

Dokumentasi Praktikum Multi Threaded File Server



Pemrograman Jaringan
Kelas E

Muhammad Afif Fadhlurrahman
0511184000093

Institut Teknologi Sepuluh Nopember
Surabaya
2021

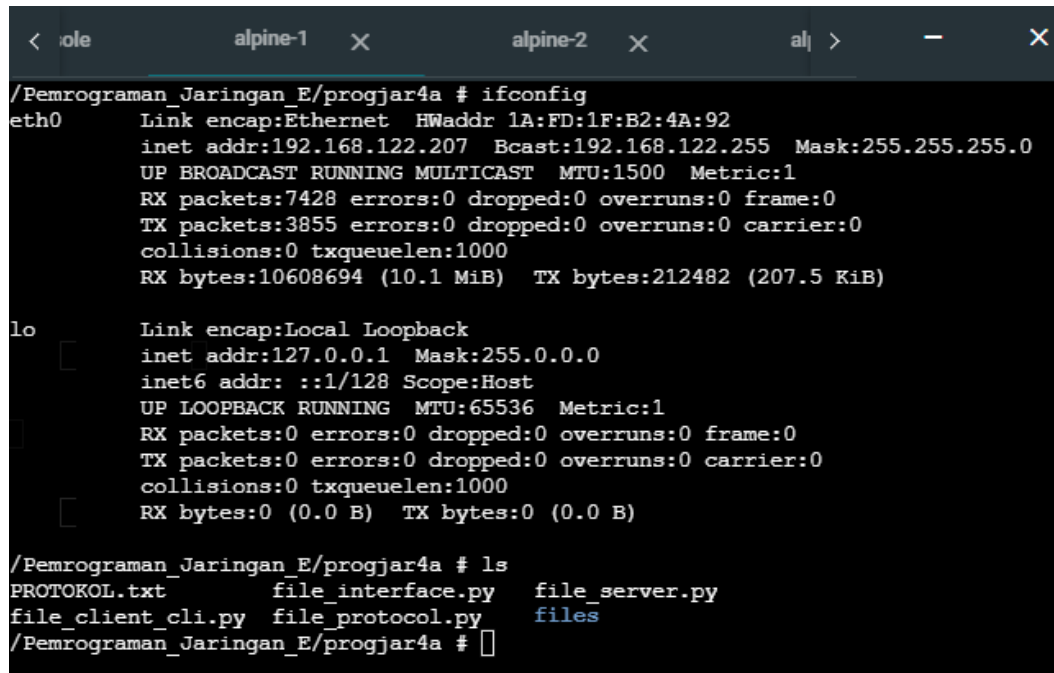
Praktikum :

Semua tugas berikut ini harus dijalankan di simulator

1. Jalankan program server seperti dalam pembahasan
2. Buatlah program client yang dapat melakukan 100 request get pada dalam satu saat untuk operasi get file "pokijan.jpg"
3. Capture dan submitlah poin 1 dan 2 dalam satu dokumen pdf. berikan deskripsi dan penjelasan

Langkah-langkah :

1. Load project yang sudah tersedia, kita bisa menggunakan project dengan 3 alpine
2. Jalankan command `git clone`
https://github.com/afiffadhilurrahman/Pemrograman_Jaringan_E.git pada seluruh alpine.
3. Cek ifconfig dan sesuaikan IP alpine 1 pada file `file_server.py`



```
< |ole      alpine-1  X      alpine-2  X      alpine-3  X      -      X
/Pemrograman_Jaringan_E/progjar4a # ifconfig
eth0      Link encap:Ethernet  HWaddr 1A:FD:1F:B2:4A:92
          inet addr:192.168.122.207  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7428 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3855 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10608694 (10.1 MiB)  TX bytes:212482 (207.5 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

/Pemrograman_Jaringan_E/progjar4a # ls
PROTOKOL.txt      file_interface.py  file_server.py
file_client_cli.py file_protocol.py   files
/Pemrograman_Jaringan_E/progjar4a #
```

```
GNU nano 4.6 file_server.py Modified
    hasil = fp.proses_string(d)
    hasil=hasil+"\r\n\r\n"
    self.connection.sendall(hasil.encode())
    else:
        break
    self.connection.close()

class Server(threading.Thread):
    def __init__(self,ipaddress='192.168.122.207',port=8889):
        self.ipinfo=(ipaddress,port)
        self.the_clients = []
        self.my_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.my_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
        threading.Thread.__init__(self)

    def run(self):
        logging.warning(f"server berjalan di ip address {self.ipinfo}")
        self.my_socket.bind(self.ipinfo)
        self.my_socket.listen(1)

^G Get Help ^C Write Out ^W Where Is ^R Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

