

2019년 데이터통신

- HW 10 -

이름	노효근
학번	201502049
분반	02

* 과제 목표

- Stop-and-Wait 개념에 대해서 이해한다.
- Stop-and-Wait을 파이썬에서 구현해본다.

* sender.py

```
sender_HW10_test.py - C:\Users\HGW\Desktop\Data Communication\HW10\sender_HW10_test.py (3.7.3)
File Edit Format Run Options Window Help

import socket
import os
import struct

# checksum 계산
def calc_checksum(string):
    sum = 0
    for i in range(len(string)):
        sum = sum + string[i]
        temp = sum >> 16
        chs = sum + temp
    if chs >= 65536:
        chs -= 65536
    if chs >= 131072:
        chs -= 131072
    chs = chs ^ 0xffff
    return chs

# 퍼센트 계산
def calc_percent(current_length, file_size):
    percent = current_length / file_size * 100
    if (current_length >= file_size):
        current_size = file_size
        percent = 100.0
    print("current_size / total_size : ", current_length, "/", file_size, percent, "%")

s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s.settimeout(4)

host = '192.168.230.128'
port = 6000

print("Sender Socket open...")
print("Receiver IP : ", host)
print("Receiver Port : ", port)

file_type = "s"
file_name = input("Input File name : ")
file_size = os.path.getsize(file_name)

b_file_size = file_size.to_bytes(20, byteorder = "big")
b_file_name = file_name.encode().ljust(15)
current_length = 0
seq_num = "1"

with open(file_name, 'rb') as file:
    payload = file.read(1024)
    header_frame = file_type.encode() + b_file_size + b_file_name + payload
    header_checksum = calc_checksum(header_frame).to_bytes(20, byteorder = "big")
    header_frame = file_type.encode() + header_checksum + b_file_size + b_file_name + payload
    pre_header_frame = header_frame
    s.sendto(header_frame, (host, port))
    print("Send File Info(file_Type, Checksum, File Name, File Size, Payload) to Server...")
    while(1):
        try:
            ACK, _ = s.recvfrom(1000)
            break
        except socket.timeout: # timeout으로 인한 재전송 요청
            print("TimeOut!! ***")
            print("Retransmission")
            s.sendto(pre_header_frame, (host, port))
            calc_percent(current_length, file_size)
            s.settimeout(4)
            continue
    if(ACK.decode() == "1"):
```

이전 과제와 마찬가지로 기본 상태를 세팅해준다. socket 라이브러리를 import한다. 이후, socket객체를 생성하여 통신에 필요한 배경을 만들어준다.

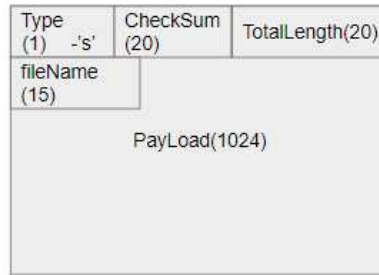
- def calc_checksum을 통해 내가 보낼 header와 payload에 대한 checksum을 구하는 함수를 구현하였다.

- def calc_percent을 통해 내가 보낸 파일에 대한 %를 출력하는 함수를 구현한다.

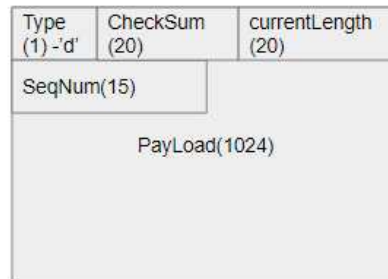
내가 보내줄 파일을 입력할 창을 만들고, 내가 보낼 파일의 총 size를 os.path.getsize() 메소드를 이용하여 구한다.

이후 with open을 통해 파일을 열고 처음으로 보낼 head에 관한 내용을 가져온다.

header에 필요한 내용을 가져와, 헤더에 대한 전송프레임에 맞게 to_bytes를 이용하여 아래 그림과 같은 format으로 Frame을 생성한다.



이후 ACK를 받고, header와 같은방식으로 전송 프레임에 맞게 to_bytes를 이용하여 아래 그림과 같은 format으로 Frame을 생성한다.



