Tugas 3
Concurrency

Oleh:

Feinard – 05111840000081

Progjar E

Fakultas Teknologi Elektro dan Informatika Cerdas

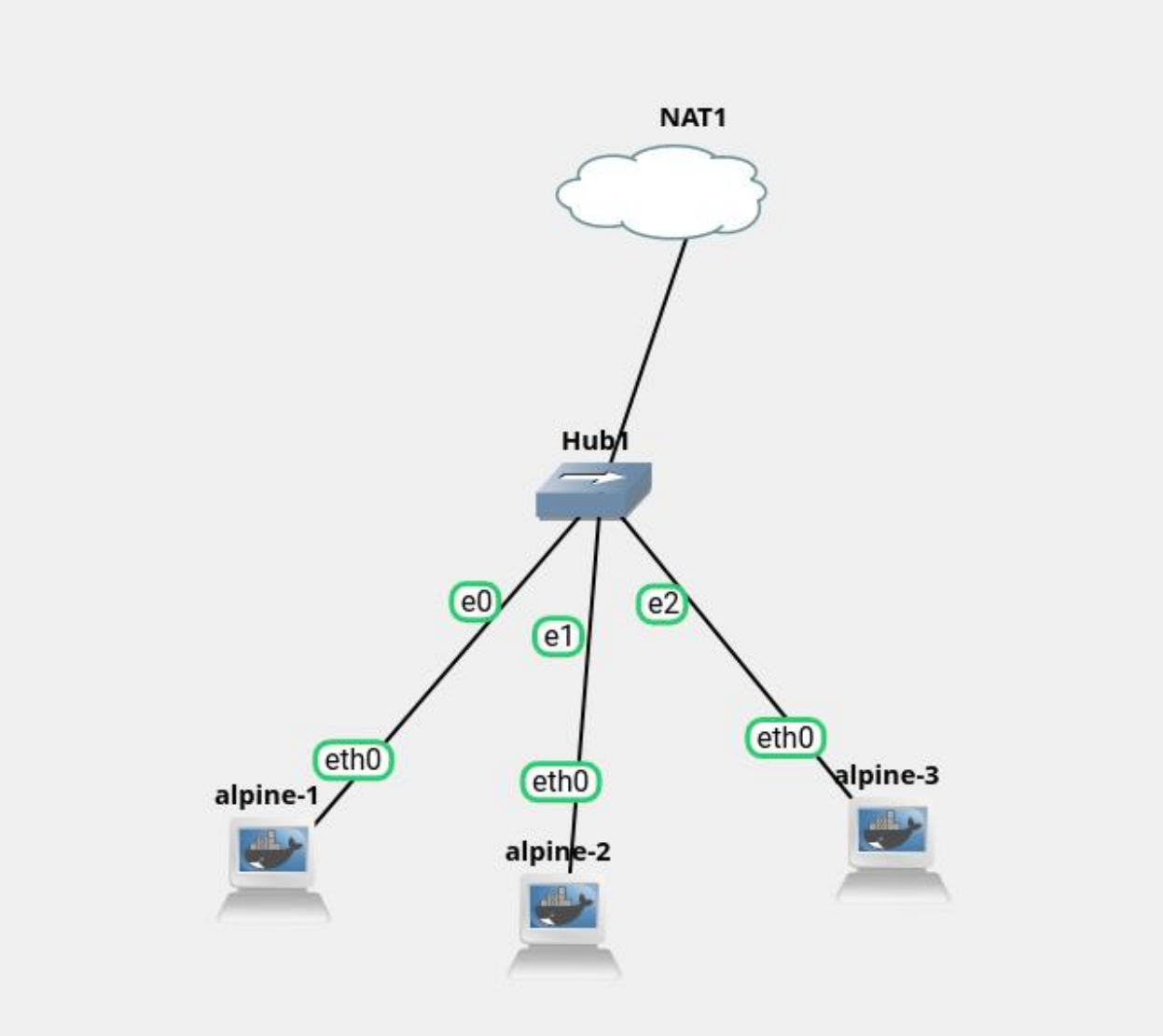
ITS Surabaya

2021

Deskripsi kasus:

Terdapat 2 client UDP, dimana setiap client ini dapat melakukan download file dengan cara multiprocess, multithreaded, asynchronous multiprocess, dan asynchronous multithreaded. Setiap client yang ada harus dapat mengirim ke suatu server. Server ini akan digunakan server multihreaded dan server multiprogramming. Tujuan dari kasus ini adalah untuk melihat performa antara tipe client yang ada, dengan total waktu yang digunakan oleh server untuk menerima data yang dikirim oleh client tersebut.