```
GNU nano 4.6 file_server.py Modified
    def run(self):
        logging.warning(f"server berjalan di ip address {self.ipinfo}")
        self.my_socket.bind(self.ipinfo)
        self.my_socket.listen(1)
        while True:
            self.connection, self.client_address = self.my_socket.accept()
            logging.warning(f"connection from {self.client_address}")

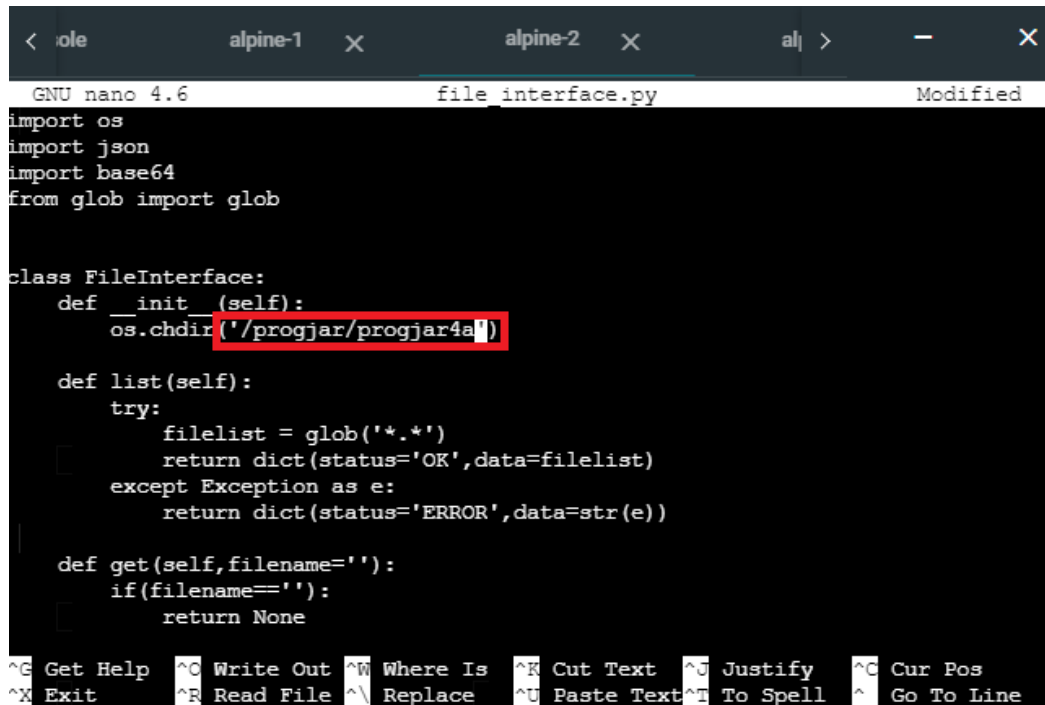
            clt = ProcessTheClient(self.connection, self.client_address)
            clt.start()
            self.the_clients.append(clt)

def main():
    svr = Server(ipaddress='192.168.122.207',port=6666)
    svr.start()

if __name__ == "__main__":
    main()

^G Get Help ^C Write Out ^W Where Is ^R Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

