULL){
    if(first == 0){
      strcpy(requestLine, tmp);
      first = 1;
    }
    else{
      char *aptr;
      if((aptr = strstr(tmp,"Accept-Encoding:"))){
        strcat(httpContent, "Accept-Encoding: identity\r\n");
      }
      else{
        strcat(httpContent, tmp);
        strcat(httpContent,"\r\n");
      }

    }
    tmp = strtok(NULL,"\r\n");
}
```



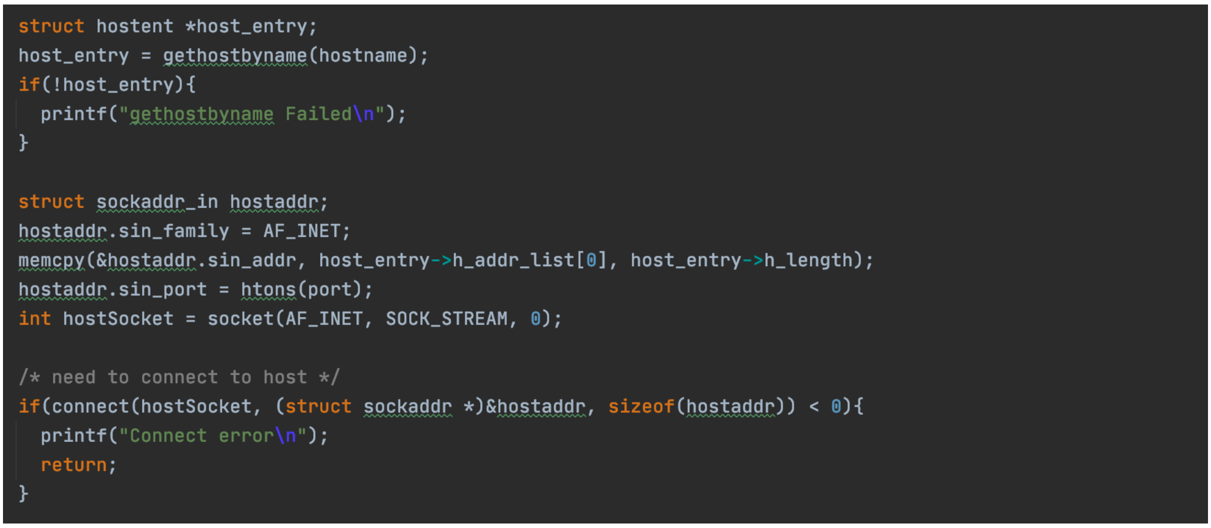
```
char method[100];
char uri[1000];
char version[1000];
// ex) GET http://snu.nxclab.org:9000/ HTTP/1.1
sscanf(requestLine, "%s %s %s", method, uri, version);

if(strcmp(method,"GET")!=0) return;

char hostname[100];
char path[100];
int port;

char *url = strstr(uri, "http://");
if(url) {
  url += 7;  //subtract http:// part
  sscanf(url, "%[^:]:%i%s", hostname, &port, path);
}

char requestHttp[3000];
sprintf(requestHttp,"%s %s %s\r\n", method, path, version);
strcat(requestHttp, httpContent);
strcat(requestHttp, "\r\n\r\n");
```

Fig 1. Parsing Http request code

Fig 2. Http header code

1-b. Connect to host and receive data from host

After making http request, proxy server must make connection with host. Use gethostbyname to get host address and connect to host by host address



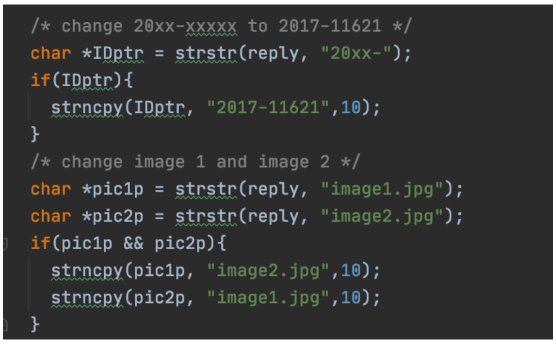
```
struct hostent *host_entry;
host_entry = gethostbyname(hostname);
if(!host_entry){
  printf("gethostbyname Failed\n");
}

struct sockaddr_in hostaddr;
hostaddr.sin_family = AF_INET;
memcpy(&hostaddr.sin_addr, host_entry->h_addr_list[0], host_entry->h_length);
hostaddr.sin_port = htons(port);
int hostSocket = socket(AF_INET, SOCK_STREAM, 0);