Gambar Arsitektur Sistem Jaringan:



Program yang dibuat:

1. Server Multithreaded

```
from socket import *
import socket
```

```
import threading
import logging
import time
import sys
from library import *
import random

class ProcessTheClient(threading.Thread):
    def __init__(self, connection, address):
        self.connection = connection
        self.address = address
        self.buffername = random.randint(65,90)
        self.buffername = 'buffer_' + chr(self.buffername)
        self.nama = 'server_'
        self.check = 0
        print('connection started')
        threading.Thread.__init__(self)

    def run(self):
        time_start = datetime.datetime.now()
        while True:
            try:
                incoming = self.connection.recv(32)
                # print(incoming)
                endcheck = incoming
                endcheck = endcheck[-3:]
                if endcheck == b'end' or endcheck == b'':
                    fp = open(self.buffername, 'ab')
                    fp.write(incoming[:-3])
                    fp.close()

                    print(self.buffername,'with',self.nama)
                    os.rename(self.buffername,self.nama)
                    time_end = datetime.datetime.now()
                    process_time = time_end - time_start
                    logging.warning(f'File {self.nama} Dimulai:{time_start}
selesai:{time_end} Lama:{process_time}')
                    return

                elif(self.check == 0):
                    try:
                        file_information = incoming.decode().split(" ")
                        self.nama += file_information[0].strip()
                        # print(self.nama)
                        self.check = 1
                    except Exception as e:
                        print(repr(e))

                fp = open(self.buffername,'ab')
                fp.write(incoming)
                fp.close()
            except Exception as e:
                print(repr(e))
        self.connection.close()

class Server(threading.Thread):
    def __init__(self):
```

```
        self.the_clients = []
        self.my_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        threading.Thread.__init__(self)

    def run(self):
        self.my_socket.bind(('0.0.0.0', 8989))
        self.my_socket.listen(10)
        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()
    svr.start()


if __name__ == "__main__":
    main()
```

2. Server MultiProcess

```
from socket import *
import socket
import multiprocessing
import logging
import time
import sys
from library import *
import random

def handle(self, connection, address ,buffername,filename):
    # import logging
    # logging.basicConfig(level=logging.DEBUG)
    # logger = logging.getLogger("process-%r" % (address,))
    self.connection = connection
    self.address = address
    self.check = 0
    time_start = datetime.datetime.now()
    while True:
        try:
            incoming = self.connection.recv(32)
            # print(incoming)
            # time.sleep(0.001)
            endcheck = incoming
            endcheck = endcheck[-3:]
            if endcheck == b'end' or endcheck == b'':
                fp = open(buffername, 'ab')
                fp.write(incoming[:-3])
                fp.close()

                print(buffername, 'with', filename)
```

```
                name = 'server_' + filename.decode()
                os.rename(buffername,name)
                time_end = datetime.datetime.now()
                process_time = time_end - time_start
                return

            elif incoming == "":
                logger.debug("Socket closed remotely")
                break

            fp = open(buffername, 'ab')
            fp.write(incoming)
            fp.close()
        except Exception as e:
            print(repr(e))

    self.connection.close()


class Server():
    def __init__(self):
        self.the_clients = []
        self.my_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    def start(self):
        self.my_socket.bind(('', 8989))
        self.my_socket.listen(10)
        while True:
            self.connection, self.client_address = self.my_socket.accept()
            logging.warning(f"connection from {self.client_address}")
            try:
                self.filename = self.connection.recv(32)
                self.buffername = random.randint(65, 90)
                self.buffername = 'buffer_' + chr(self.buffername)
                clt = multiprocessing.Process(target=handle, args=(self,
self.connection, self.client_address,self.buffername,self.filename))
                # clt.daemon = True
                self.the_clients.append(clt)
                clt.start()
            except:
                print('error')
                self.connection.close()

def main():
    svr = Server()
    svr.start()


if __name__ == "__main__":
    main()
```

