

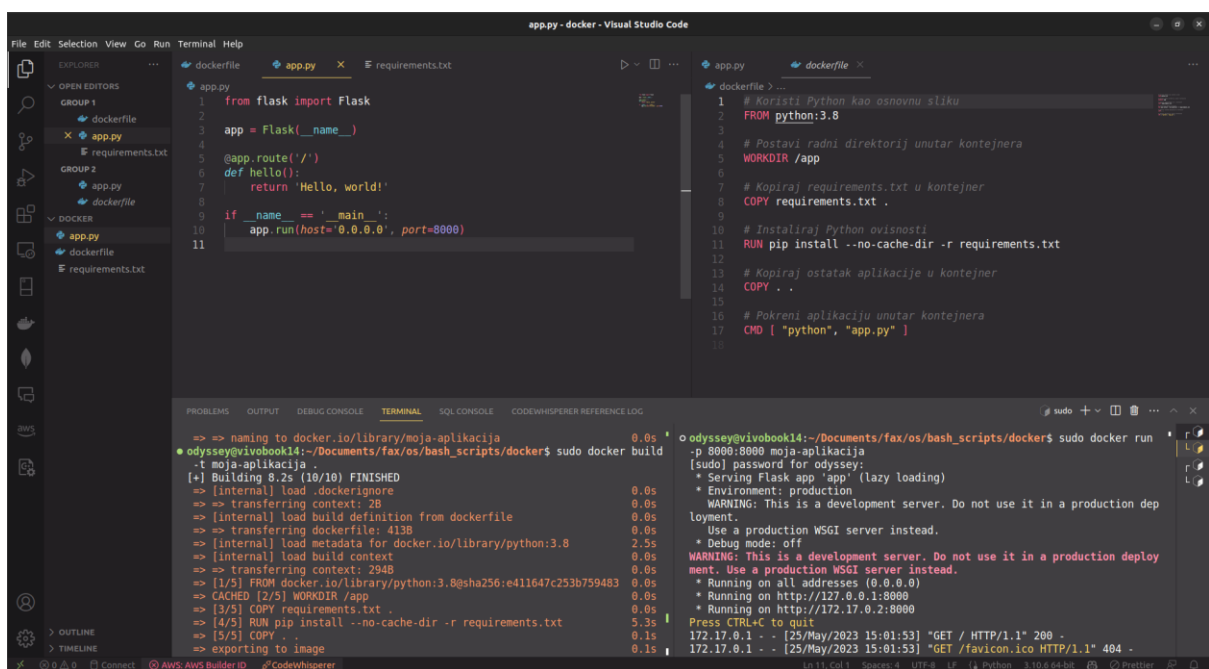
ZADACI ZA IZRADU

Operativni sistemi

Sedmica 13

1 - Zadatak: Kreiranje Dockerfile-a i pokretanje kontejnera

Napravite Dockerfile za vašu aplikaciju koja se temelji na određenoj tehnologiji (npr. Node.js, Python). Izgradite Docker sliku koristeći Dockerfile. Pokrenite kontejner temeljen na izgrađenoj slici i provjerite da li aplikacija ispravno radi unutar kontejnera.



The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** Shows the project structure with files `dockerfile`, `app.py`, `requirements.txt`, and a `DOCKER` folder.
- EDITOR:** Displays the `dockerfile` and `app.py` files.
 - `app.py` contains a simple Flask application with a `hello()` endpoint.
 - `dockerfile` contains the following instructions:

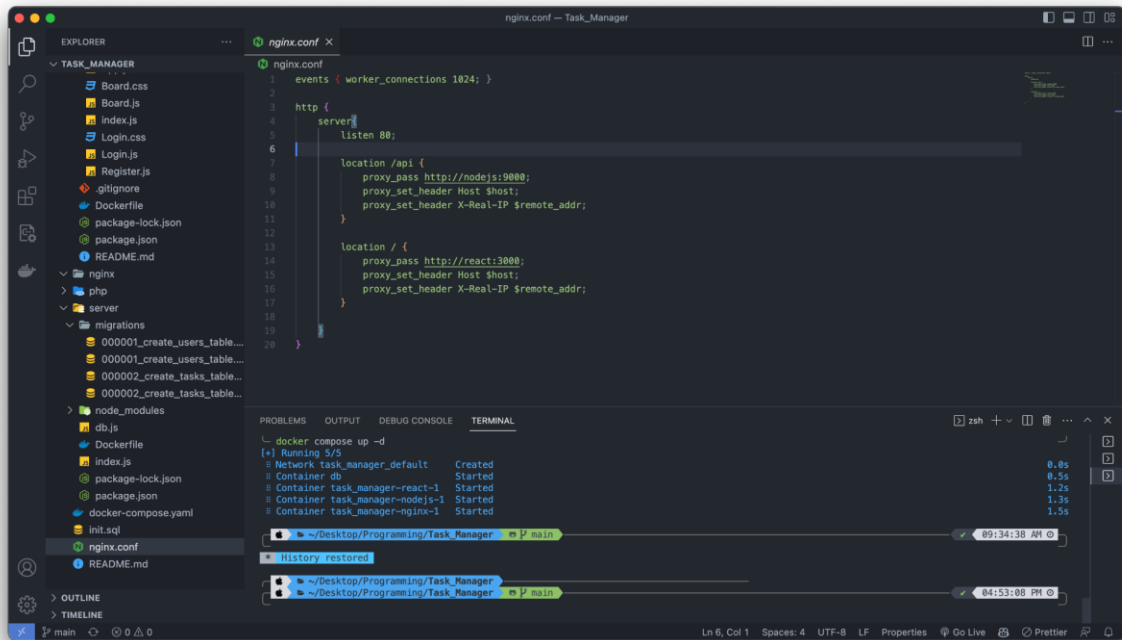
```
1 # Koristi Python kao osnovnu sliku
2 FROM python:3.8
3
4 # Postavi radni direktorij unutar kontejnera
5 WORKDIR /app
6
7 # Kopiraj requirements.txt u kontejner
8 COPY requirements.txt .
9
10 # Instaliraj Python ovisnosti
11 RUN pip install --no-cache-dir -r requirements.txt
12
13 # Kopiraj ostatak aplikacije u kontejner
14 COPY . .
15
16 # Pokreni aplikaciju unutar kontejnera
17 CMD [ "python", "app.py" ]
```
- TERMINAL:** Shows the execution of the following commands:

```
o odysey@vivoobook14:~/Documents/fax/os/bash_scripts/docker$ sudo docker build
-> naming to docker.io/library/moja-aplikacija 0.0s
-> exporting to image 0.1s
[+] Building 8.2s (10/10) FINISHED
=> [internal] load .dockerignore 0.0s
=> transferring context: 2B 0.0s
=> [internal] load build definition from dockerfile 0.0s
=> transferring dockerfile: 413B 0.0s
=> [internal] load metadata for docker.io/library/python:3.8 2.5s
=> [internal] load build context 0.0s
=> transferring context: 294B 0.0s
=> [1/5] FROM docker.io/library/python:3.8@sha256:e411647c253b759483 0.0s
=> CACHED [2/5] WORKDIR /app 0.0s
=> [3/5] COPY requirements.txt . 0.0s
=> [4/5] RUN pip install --no-cache-dir -r requirements.txt 5.3s
=> [5/5] COPY . . 0.1s
```

- Almir Mujanovic

2 - Zadatak: Konfiguriranje Nginx-a kao reverse proxy-ja

Instalirajte Nginx na vašem poslužitelju. Konfigurirajte Nginx da djeluje kao reverse proxy za više web aplikacija. Testirajte pravilno usmjeravanje zahtjeva prema odgovarajućim backend servisima.



-Inas Malkic

3 - Zadatak: Automatizacija deployment procesa s Docker Compose-om i bash skriptiranjem

Napravite Docker Compose datoteku koja definira i pokreće više kontejnera za vašu aplikaciju i njen backend. Napišite bash skriptu koja automatizira postupak izgradnje i pokretanja kontejnera pomoću Docker Compose-a. Pokrenite skriptu i provjerite da li su svi kontejneri ispravno pokrenuti i povezani.