이때 reciver에게 ACK를 잘 받았을 경우 (ACK = 1), seq_num을 0에서1 바꾸고 다음 data를 읽어 보내도록 한다.

```

sender_HW10_test.py - C:\Users\HG\Desktop\Data Communication\HW10\sender_HW10_test.py (3.7.3)
File Edit Format Run Options Window Help
print("Send File Info(file_type, checksum, file Name, file Size, Payload ) to Server...")
while(1) :
    try :
        ACK,_ = s.recvfrom(1000)
        break
    except socket.timeout : # timeout으로 인한 재전송 요청
        print(" * Timeout!! *++")
        print("Retransmission")
        s.sendto(pre_header_frame,(host,port))
        calc_percent(current_length,file_size)
        s.settimeout(4)
        continue
    if(ACK.decode() == "1"):
        print("Start File send")
        current_length +=1024
        calc_percent(current_length,file_size)

    while (1) :
        file_type = "d"
        current_length +=1024
        b_current_length = current_length.to_bytes(20,byteorder = "big")
        b_seq_num = seq_num.encode().ljust(15)
        payload = file.read(1024)
        pre_payload = payload
        b_payload = file_type.encode()+b_current_length+b_seq_num+payload
        payload_checksum = calc_checksum(b_payload).to_bytes(20,byteorder = "big")
        payload_frame = file_type.encode()+payload_checksum+b_current_length+b_seq_num+payload
        pre_payload_frame = file_type.encode()+payload_checksum+b_current_length+b_seq_num+payload
        s.sendto(payload_frame,(host,port))

        if(current_length >= file_size):
            calc_percent(current_length,file_size)
            print("File send end")
            break

    try:
        ACK,_ = s.recvfrom(1000)

        if(ACK.decode() == "1"):
            calc_percent(current_length,file_size)
            seq_num = "1"
            continue

        elif(ACK.decode() == "0"):
            calc_percent(current_length,file_size)
            seq_num = "0"
            continue

        elif(ACK.decode() == "NAK2"): #프레임 순상으로 재전송 요청 / 수신측에서 버림
            print(" * Received NAK2")
            s.sendto(pre_payload_frame,(host,port))
            calc_percent(current_length,file_size)
            continue

        elif(ACK.decode() == "NAK1"): #순서가 뒤 바뀐 재전송 요청 / 수신측에서 버림
            print(" * Received NAK1")
            s.sendto(pre_payload_frame,(host,port))
            calc_percent(current_length,file_size)
            continue

    except socket.timeout : # timeout으로 인한 재전송 요청
        print(" * Timeout!! *++")
        print("Retransmission")
        s.sendto(pre_payload_frame,(host,port))
        calc_percent(current_length,file_size)
        s.settimeout(4)
        continue

```

해당 부분은 Error부분을 Control하는 부분이다.

위에서, settimeout을 이용하여 timeout이 걸리는 상황에 해야할 Error를 처리한다.

- NAK2를 받는 경우는 프레임 손상으로 reciver측에서 재전송 요청하여 이전에 보냈던 파일을 재전송한다.

- NAK1를 받는 경우는 순서가 뒤바뀌어 reciver측에서 재전송 요청하여 이전에 보냈던 파일을 재전송한다.

- settimeout을 이용하여, socket에 대한 time이 지나면, timeout으로 인하여, reciver측에서 재전송 요청하여 이전에 보냈던 파일을 재전송한다.

* reciver.py

```
reciver_HW10_test.py - C:\Users\HGW\Desktop\Data Communication\HW10\reciver_HW10_test.py (3.7.3)
File Edit Format Run Options Window Help

import socket
import os
import struct
import time as t
import sys

# checksum 계산
def calc_checksum(string,header_checksum):
    sum = 0
    for i in range(len(string)):
        sum = sum + string[i]
    temp = sum >> 16
    chs = sum + temp + header_checksum #byte형

    if chs >= 65536 :
        chs -= 65536
    if chs >= 131072 :
        chs -= 131072

    chs = chs ^ 0xffff
    return chs

# 퍼센트 계산
def calc_percent(current_length,file_size):
    percent = current_length/file_size*100

    if(current_length >= file_size):
        current_size = file_size
        percent = 100.0

    print("current_size / total_size : ", current_length, "/", file_size , percent,"%")

    if(percent == 100.0):
        print("File receive end")
        sys.exit()

s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
s.bind('',6000)

#host = '192.168.100.153'
#port = 6000

header_frame, addr = s.recvfrom(1080)

print("Receiver Socket open ")
print("Sender IP : ",addr[0])
print("Sender Port : ",addr[1])

file_type = chr(header_frame[0])
header_checksum = int.from_bytes(header_frame[1:21],byteorder = "big")
print(header_checksum)
file_size = int.from_bytes(header_frame[21:41],byteorder = "big")
file_name = header_frame[41:56]
file_name = file_name[0:file_name.find(32)].decode()
header_payload = header_frame[56:]

print("File Name = ", file_name)
print("File Size = ", file_size)

check_frame = file_type.encode()+file_size.to_bytes(20,byteorder = "big")+file_name.encode(),ljust(15)+header_payload.encode()

check_sum = calc_checksum(check_frame,header_checksum)
print(check_sum)

current_length = 0
ACK = "1"
pre_seqnum = "0";

while(1):
    if(check_sum == 0):
        with open(file_name, 'wb') as file :
            file.write(header_payload)
```