3. Client Multithreaded

```
from library import download_gambar,get_url_list
import time
import datetime
```

```
import threading

def download_semua():
    texec = dict()
    urls = get_url_list()

    catat_awal = datetime.datetime.now()
    for k in urls:
        print(f"mendownload {urls[k]}")
        waktu = time.time()
        #bagian ini merupakan bagian yang mengistruksikan eksekusi fungsi
download gambar secara multithread
        texec[k] = threading.Thread(target=download_gambar, args=(urls[k],))
        texec[k].start()
        # texec[k].join()

    #setelah menyelesaikan tugasnya, dikembalikan ke main thread dengan join
    for k in urls:
        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 download_gambar akan dijalankan secara multithreading

if __name__=='__main__':
    download_semua()
```

4. Client Multiprocess

```
from library import download_gambar, get_url_list
import time
import datetime
from multiprocessing import Process

def download_semua():
    texec = dict()
    urls = get_url_list()
    catat_awal = datetime.datetime.now()
    for k in urls:
        print(f"mendownload {urls[k]}")
        waktu = time.time()
        #bagian ini merupakan bagian yang mengistruksikan eksekusi fungsi
download gambar secara multiprocess
        texec[k] = Process(target=download_gambar, args=(urls[k],))
        texec[k].start()
    #setelah menyelesaikan tugasnya, dikembalikan ke main process dengan join
    for k in urls:
        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 download_gambar akan dijalankan secara multi process
if __name__=='__main__':
    download_semua()
```

5. Client Asynchronous Multithreaded

```
from library import download_gambar,get_url_list
import time
import datetime
import concurrent.futures




def download_semua():
    texec = dict()
    urls = get_url_list()
    status_task = dict()
    task = concurrent.futures.ThreadPoolExecutor(max_workers=4)
    catat_awal = datetime.datetime.now()
    for k in urls:
        print(f"mendownload {urls[k]}")
        waktu = time.time()
        #bagian ini merupakan bagian yang mengistruksikan eksekusi fungsi
download gambar secara multithread
        texec[k] = task.submit(download_gambar, urls[k])

    #setelah menyelesaikan tugasnya, dikembalikan ke main thread dengan
memanggil result
    for k in urls:
        status_task[k]=texec[k].result()

    catat_akhir = datetime.datetime.now()
    selesai = catat_akhir - catat_awal
    print(f"Waktu TOTAL yang dibutuhkan {selesai} detik {catat_awal} s/d
{catat_akhir}")
    print("hasil task yang dijalankan")
    print(status_task)


#fungsi download_gambar akan dijalankan secara multithreading

if __name__=='__main__':
    download_semua()
```

6. Client Asynchronous Multiprocess

```
from library import download_gambar, get_url_list
import time
import datetime
from multiprocessing import Process, Pool

def download_semua():
    texec = dict()
    urls = get_url_list()
```

```
    status_task = dict()
    task_pool = Pool(processes=20) #2 task yang dapat dikerjakan secara
simultan, dapat diset sesuai jumlah core
    catat_awal = datetime.datetime.now()
    for k in urls:
        print(f"mendownload {urls[k]}")
        #bagian ini merupakan bagian yang mengistruksikan eksekusi fungsi
download gambar secara multiprocess
        texec[k] = task_pool.apply_async(func=download_gambar,
args=(urls[k],))

    #setelah menyelesaikan tugasnya, dikembalikan ke main process dengan
mengambil hasilnya dengan get
    for k in urls:
        status_task[k]=texec[k].get(timeout=10)

    catat_akhir = datetime.datetime.now()
    selesai = catat_akhir - catat_awal
    print(f"Waktu TOTAL yang dibutuhkan {selesai} detik {catat_awal} s/d
{catat_akhir}")
    print("status TASK")
    print(status_task)


#fungsi download_gambar akan dijalankan secara multi process

if __name__=='__main__':
    download_semua()
```

