**BÁO CÁO ĐỒ ÁN**

**Họ và tên:** Phạm Ngọc Tâm

**Giảng viên hướng dẫn**: Nguyễn Thanh Hoà

**Khoa:** Mạng Máy Tính và Truyền Thông

**I.Nội dung báo cáo**

**1.Thành phần source code:**

- Gồm 2 file Client.go và Server.go. Client.go thực hiện các chức năng ở phía Client và Server.go đảm nhiệm ở Server.

+ **Client.go**

Gồm 2 hàm chính là:

func main() {

    Option, Path, IntervalTime, AllTime := Input()

    Client(Option, Path, IntervalTime, AllTime)

}

**Input()** : sẽ nhận thông số user nhập vào và trả về 4 tham số: Option, Path, Interval Time, AllTime.

func Input() (string, string, string, string) {

    scanner := bufio.NewScanner(os.Stdin)

    fmt.Println("Enter option: ")

    scanner.Scan()

    Option := scanner.Text()

// Path, IntervalTime, AllTime same Option

    fmt.Println("-------------------------")

    return Option, Path, IntervalTime, AllTime

}

**Client()** : Nhận vào 4 tham số đầu ra từ Input(), thực hiện việc lấy hình ảnh từ đường dẫn Path, tạo request và gửi lên Server.

func Client(Option string, Path string, Time string, AllTime string) {

    Time1, \_ := time.ParseDuration(Time + "s")

    AllTime1, \_ := time.ParseDuration(AllTime + "s")

    Start := time.Now()

    FileCount := CountFile(Path)

    for {

        if time.Since(Start) > AllTime1 {

            break

        }

        if Option == "1" {

            fmt.Println("Option 1")

            sendSequenceRequests(FileCount, Path)

        } else if Option == "2" {

            fmt.Println("Option 2")

            sendParallelRequests(FileCount, Path)

        }

        time.Sleep(Time1)

        continue

    }

}

Option 1: Cách gửi tuần tự

func sendSequenceRequests(count int, Path string) {

    // Make a byte writer

    var bodyRequest bytes.Buffer

    // Make Multipart Writer of BodyRequest

    MultiWriter := multipart.NewWriter(&bodyRequest)

    for i := 1; i <= count; i++ {

        FileImageUp, \_ := os.Open(Path + "/" + ConvItoStr(i) + ".jpg")

        defer FileImageUp.Close()

        //Make an Form Upload Field

        WriterFile, \_ := MultiWriter.CreateFormFile("FileUpload", "image"+ConvItoStr(i)+".jpg") // ConvItoStr(): Convert int to String

        // Copy FileImageUp into WriterFile

        \_, err := io.Copy(WriterFile, FileImageUp)

        if err != nil {

            log.Fatal(err)

        }

    }

    MultiWriter.Close()

    MakeRequest(MultiWriter, bodyRequest)

}

Option 2: Cách gửi song song, thay thế vòng for ở Option 1 thành

func sendParallelRequest(count int, Path string) {

    for i := 1; i <= count; i++ {

        go func(u int) {

            //..

            FileImageUp, \_ := os.Open(Path + "/" + ConvItoStr(u) + ".jpg")

            defer FileImageUp.Close()

            //Make an Form Upload Field

            WriterFile, \_ := MultiWriter.CreateFormFile("FileUpload", "image"+ConvItoStr(u)+".jpg")

            // Copy FileImageUp into WriterFile

            \_, err := io.Copy(WriterFile, FileImageUp)

            if err != nil {

                log.Fatal(err)

            }

            //Send Request (same option 1) …

        }(i)

        time.Sleep(time.Second)

    }

}

Trong đó Hàm MakeRequest sẽ nhận dữ liệu bodyRequest, đóng gói lại và gửi lên Server

func MakeRequest(MultiWriter \*multipart.Writer, bodyRequest bytes.Buffer) {

    // Make request send to Server

    Req, \_ := http.NewRequest("POST", "http://localhost:12345/MultiFile", &bodyRequest)

    Req.Header.Set("Content-Type", MultiWriter.FormDataContentType())

    //Create Client Object

    ClientObject := &http.Client{}

    Rep, \_ := ClientObject.Do(Req)

    defer Rep.Body.Close()

    ByteFile, \_ := ioutil.ReadAll(Rep.Body)

    fmt.Println(string(ByteFile))

}

+ **Server.go**

func main() {

    http.HandleFunc("/MultiFile", GetMultiFile)

    http.ListenAndServe(":12345", nil)

}

func GetMultiFile(Rep http.ResponseWriter, Req \*http.Request) {

    Req.ParseMultipartForm(10 \* 1024 \* 1024) // ParseMultipartForm parse the submited MultipartForm

    fmt.Fprintln(Rep, "------------------------")

    for key, value := range Req.MultipartForm.File {

        if key == "FileUpload" {

            for \_, MultiFile := range value {

                SavetoFileLog(MultiFile, Rep) // Save to file log in Server

                fmt.Fprintln(Rep, "------------------------")

            }

        }

    }

    fmt.Fprintln(Rep, "Sucessful!...")

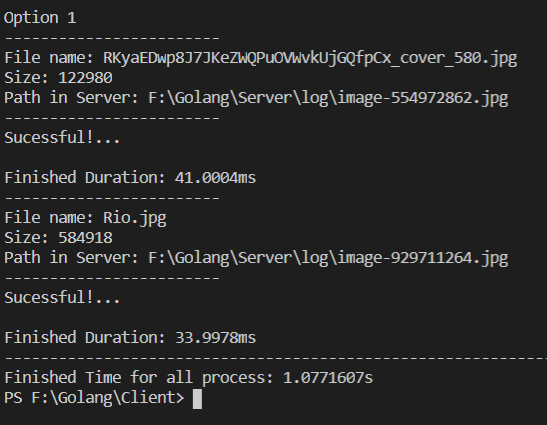
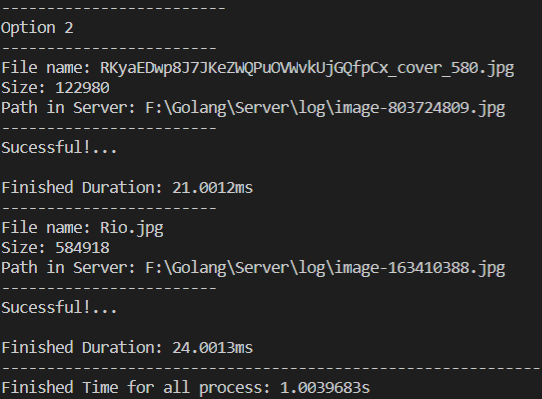
}

Server thực việc nhận MultipartForm từ Client và lưu lại vào file log

**2.Kết quả thực nghiệm**

Để trong folder Image ở Client 3 tấm hình bất kì

* Nhập input: Option bất kì( 1 hoặc 2), Path: “F:\\Golang\\Client\\Image”, Time = 1s, và AllTime = 2s
* Run Server.go, sau đó Run Client.go

**Kết quả:**

-----Hết-----