이전 과제와 마찬가지로 기본 상태를 세팅해준다. socket 라이브러리를 import한다.

이후, socket객체를 생성하여 통신에 필요한 배경을 만들어준다.

- def calc_checksum을 통해 내가 보낼 header와 payload에 대한 checksum을 구하는 함수를 구현하였다.

- def calc_percent을 통해 내가 보낸 파일에 대한 %를 출력하는 함수를 구현다.

sender에서 보낸 header의 내용을 읽어, checksum을 계산하여 잘 받았으면,

with open을 통해 파일을 열고 쓴다. 이후 sender에서 보내는 payload에 대한, 내용을 읽어, checksum을 계산하여 데이터를 계속 쓰도록한다.

```
reciver_HW10_test.py - C:\Users\HGW\Desktop\Data Communication\HW10\reciver_HW10_test.py (3.7.3)
File Edit Format Run Options Window Help

print(header_checksum)
file_size = int.from_bytes(header_frame[21:41],byteorder = "big")
file_name = header_frame[41:56]
file_name = file_name[0:file_name.find(32)].decode()
header_payload = header_frame[56:]
print("File Name = ", file_name)
print("File Size = ", file_size)
check_frame = file_type.encode()+file_size.to_bytes(20,byteorder = "big")+file_name.encode(),ljust(15)+header_payload.encode()
check_sum = calc_checksum(check_frame,header_checksum)
print(check_sum)
current_length = 0
ACK = "1"
pre_seqnum = "0";
while (1) :
    if(check_sum == 0):
        with open(file_name, 'wb') as file :
            file.write(header_payload)
            current_length += 1024
            calc_percent(current_length,file_size)
            s.sendto(ACK.encode(),addr)
        while (1) :
            payload_frame,addr = s.recvfrom(1080)
            payload_type = chr(payload_frame[0])
            payload_checksum = int.from_bytes(payload_frame[1:21],byteorder = "big")
            payload_current_length = int.from_bytes(payload_frame[21:41],byteorder = "big")
            payload_seqnum = payload_frame[41:56]
            payload_seqnum = payload_seqnum[0:payload_seqnum.find(32)].decode()
            payload = payload_frame[56:]
            check_frame = payload_type.encode()+payload_current_length.to_bytes(20,byteorder = "big")+payload_seqnum.encode(),ljust(15)+payload.encode()
            check_sum = calc_checksum(check_frame,payload_checksum)
            if(check_sum == 0):
                if(payload_seqnum != pre_seqnum):
                    file.write(payload)
                    calc_percent(payload_current_length,file_size)
                    pre_seqnum = payload_seqnum
                    if(payload_seqnum == "0"):
                        ACK = "1"
                        s.sendto(ACK.encode(),addr)
                        continue
                    if(payload_seqnum == "1"):
                        ACK = "0"
                        s.sendto(ACK.encode(),addr)
                        continue
                else:
                    ACK = "NAK1" #순서가 뒤바뀜 확인 = 재전송 요청 / 수신측에서 버림
                    s.sendto(ACK.encode(),addr)
                    continue
            else:
                ACK = "NAK2" #checksum 계산을 통한 프레임 손상 확인 = 재전송 요청 /수신측에서 버림
                s.sendto(ACK.encode(),addr)
                continue
        else:
            ACK = "NAK2" #checksum 계산을 통한 프레임 손상 확인 = 재전송 요청 /수신측에서 버림
            s.sendto(ACK.encode(),addr)
            continue
Ln: 33 Col: 32
```

이후 아래 해당 부분은 Error부분을 Control하는 부분이다.