Library:

```
import logging
import requests
import os
import time
import datetime
import socket


def get_url_list():
    urls = dict()

urls['kompas']='https://asset.kompas.com/crops/qz_jJxyaZgGgboomdCEXsfbSpec=/0
x0:998x665/740x500/data/photo/2020/03/01/5e5b52f4db896.jpg'
    urls['its']='https://www.its.ac.id/wp-content/uploads/2017/10/logo-its-
1.png'

urls['detik']='https://akcdn.detik.net.id/community/media/visual/2021/04/22/d
etikcom-ramadan-desktop-1.gif'
    urls['file1']='https://file-examples-
com.github.io/uploads/2018/04/file_example_MOV_480_700kB.mov'
    # urls['file2']='https://file-examples-
com.github.io/uploads/2018/04/file_example_MOV_1280_1_4MB.mov'
    urls['file3']='https://file-examples-
com.github.io/uploads/2017/02/zip_2MB.zip'
    return urls
```

```
def download_gambar(url=None,tuliskefile=True):
    waktu_awal = datetime.datetime.now()
    if (url is None):
        return False
    ff = requests.get(url)
    tipe = dict()
    tipe['image/png']='png'
    tipe['image/jpg']='jpg'
    tipe['image/gif']='gif'
    tipe['image/jpeg']='jpg'
    tipe['application/zip']='jpg'
    tipe['video/quicktime']='mov'

    content_type = ff.headers['Content-Type']
    logging.warning(content_type)
    if (content_type in list(tipe.keys())):
        namafile = os.path.basename(url)
        ekstensi = tipe[content_type]
        if (tuliskefile):
            fp = open(f"{namafile}","wb")
            fp.write(ff.content)
            fp.close()
        waktu_process = datetime.datetime.now() - waktu_awal
        waktu_akhir =datetime.datetime.now()
        logging.warning(f"writing {namafile}.{ekstensi} dalam waktu
{waktu_process} {waktu_awal} s/d {waktu_akhir} \n Sending File now")
        return send(namafile)
    else:
        return False

def send(filename):
    waktu_sekarang = datetime.datetime.now()
    host = '127.0.0.1'
    port = 8989
    address = (host,port)
    sock = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    sock.connect(address)
    time.sleep(2)

    fp = open(f'{filename}', 'rb')
    # print('nama file',filename.encode())
    sock.send(filename.encode())
    time.sleep(1)

    # send file
    while True:
        data_buffer = fp.read(32)
        # print(data_buffer)
        if not data_buffer:
            break
        sock.sendall(data_buffer)

    sock.send('\x00end'.encode())
    sock.close()
    fp.close()
```

```
    waktu_akhir = datetime.datetime.now()
    total_process = waktu_akhir - waktu_sekarang
    logging.warning(f"sending dalam waktu {waktu_sekarang}  s/d {waktu_akhir}
dengan process selama {total_process}")
    return

if __name__=='__main__':
    #check fungsi
    k =
download_gambar('https://asset.kompas.com/crops/qz_jJxyaZgGgboomdCEXsfbSpec=/
0x0:998x665/740x500/data/photo/2020/03/01/5e5b52f4db896.jpg')
    print(k)
```

Hasil:

Contoh dengan server multithread dan client multiprocess

```
/Pemprograman_Jaringan_E/progjar3/Tugas 3 # python3 multi_process.py
mendownload https://asset.kompas.com/crops/qz_jJxyaZgGgboomdCEXsfbSpec=/0x0:998x665/740x500/data/photo/2020/03/01/5e5b52f4db896.jpg
mendownload https://www.its.ac.id/wp-content/uploads/2017/10/logo-its-1.png
mendownload https://akcdn.detik.net.id/community/media/visual/2021/04/22/detikcom-ramadan-desktop-1.gif
mendownload https://file-examples-com.github.io/uploads/2018/04/file_example_MOV_480_700kB.mov
mendownload https://file-examples-com.github.io/uploads/2017/02/zip_2MB.zip
WARNING:root:image/jpeg
WARNING:root:writing 5e5b52f4db896.jpg.jpg dalam waktu 0:00:00.214464 2021-06-10 13:03:44.771553 s/d 2021-06-10 13:03:44.986022
 Sending File now
WARNING:root:image/png
WARNING:root:writing logo-its-1.png.png dalam waktu 0:00:00.313214 2021-06-10 13:03:44.773760 s/d 2021-06-10 13:03:45.086978
 Sending File now
WARNING:root:image/gif; charset="utf-8"
WARNING:root:video/quicktime
WARNING:root:writing file_example_MOV_480_700kB.mov.mov dalam waktu 0:00:02.093198 2021-06-10 13:03:44.787160 s/d 2021-06-10 13:03:46.880362
 Sending File now
WARNING:root:application/zip
WARNING:root:writing zip_2MB.zip.jpg dalam waktu 0:00:02.352245 2021-06-10 13:03:44.796190 s/d 2021-06-10 13:03:47.148440
 Sending File now
WARNING:root:sending dalam waktu 2021-06-10 13:03:44.987407  s/d 2021-06-10 13:03:48.006745 dengan process selama 0:00:03.019338
WARNING:root:sending dalam waktu 2021-06-10 13:03:45.087183  s/d 2021-06-10 13:03:48.093385 dengan process selama 0:00:03.006202
WARNING:root:sending dalam waktu 2021-06-10 13:03:46.880484  s/d 2021-06-10 13:03:50.786059 dengan process selama 0:00:03.905575
WARNING:root:sending dalam waktu 2021-06-10 13:03:47.149879  s/d 2021-06-10 13:03:52.698556 dengan process selama 0:00:05.548677
Waktu TOTAL yang dibutuhkan 0:00:07.945842 detik 2021-06-10 13:03:44.755869 s/d 2021-06-10 13:03:52.701711
```