- Ubah directory yang terdapat pada file file_interface.py di alpine 2 (sesuaikan dengan path hasil clone).



```
GNU nano 4.6 file_interface.py Modified
import os
import json
import base64
from glob import glob

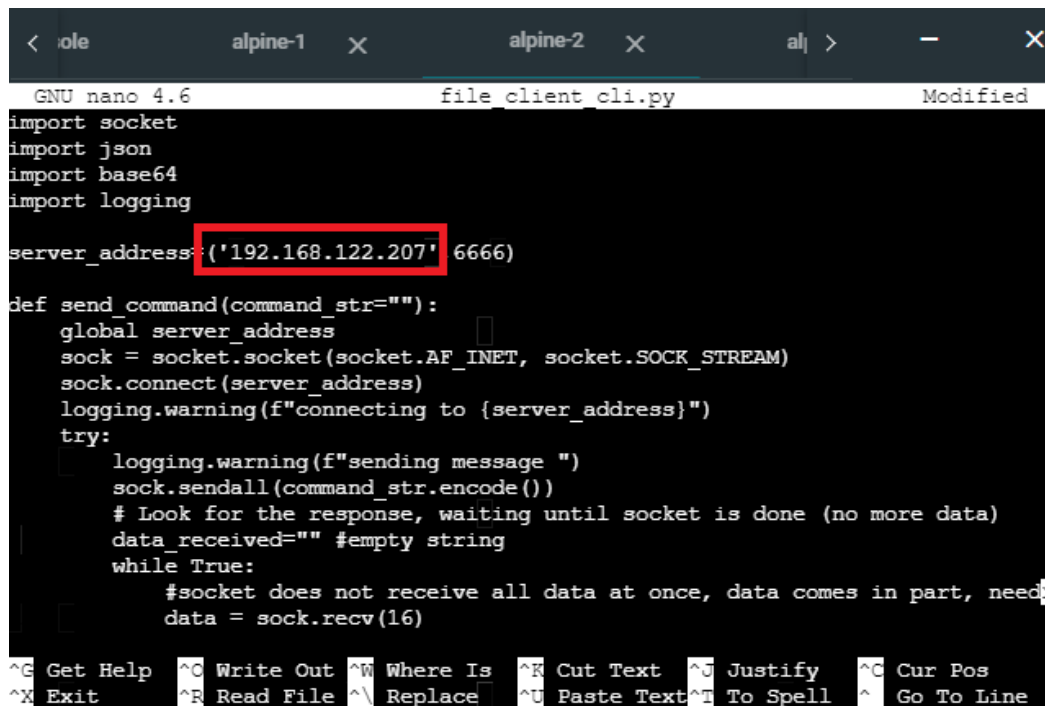
class FileInterface:
    def __init__(self):
        os.chdir('/progjar/progjar4a')

    def list(self):
        try:
            filelist = glob('*.*)
            return dict(status='OK',data=filelist)
        except Exception as e:
            return dict(status='ERROR',data=str(e))

    def get(self,filename=''):
        if(filename==''):
            return None

^G Get Help ^C Write Out ^W Where Is ^R Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

- Ubah IP address sesuai IP alpine 1 (server) pada file file_client_cli.py

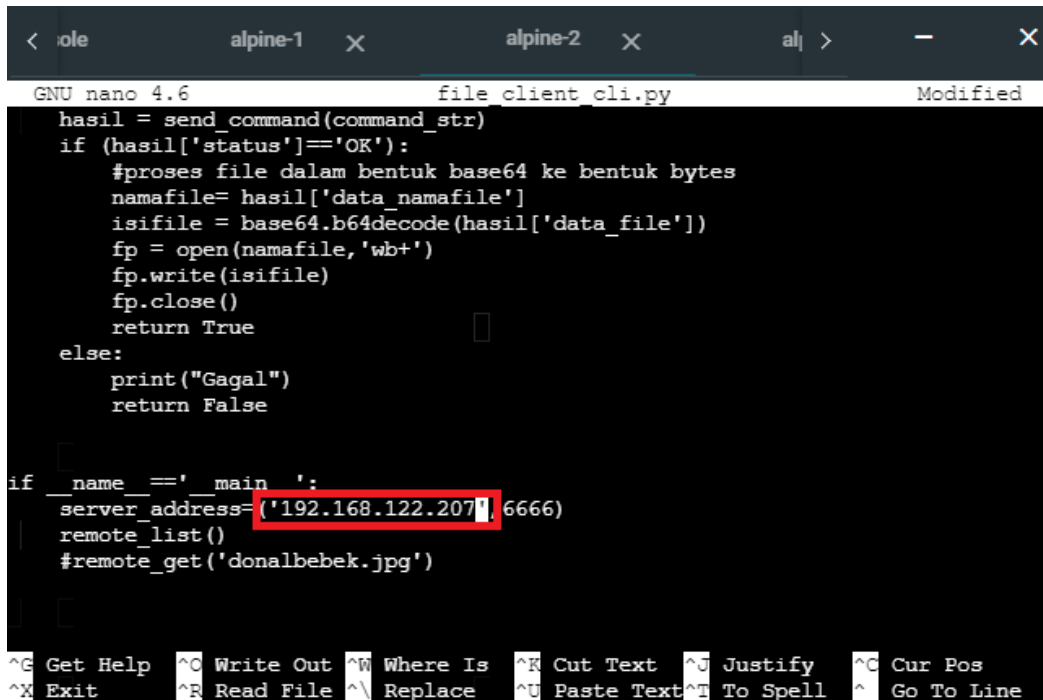


```
GNU nano 4.6 file_client_cli.py Modified
import socket
import json
import base64
import logging

server_address=('192.168.122.207' 6666)

def send_command(command_str=""):
    global server_address
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.connect(server_address)
    logging.warning(f"connecting to {server_address}")
    try:
        logging.warning(f"sending message ")
        sock.sendall(command_str.encode())
        # Look for the response, waiting until socket is done (no more data)
        data_received="" #empty string
        while True:
            #socket does not receive all data at once, data comes in part, need
            data = sock.recv(16)

^G Get Help ^C Write Out ^W Where Is ^R Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```



```
GNU nano 4.6 file_client_cli.py Modified
hasil = send_command(command_str)
if (hasil['status']=='OK'):
    #proses file dalam bentuk base64 ke bentuk bytes
    namafile= hasil['data_namafile']
    isifile = base64.b64decode(hasil['data_file'])
    fp = open(namafile,'wb+')
    fp.write(isifile)
    fp.close()
    return True
else:
    print("Gagal")
    return False

if __name__ == '__main__':
    server_address=('192.168.122.207', 6666)
    remote_list()
    #remote_get('donalbebek.jpg')
```

