

Proiect programare concurenta si distribuita

Iulian Cimpan, Teodor Branescu, Sebastian Carabasiu, Bogdan Petri, Adrian Bordeianu

2020

Cuprins

1	Introducere	3
2	Descriere	3
3	Tehnologii folosite	3
3.1	C server	3
3.2	C client	4
3.3	Scripturi Python	4
3.4	Rotire imagine	4
3.5	Solarize	5
3.6	Posterize	5
3.7	Invert	5
3.8	Expend	5
3.9	Transformarea imaginii in binar	5
4	Directii viitoare de dezvoltare	6

1 Introducere

Proiectul pe care l-am implementat are rolul de a transfera imagini de la un calculator la altul asigurand prelucrarea acestora prin intermediul unor scripturi Python folosind biblioteca Pillow.

2 Descriere

In dezvoltarea aplicatiei am avut ca obiective principale:

1. Crearea unui server in C
2. Crearea unui client in C
3. Introducerea de scripturi Python embedded in C pentru prelucrarea imaginilor

3 Tehnologii folosite

3.1 C server

Serverul are rolul de a primi fisierele de la client. In implementarea lui am folosit socket-uri_INET si protocolul TCP. Pentru server folosim biblioteci pentru:

1. time
2. sockets
3. types
4. etc.

Exemplu de functie folosita pentru crearea unui socket:

```
int createSocket()
{
    int fd = socket(AF_INET, SOCK_STREAM, 0);
    if (fd < 0)
    {
        fprintf(stderr, "Eroare: %s\n", strerror(errno));
        exit(-1);
    }
    return fd;
}

//Legare
void socketBinding(int fd, struct sockaddr_in *server, int portNumber)
{
    memset(server, 0, sizeof(struct sockaddr_in));
    server->sin_family = AF_INET;
    server->sin_port = htons(portNumber);
    server->sin_addr.s_addr = htonl(INADDR_ANY);
    if ((bind(fd, (struct sockaddr *)server, sizeof(servAddr)) < 0))
```

```
{  
    perror("Legare esuata");  
    exit(-1);  
}  
}
```

3.2 C client

Clientul are rolul de a transmite comenzi serverului, dintre acestea putem enumera:

1. cd - Pentru modificarea directorului in care se situeaza clientul.
2. rcd - Pentru modificarea directorului in care se situeaza serverul.
3. ls - Pentru afisarea fisierelor din folderul clientului.
4. rls - Pentru afisarea fisierelor din folderul serverului.
5. show nume fisier - Afiseaza continutul unui fisier.
6. cng nume imagine nume functie - Modifica imaginea in functie de metoda apelata.

Pentru a rula clientul trebuie sa specificam: adresa ip de destinatie. De asemenea si pentru client folosim socket-uri INET si protocolul TCP de transmisie a fisierelor. Biblioteci folosite:

1. time
2. sockets
3. types
4. Python.h
5. etc.

3.3 Scripturi Python

3.4 Rotire imagine

```
def rotate(angle = 30):  
    angle = float(angle)  
    image = Image.open(name)  
    image.save(name)  
    image = image.rotate(angle)  
    image.save("{}".format(name))
```

3.5 Solarize

```
def solarize(v = 30):  
    v= getInt(v)  
    image = Image.open(name)  
    image = ImageOps.solarize(image, threshold=v) # default  
    image.save(name)
```

3.6 Posterize

```
def posterize(v = 1):  
    v= int(v)  
    image = Image.open(name)  
    image = ImageOps.posterize(image, bits=v)  
    image.save(name)
```

3.7 Invert

```
def invert(v = 30):  
    v= getInt(v)  
    image = Image.open(name)  
    image = ImageOps.invert(image)  
    image.save(name)
```

3.8 Expend

```
def expand(v = 30):  
    v= int(v)  
    image = Image.open(name)  
    image = ImageOps.expand(image, border=v)  
    image.save(name)
```

3.9 Transformarea imaginii in binar

```
def bnw():  
    image = cv2.imread(name)  
    image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)  
    (thresh, image) = cv2.threshold(image, 127, 255, cv2.THRESH_BINARY)  
    cv2.imwrite(name, image)
```

4 Directii viitoare de dezvoltare

Idei viitoare:

1. Adaugarea unui interfete grafice.
2. Adaugarea posibilitatii de prelucrare a unui flux video.
3. Usurarea experientei de utilizare.