Server multithread

```
/Pemprograman_Jaringan_E/progjar3/Tugas 3 # python3 server.py
WARNING:root:connection from ('192.168.122.170', 41954)
connection started
WARNING:root:connection from ('192.168.122.170', 41956)
connection started
WARNING:root:connection from ('192.168.122.170', 41958)
connection started
WARNING:root:connection from ('192.168.122.170', 41960)
connection started
buffer_C with server_5e5b52f4db896.jpg
WARNING:root:File server_5e5b52f4db896.jpg Dimulai:2021-06-10 13:03:44.989849 selesai:2021-06-10 13:03:48.102496 Lama:0:00:03.112647
buffer_Y with server_logo-its-1.png
WARNING:root:File server_logo-its-1.png Dimulai:2021-06-10 13:03:45.090327 selesai:2021-06-10 13:03:48.114317 Lama:0:00:03.023990
buffer_L with server_file_example_MOV_480_700kB.mov
WARNING:root:File server_file_example_MOV_480_700kB.mov Dimulai:2021-06-10 13:03:46.883457 selesai:2021-06-10 13:03:51.307692 Lama:0:00:04.424235
buffer_I with server_zip_2MB.zip
WARNING:root:File server_zip_2MB.zip Dimulai:2021-06-10 13:03:47.151582 selesai:2021-06-10 13:03:52.948416 Lama:0:00:05.796834
```

Contoh menggunakan server multithread dengan client multithread

Client:

```
/Pemprograman_Jaringan_E/progjar3/Tugas 3 # python3 multi_thread.py
mendownload https://asset.kompas.com/crops/qz_jJxyaZgGgboomdCEXsfbSpec=/0x0:998x665/740x500/data/photo/2020/03/01/5e5b52f4db896.jpg
mendownload https://www.its.ac.id/wp-content/uploads/2017/10/logo-its-1.png
mendownload https://akcdn.detik.net.id/community/media/visual/2021/04/22/detikcom-ramadan-desktop-1.gif
mendownload https://file-examples-com.github.io/uploads/2018/04/file_example_MOV_480_700kB.mov
mendownload https://file-examples-com.github.io/uploads/2017/02/zip_2MB.zip
WARNING:root:image/jpeg
WARNING:root:writing 5e5b52f4db896.jpg.jpg dalam waktu 0:00:00.142408 2021-06-10 13:06:27.700400 s/d 2021-06-10 13:06:27.842813
 Sending File now
WARNING:root:image/png
WARNING:root:writing logo-its-1.png.png dalam waktu 0:00:00.189599 2021-06-10 13:06:27.706882 s/d 2021-06-10 13:06:27.896485
 Sending File now
WARNING:root:sending dalam waktu 2021-06-10 13:06:27.843013  s/d 2021-06-10 13:06:30.885956 dengan process selama 0:00:03.042943
WARNING:root:sending dalam waktu 2021-06-10 13:06:27.896660  s/d 2021-06-10 13:06:30.904280 dengan process selama 0:00:03.007620
WARNING:root:image/gif; charset="utf-8"
WARNING:root:video/quicktime
WARNING:root:writing file_example_MOV_480_700kB.mov.mov dalam waktu 0:00:07.040541 2021-06-10 13:06:27.717853 s/d 2021-06-10 13:06:34.758398
 Sending File now
WARNING:root:application/zip
WARNING:root:writing zip_2MB.zip.jpg dalam waktu 0:00:07.195060 2021-06-10 13:06:27.723311 s/d 2021-06-10 13:06:34.918376
 Sending File now
WARNING:root:sending dalam waktu 2021-06-10 13:06:34.758608  s/d 2021-06-10 13:06:38.163689 dengan process selama 0:00:03.405081
WARNING:root:sending dalam waktu 2021-06-10 13:06:34.920693  s/d 2021-06-10 13:06:40.522338 dengan process selama 0:00:05.601645
Waktu TOTAL yang dibutuhkan 0:00:12.825752 detik 2021-06-10 13:06:27.698839 s/d 2021-06-10 13:06:40.524591
```