4 - Zadatak: Konfiguriranje Nginx-a za load balancing

Postavite više instanci aplikacije u Docker kontejnerima. Konfigurirajte Nginx da djeluje kao load balancer i ravnomjerno raspoređuje dolazne zahtjeve na sve instance aplikacije. Provjerite da li se zahtjevi pravilno raspoređuju i balansiraju između kontejnera.

```
root@HP-3203-107: /etc/nginx
GNU nano 6.2 nginx.conf
ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
ssl_prefer_server_ciphers on;

##
# Logging Settings
##
access_log /var/log/nginx/access.log;
error_log /var/log/nginx/error.log;

##
# Gzip Settings
##
gzip on;

# gzip_vary on;
# gzip_proxied any;
# gzip_comp_level 6;
# gzip_buffers 16 8k;
# gzip_http_version 1.1;
# gzip_types text/plain text/css application/json application/javascript text/xml application/xml application/xml+rss text/javascript;

##
# Virtual Host Configs
##

include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;

upstream app {
    # IP adrese za load balancing
    server 192.168.1.10;
    server 192.168.1.11;
    server 192.168.1.12;
}

server {
    listen 80;
    server_name app.com;

    location / {
        proxy_pass https://app.com;
    }
}
```

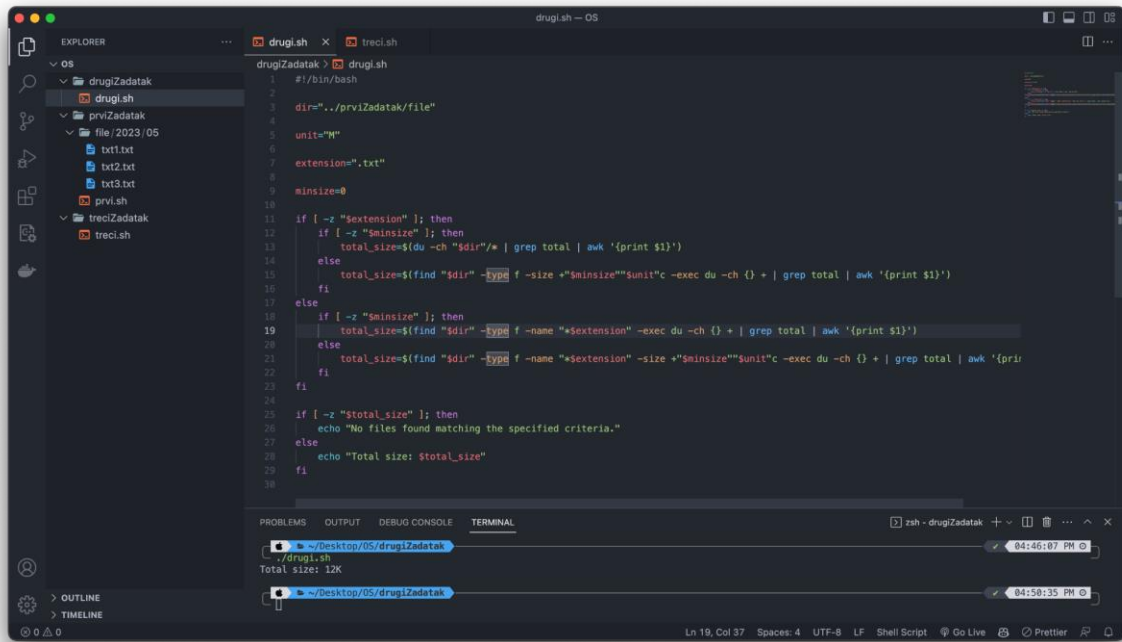
Adi Kadušić, Tarik Glamočak

5 - Zadatak: Automatsko stvaranje rezervne kopije baze podataka

Napišite bash skriptu koja se povezuje na bazu podataka i stvara rezervnu kopiju svih tablica. Koristite alat poput mysqldump ili pg_dump za izradu SQL datoteka rezervnih kopija. Automatizirajte izvršavanje skripte kroz planirani posao (cron job) kako biste redovito izrađivali rezervne kopije.