/* need to connect to host */
if(connect(hostSocket, (struct sockaddr *)&hostaddr, sizeof(hostaddr)) < 0){
  printf("Connect error\n");
  return;
}
```

Fig 3. Connection with host code

Next get response from host by using recv() function same as skeleton code.

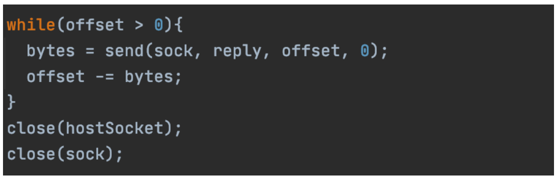
1-c. Modify data and send data to client

To locate data's location, use strstr() function as below. And change data by strcpy() function. Then send data by using send() function. As we must send jpg file use while statement as jpg file is binary file. After sending is finished, close sockets.



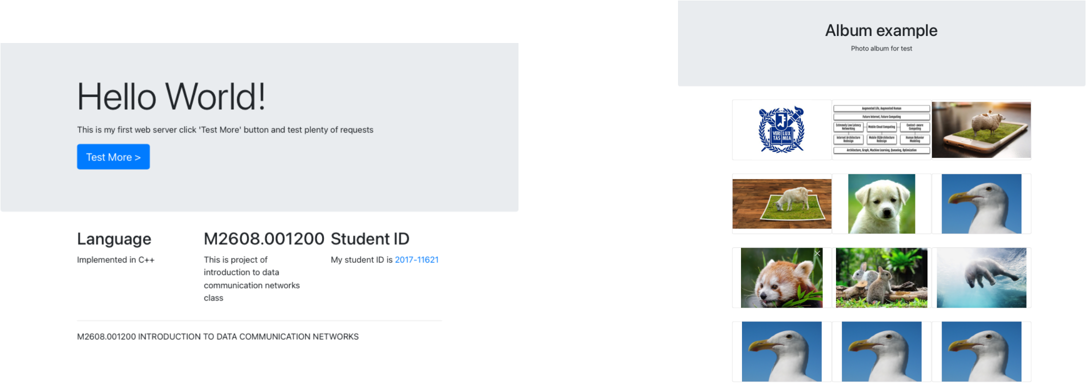
```
/* change 20xx-xxxxx to 2017-11621 */
char *IDptr = strstr(reply, "20xx-");
if(IDptr){
  strncpy(IDptr, "2017-11621",10);
}
/* change image 1 and image 2 */
char *pic1p = strstr(reply, "image1.jpg");
char *pic2p = strstr(reply, "image2.jpg");
if(pic1p && pic2p){
  strncpy(pic1p, "image2.jpg",10);
  strncpy(pic2p, "image1.jpg",10);
}
```

Fig 4. Code of modifying data



```
while(offset > 0){
  bytes = send(sock, reply, offset, 0);
  offset -= bytes;
}
close(hostSocket);
close(sock);
```

Fig 5. Code of send()



Hello World!

This is my first web server click 'Test More' button and test plenty of requests

Test More >

Language

Implemented in C++

M2608.001200

This is project of introduction to data communication networks class

Student ID

My student ID is 2017-11621

M2608.001200 INTRODUCTION TO DATA COMMUNICATION NETWORKS

Album example

Photo album for test

Fig 6. Result of changed response on client

2. Pros & Cons of Proxy/NAT

2-a. Proxy

Pros

1. Performance: As proxy can be serve role as Cache, it can reduce the load of host server, and also reduce RTT for client if proxy is closer to client.
2. Security: proxy can hide client's address and improve security.
3. Filtering: for some companies, they can use proxy to restrict communication with external network for their security.

Cons

1. Tracking: proxy can remember every data of clients, there are chances that someone who works on proxy can use this data.
2. Incompatibility: proxy is compatible with local network, but it can have problem with external network.
3. Cost: proxy server consumes additional cost.

2-b. NAT

Pros

1. Security: NAT hide internal network from external network.

2. Save public IP: by using NAT, we can save insufficient public IP
3. Simple network management: NAT can simplify network management as it is easy to add new network devices and administrator can manage network easily
Cons
1. Compatibility: NAT can also have problem with external network.
2. Control: NAT make administrator has limited control over network.
3. Complexity: NAT increase network complexity, results additional management.

3. Pseudo Code of simple web page caching HTTP proxy

```
HTTP proxy with Cache(){

  http request = Wait for HTTP request

  if (response = search Cache(http request)) is true:
      send to Client(response)

  else:
      response = send request to Server(http request)
      insert response to Cache(response)
            if Cacheable is true:
                insert response into Cache
            else:
                evict Cache or drop response according to replacement algorithm
        send to Client(response)

}
```