Server:

```
WARNING:root:connection from ('192.168.122.170', 41966)
connection started
WARNING:root:connection from ('192.168.122.170', 41968)
connection started
buffer_T with server_logo-its-1.png
WARNING:root:File server_logo-its-1.png Dimulai:2021-06-10 13:06:27.898061 selesai:2021-06-10 13:06:30.922912 Lama:0:00:03.024851
buffer_Q with server_5e5b52f4db896.jpg
WARNING:root:File server_5e5b52f4db896.jpg Dimulai:2021-06-10 13:06:27.844414 selesai:2021-06-10 13:06:30.961000 Lama:0:00:03.116586
WARNING:root:connection from ('192.168.122.170', 41976)
connection started
WARNING:root:connection from ('192.168.122.170', 41978)
connection started
buffer_K with server_file_example_MOV_480_700kB.mov
WARNING:root:File server_file_example_MOV_480_700kB.mov Dimulai:2021-06-10 13:06:34.766493 selesai:2021-06-10 13:06:39.252269 Lama:0:00:04.485776
buffer_K with server_zip_2MB.zip
WARNING:root:File server_zip_2MB.zip Dimulai:2021-06-10 13:06:34.922602 selesai:2021-06-10 13:06:40.765552 Lama:0:00:05.842950
```