6 - Zadatak: Praćenje veličine direktorija

Napišite bash skriptu koja rekurzivno pretražuje određeni direktorij i izračunava ukupnu veličinu datoteka. Skripta može ispisati ukupnu veličinu u megabajtima ili gigabajtima. Dodajte opciju za filtriranje po vrsti datoteka (npr. samo .txt datoteke) ili po veličini.



```
1 #!/bin/bash
2
3 dir="../priviZadatak/file"
4
5 unit="M"
6
7 extension=".txt"
8
9 minsize=0
10
11 if [ -z "$extension" ]; then
12     if [ -z "$minsize" ]; then
13         total_size=$(du -ch "$dir"/* | grep total | awk '{print $1}')
14     else
15         total_size=$(find "$dir" -type f -size +"$minsize""$unit"c -exec du -ch {} + | grep total | awk '{print $1}')
16     fi
17 else
18     if [ -z "$minsize" ]; then
19         total_size=$(find "$dir" -type f -name "$extension" -exec du -ch {} + | grep total | awk '{print $1}')
20     else
21         total_size=$(find "$dir" -type f -name "$extension" -size +"$minsize""$unit"c -exec du -ch {} + | grep total | awk '{print $1}')
22     fi
23 fi
24
25 if [ -z "$total_size" ]; then
26     echo "No files found matching the specified criteria."
27 else
28     echo "Total size: $total_size"
29 fi
30
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

zsh - drugiZadatak

~/Desktop/05/drugiZadatak
./drugi.sh
Total size: 12K

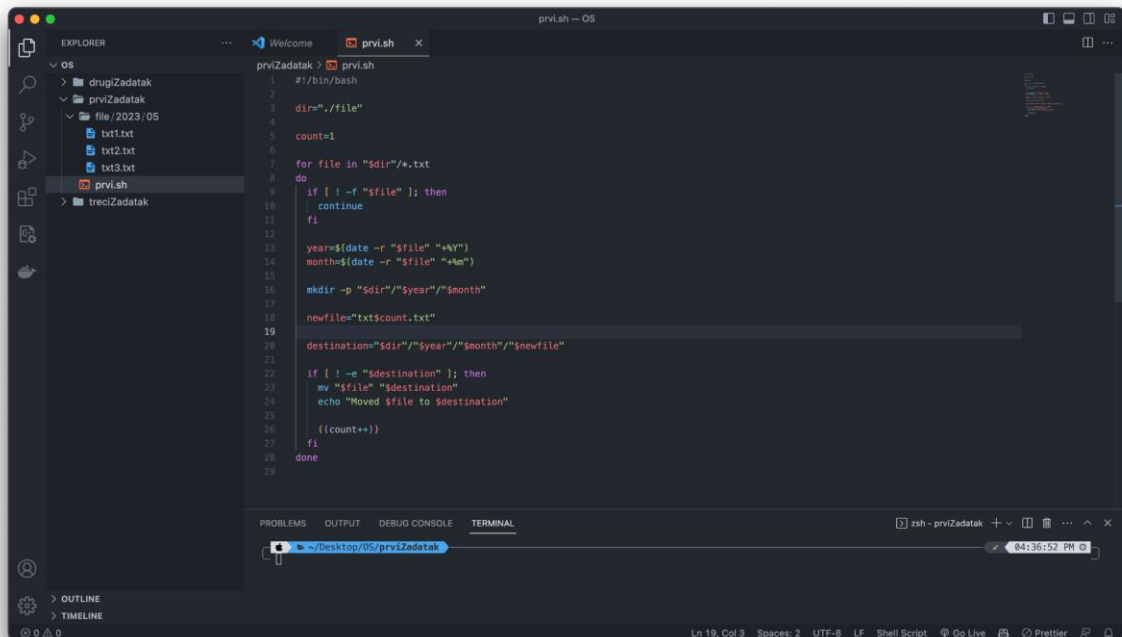
-Inas Malkic

7 - Zadatak: Automatsko preimenovanje i organizacija datoteka

Napišite bash skriptu koja preimenuje datoteke u određenom direktoriju prema određenom uzorku.

Na primjer, može preimenovati sve .jpg datoteke u obliku "slika1.jpg", "slika2.jpg", i tako dalje.

Dodajte logiku za organizaciju datoteka u poddirektorije prema određenim kriterijima, poput datuma ili vrste datoteke.



```
1 #!/bin/bash
2
3 dir="./file"
4
5 count=1
6
7 for file in "$dir"/*.txt
8 do
9     if [ ! -f "$file" ]; then
10         continue
11     fi
12     year=$(date -r "$file" +%Y)
13     month=$(date -r "$file" +%m)
14     mkdir -p "$dir"/"$year"/"$month"
15     newfile="txt$count.txt"
16     destination="$dir"/"$year"/"$month"/"$newfile"
17
18     if [ ! -e "$destination" ]; then
19         mv "$file" "$destination"
20         echo "Moved $file to $destination"
21     fi
22     ((count++))
23 done
24
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