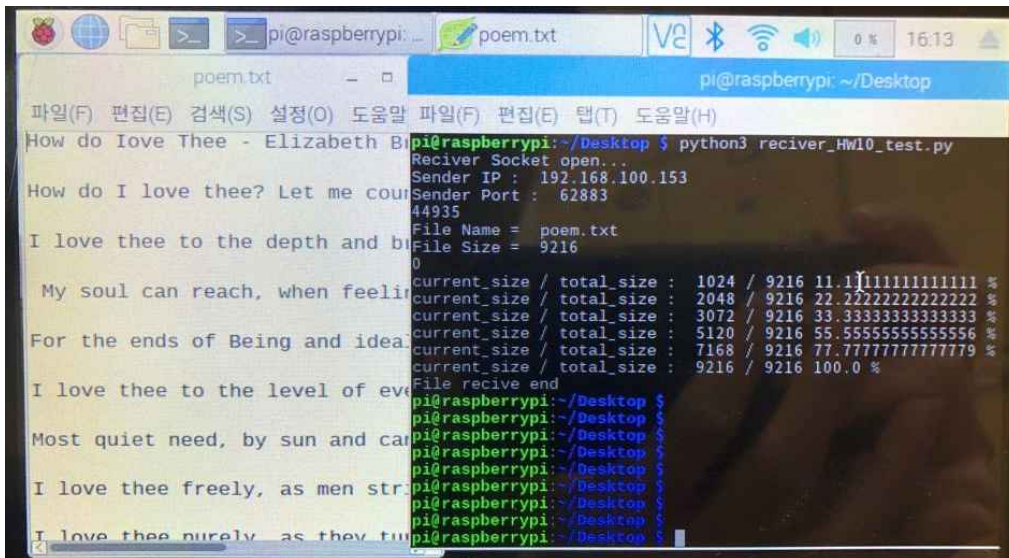
- NAK2를 받는 경우는 프레임 손상으로 sender측에 재전송 요청하여 이전에 보냈던 파일을 받도록 한다.
- NAK1를 받는 경우는 순서가 뒤바뀌어 sender측에 재전송 요청하여 이전에 보냈던 파일을 받도록 한다.

* 실행 결과

: 파일 전송을 하는 sender - 개인 노트북

```
u201502049@u201502049:~/Desktop/DC_10$ python3 sender_HW10_test.py
Sender Socket open...
Receiver IP : 192.168.100.152
Receiver Port : 6000
Input File name : poem.txt
Send File Info(file_Type, Checksum, file Name, file Size, Payload ) to Server...
Start File send
current_size / total_size : 1024 / 9216 11.111111111111 %
current_size / total_size : 2048 / 9216 22.222222222222 %
* Timeout!! ***
Retransmission
current_size / total_size : 3072 / 9216 33.333333333333 %
current_size / total_size : 4096 / 9216 44.444444444444 %
* Received NAK1
current_size / total_size : 5120 / 9216 55.555555555556 %
current_size / total_size : 6144 / 9216 66.666666666666 %
* Received NAK1
current_size / total_size : 7168 / 9216 77.777777777777 %
current_size / total_size : 8192 / 9216 88.888888888889 %
current_size / total_size : 9216 / 9216 100.0 %
File send end
```

: 파일을 받는 reciver - 지급받은 라즈베리파이



```
pi@raspberrypi: ~/Desktop
poem.txt
How do I love Thee - Elizabeth Barrett Browning
How do I love thee? Let me count the ways
I love thee to the depth and breadth and height
My soul can reach, when feeling out of sight
For the ends of Being and idealings
I love thee to the level of every soul
Most quiet need, by sun and candle-light
I love thee freely, as men strive for right
I love thee purely, as they turn from praise
```

```
pi@raspberrypi:~/Desktop$ python3 reciver_HW10_test.py
Receiver Socket open...
Sender IP : 192.168.100.153
Sender Port : 62883
44935
File Name = poem.txt
File Size = 9216
current_size / total_size : 1024 / 9216 11.111111111111 %
current_size / total_size : 2048 / 9216 22.222222222222 %
current_size / total_size : 3072 / 9216 33.333333333333 %
current_size / total_size : 5120 / 9216 55.555555555556 %
current_size / total_size : 7168 / 9216 77.777777777777 %
current_size / total_size : 9216 / 9216 100.0 %
File recive end
pi@raspberrypi:~/Desktop$
pi@raspberrypi:~/Desktop$
pi@raspberrypi:~/Desktop$
pi@raspberrypi:~/Desktop$
pi@raspberrypi:~/Desktop$
pi@raspberrypi:~/Desktop$
pi@raspberrypi:~/Desktop$
pi@raspberrypi:~/Desktop$
```

* GitHub ID : Nroot33