Contoh implementasi server multiprocess dengan client asynchronous

Client:

```
/Pemprograman_Jaringan_E/progjar3/Tugas 3 # python3 multi_thread_async.py
mendownload https://asset.kompas.com/crops/qz_jJxyaZgGgboomdCEXsfbSpec=/0x0:998x665/740x500/data/photo/2020/03/01/5e5b52f4db896.jpg
mendownload https://www.its.ac.id/wp-content/uploads/2017/10/logo-its-1.png
mendownload https://akcdn.detik.net.id/community/media/visual/2021/04/22/detikcom-ramadan-desktop-1.gif
mendownload https://file-examples-com.github.io/uploads/2018/04/file_example_MOV_480_700kB.mov
mendownload https://file-examples-com.github.io/uploads/2017/02/zip_2MB.zip
WARNING:root:image/jpeg
WARNING:root:writing 5e5b52f4db896.jpg.jpg dalam waktu 0:00:00.232393 2021-06-10 13:08:45.012283 s/d 2021-06-10 13:08:45.244680
 Sending File now
WARNING:root:image/png
WARNING:root:writing logo-its-1.png.png dalam waktu 0:00:00.255424 2021-06-10 13:08:45.024215 s/d 2021-06-10 13:08:45.279644
 Sending File now
WARNING:root:image/gif; charset="utf-8"
WARNING:root:video/quicktime
WARNING:root:writing file_example_MOV_480_700kB.mov.mov dalam waktu 0:00:01.641010 2021-06-10 13:08:45.042583 s/d 2021-06-10 13:08:46.683597
 Sending File now
WARNING:root:application/zip
WARNING:root:writing zip_2MB.zip.jpg dalam waktu 0:00:00.651416 2021-06-10 13:08:46.615300 s/d 2021-06-10 13:08:47.266722
 Sending File now
WARNING:root:sending dalam waktu 2021-06-10 13:08:45.244908  s/d 2021-06-10 13:08:48.260714 dengan process selama 0:00:03.015806
WARNING:root:sending dalam waktu 2021-06-10 13:08:45.279895  s/d 2021-06-10 13:08:48.289980 dengan process selama 0:00:03.010085
WARNING:root:sending dalam waktu 2021-06-10 13:08:46.683721  s/d 2021-06-10 13:08:50.173699 dengan process selama 0:00:03.489978
WARNING:root:sending dalam waktu 2021-06-10 13:08:47.270113  s/d 2021-06-10 13:08:52.398595 dengan process selama 0:00:05.128482
Waktu TOTAL yang dibutuhkan 0:00:07.392001 detik 2021-06-10 13:08:45.009546 s/d 2021-06-10 13:08:52.401547
hasil task yang dijalankan
{'kompas': None, 'its': None, 'detik': False, 'file1': None, 'file3': None}
```

```
/Pemprograman_Jaringan_E/progjar3/Tugas 3 # python3 multi_process_async.py
mendownload https://asset.kompas.com/crops/qz_jJxyaZgGgboomdCEXsfbSpec=/0x0:998x665/740x500/data/photo/2020/03/01/5e5b52f4db896.jpg
mendownload https://www.its.ac.id/wp-content/uploads/2017/10/logo-its-1.png
mendownload https://akcdn.detik.net.id/community/media/visual/2021/04/22/detikcom-ramadan-desktop-1.gif
mendownload https://file-examples-com.github.io/uploads/2018/04/file_example_MOV_480_700kB.mov
mendownload https://file-examples-com.github.io/uploads/2017/02/zip_2MB.zip
WARNING:root:image/png
WARNING:root:writing logo-its-1.png.png dalam waktu 0:00:00.762647 2021-06-10 13:09:32.569353 s/d 2021-06-10 13:09:33.332004
 Sending File now
WARNING:root:image/gif; charset="utf-8"
WARNING:root:video/quicktime
WARNING:root:writing file_example_MOV_480_700kB.mov.mov dalam waktu 0:00:02.022012 2021-06-10 13:09:32.576223 s/d 2021-06-10 13:09:34.598239
 Sending File now
WARNING:root:image/jpeg
WARNING:root:writing 5e5b52f4db896.jpg.jpg dalam waktu 0:00:02.331455 2021-06-10 13:09:32.567324 s/d 2021-06-10 13:09:34.898783
 Sending File now
WARNING:root:application/zip
WARNING:root:writing zip_2MB.zip.jpg dalam waktu 0:00:02.481699 2021-06-10 13:09:32.578246 s/d 2021-06-10 13:09:35.059953
 Sending File now
WARNING:root:sending dalam waktu 2021-06-10 13:09:33.332193  s/d 2021-06-10 13:09:36.334956 dengan process selama 0:00:03.002763
WARNING:root:sending dalam waktu 2021-06-10 13:09:34.898961  s/d 2021-06-10 13:09:37.909562 dengan process selama 0:00:03.010601
WARNING:root:sending dalam waktu 2021-06-10 13:09:34.598500  s/d 2021-06-10 13:09:38.388647 dengan process selama 0:00:03.790147
WARNING:root:sending dalam waktu 2021-06-10 13:09:35.062974  s/d 2021-06-10 13:09:40.334667 dengan process selama 0:00:05.271693
Waktu TOTAL yang dibutuhkan 0:00:07.775952 detik 2021-06-10 13:09:32.562857 s/d 2021-06-10 13:09:40.338809
status TASK
{'kompas': None, 'its': None, 'detik': False, 'file1': None, 'file3': None}
```

Server:

```
/Pemprograman_Jaringan_E/progjar3/Tugas 3 # python3 server_multiprocess.py
WARNING:root:connection from ('192.168.122.170', 42012)
WARNING:root:connection from ('192.168.122.170', 42014)
WARNING:root:connection from ('192.168.122.170', 42018)
buffer_K with b'logo-its-1.png'
buffer_Y with b'5e5b52f4db896.jpg'
WARNING:root:connection from ('192.168.122.170', 42020)
buffer_P with b'file_example_MOV_480_700kB.mov'
buffer_M with b'zip_2MB.zip'
WARNING:root:connection from ('192.168.122.170', 42046)
WARNING:root:connection from ('192.168.122.170', 42048)
buffer_R with b'logo-its-1.png'
WARNING:root:connection from ('192.168.122.170', 42050)
WARNING:root:connection from ('192.168.122.170', 42052)
buffer_S with b'5e5b52f4db896.jpg'
buffer_A with b'file_example_MOV_480_700kB.mov'
buffer_U with b'zip_2MB.zip'
```

Hasil semua program yang dijalankan adalah server_”nama file lama”

Contoh hasil yang didapat dari program server:

```
/Pemprograman_Jaringan_E/progjar3/Tugas 3 # ls
__pycache__                          server_5e5b52f4db896.jpg
library.py                           server_file_example_MOV_480_700kB.mov
multi_process.py                     server_logo-its-1.png
multi_process_async.py               server_multiprocess.py
multi_thread.py                      server_zip_2MB.zip
multi_thread_async.py                single_thread.py
server.py
```