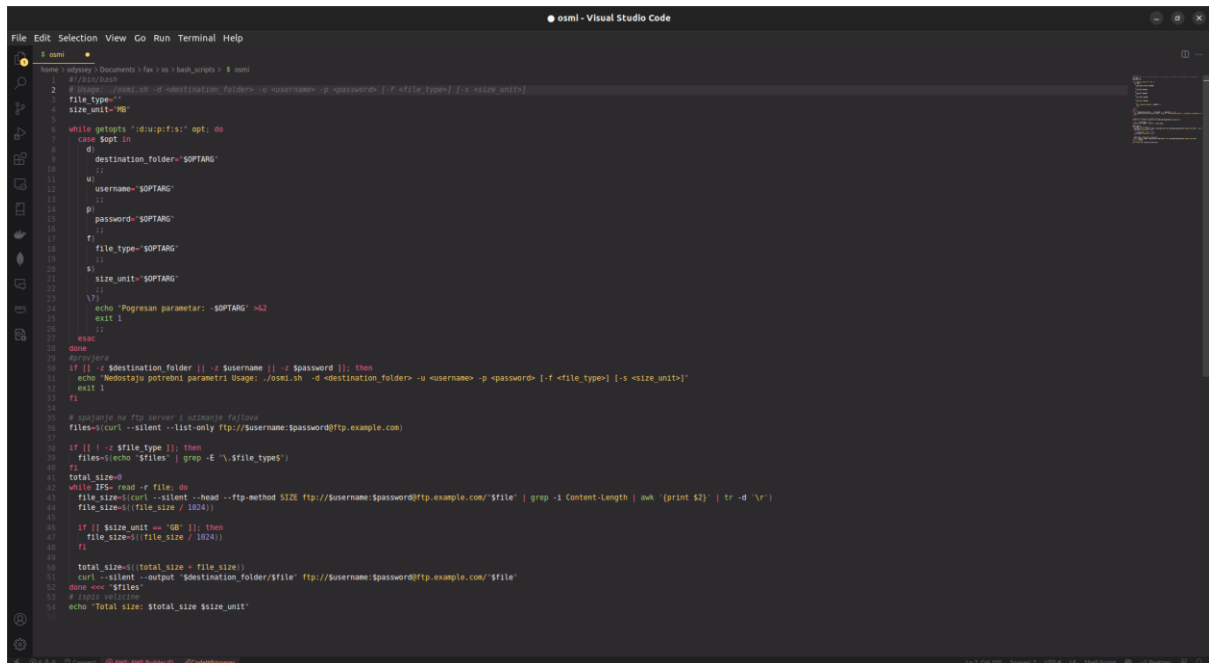
zsh - priviZadatak

~/Desktop/05/priviZadatak

-Inas Malkic

8 - Zadatak: Automatsko preuzimanje datoteka s udaljenog poslužitelja

Napišite bash skriptu koja preuzima datoteke s udaljenog poslužitelja putem FTP-a, SSH-a ili HTTP-a. Skripta može prihvatiti argumente kao parametre za određeni direktorij, korisničko ime, lozinku i slično. Dodajte funkcionalnost za preuzimanje samo određenih datoteka ili preuzimanje samo ako su novije od postojećih lokalnih datoteka.



```
File Edit Selection View Go Run Terminal Help
osmi - Visual Studio Code

name | osmi | Documents | . | bash_scripts | $ nano

1 #!/bin/bash
2 # Preuzimanje datoteka s udaljenog poslužitelja
3 file_type=""
4 size_unit="MB"
5
6 while getopts "d:u:p:f:s:" opt; do
7     case $opt in
8         d)
9             destination_folder="$OPTARG"
10
11         u)
12             username="$OPTARG"
13
14         p)
15             password="$OPTARG"
16
17         f)
18             file_type="$OPTARG"
19
20         s)
21             size_unit="$OPTARG"
22
23         \?)
24             echo "Pogresan parametar: $OPTARG" >62
25             exit 1
26         &
27     esac
28 done
29
30 # Provera parametara
31 if [[ -z $destination_folder || -z $username || -z $password ]]; then
32     echo "Potrebni su parametri: Usage: ./osmi.sh -d <destination_folder> -u <username> -p <password> [-f <file_type>] [-s <size_unit>]"
33     exit 1
34 fi
35
36 # Preuzimanje sa ftp servera (primjerice)
37 files=$(curl --silent --list-only ftp://$username:$password@ftp.example.com)
38
39 if [[ -z $file_type ]]; then
40     files=$(echo "$files" | grep -E "\.${file_type}$")
41 fi
42
43 total_size=0
44 while IFS= read -r file; do
45     file_size=$(curl --silent --head --ftp-method SIZE ftp://$username:$password@ftp.example.com/$file | grep -i Content-Length | awk '{print $2}' | tr -d '\r')
46     file_size=$((file_size / 1024))
47
48     if [[ $size_unit == "KB" ]]; then
49         file_size=$((file_size / 1024))
50     fi
51
52     total_size=$((total_size + file_size))
53
54     curl --silent --output "$destination_folder/$file" ftp://$username:$password@ftp.example.com/$file
55 done << "$files"
56 # Ispis rezultata
57 echo "Total size: $total_size $size_unit"
```