6. Buatlah file udp_multithread.py pada directory progjar4a di alpine 2

```
from file_client_cli import remote_get
import time
import datetime
import threading
import socket

def send_all():
    texec = dict()
    daftar = 'pokijan.jpg'
    catat_awal = datetime.datetime.now()
    for k in range(100):
        print(f"mengirim {k}")
        waktu = time.time()
        # bagian ini merupakan bagian yang menginstruksikan eksekusi fungsi send
        texec[k] = threading.Thread(target=remote_get, args=(daftar,))
        texec[k].start()

    # setelah menyelesaikan tugasnya, dikembalikan ke main thread dengan join
    for k in range(100):
        texec[k].join()

    catat_akhir = datetime.datetime.now()
```

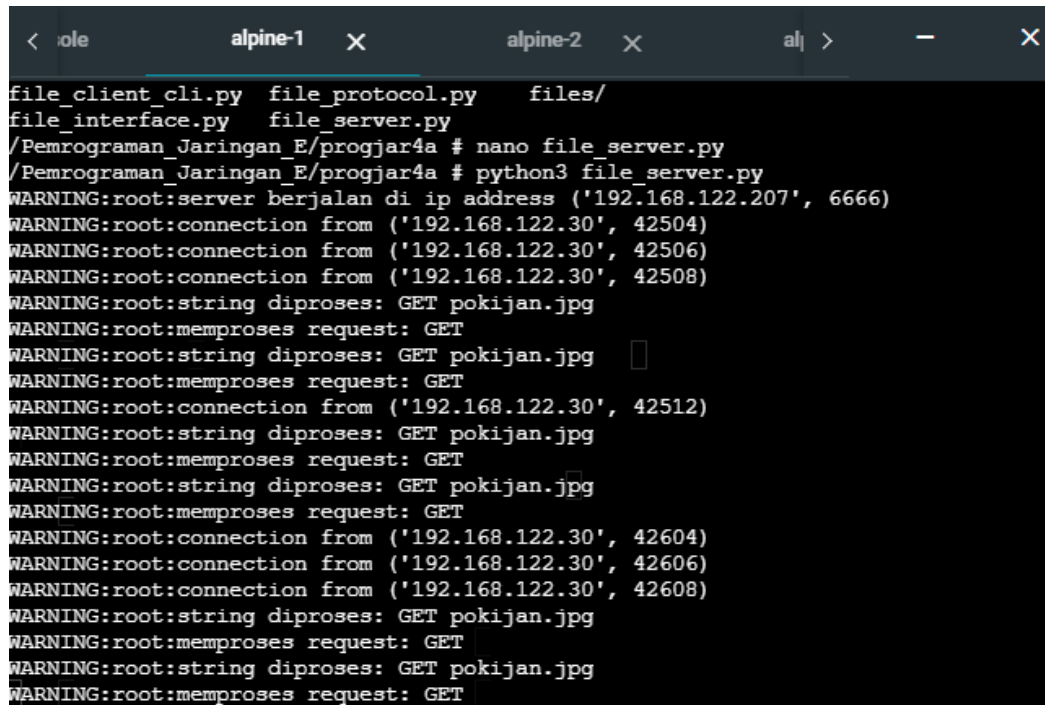
```

    selesai = catat_akhir - catat_awal
    print(f"Waktu TOTAL yang dibutuhkan {selesai} detik {catat_awal}
s/d{catat_akhir}")

# fungsi send all akan dijalankan secara multithreading
if __name__ == '__main__':
    send_all()

```

7. Jalankan udp_multithread.py
8. Hasil pada file server di alpine 1



```

< file_client_cli.py file_protocol.py files/
file_interface.py file_server.py
/Pemrograman_Jaringan_E/progjar4a # nano file_server.py
/Pemrograman_Jaringan_E/progjar4a # python3 file_server.py
WARNING:root:server berjalan di ip address ('192.168.122.207', 6666)
WARNING:root:connection from ('192.168.122.30', 42504)
WARNING:root:connection from ('192.168.122.30', 42506)
WARNING:root:connection from ('192.168.122.30', 42508)
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET
WARNING:root:connection from ('192.168.122.30', 42512)
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET
WARNING:root:connection from ('192.168.122.30', 42604)
WARNING:root:connection from ('192.168.122.30', 42606)
WARNING:root:connection from ('192.168.122.30', 42608)
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET
WARNING:root:string diproses: GET pokijan.jpg
WARNING:root:memproses request: GET

```

- [illegible]