- Almir Mujanovic

9 - Zadatak: Provjera dostupnosti web stranica

Napišite bash skriptu koja provjerava dostupnost određenog popisa web stranica. Skripta može koristiti alat poput curl ili wget za slanje HTTP zahtjeva i provjeru statusnog koda. Izvješće o dostupnosti može biti zapisano u log datoteku ili poslano putem e-pošte.

```
GNU nano 6.2 deveti *
#!/bin/bash

web_stranice=(
    "http://youtube.com"
    "http://google.com"
    "http://facebook.com"
)

log_file="dostupnost.log"

for website in "${web_stranice[@]}; do
    status_code=$(curl -o /dev/null --silent --head --write-out '%{http_code}' "$website")
    timestamp=$(date +"%Y-%m-%d %T")

    if [ "$status_code" -eq 200 ]; then
        echo "[$timestamp] $website DOSTUPAN (Status code: $status_code)" >> "$log_file"
    else
        echo "[$timestamp] $website NEDOSTUPAN (Status code: $status_code)" >> "$log_file"
    fi
done
```

- Almir Mujanovic

```
root@HP-3203-107: ~
GNU nano 6.2 script.sh
websites=(
    "https://www.google.com"
    "https://www.github.com"
)

for website in "${websites[@]}; do
    if curl --head --silent --fail "$website" >/dev/null; then
        echo "Stranica $website je dostupna"
    else
        echo "Stranica $website nije dostupna"
    fi
done
```

Adi Kadušić, Tarik Glamočak

5 - Rješenje: Automatsko stvaranje rezervne kopije baze podataka

bash

Copy code

```
#!/bin/bash
```

```
DB_HOST="localhost"
```

```
DB_USER="username"
```

```
DB_PASSWORD="password"
```

```
DB_NAME="database_name"
```

```
BACKUP_DIR="/path/to/backup/directory"
```

```
DATE=$(date +%Y%m%d%H%M%S)
```

```
BACKUP_FILE="$BACKUP_DIR/db_backup_$(DATE).sql"
```

```
# Stvaranje rezervne kopije baze podataka
```

```
mysqldump -h $DB_HOST -u $DB_USER -p$DB_PASSWORD $DB_NAME > $BACKUP_FILE
```

```
# Provjera rezultata izvođenja naredbe
```

```
if [ $? -eq 0 ]; then
```

```
    echo "Rezervna kopija baze podataka je uspješno stvorena: $BACKUP_FILE"
```

```
else
```

```
    echo "Greška prilikom stvaranja rezervne kopije baze podataka."
```

```
fi
```

6 - Rješenje: Praćenje veličine direktorija

bash

Copy code

```
#!/bin/bash
```

```
DIRECTORY="/path/to/directory"
```

```
total_size=0
```

```
# Rekurzivno pretraživanje direktorija i izračunavanje veličine datoteka
```

```
calculate_directory_size() {
```

```
    for file in "$1"/*; do
```

```
        if [ -f "$file" ]; then
```

```
            size=$(du -b "$file" | cut -f 1)
```

```
            total_size=$((total_size + size))
```

```
        elif [ -d "$file" ]; then
```

```
            calculate_directory_size "$file"
```

```
        fi
```

```
    done
```

```
}
```

```
calculate_directory_size "$DIRECTORY"
```

```
# Ispis ukupne veličine
```

```
echo "Ukupna veličina direktorija $DIRECTORY: $total_size bajtova"
```

7 - Rješenje: Automatsko preimenovanje i organizacija datoteka

```
bash
```

Copy code


```
#!/bin/bash
```

```
DIRECTORY="/path/to/directory"
```

```
count=1
```

```
# Preimenovanje datoteka
```

```
rename_files() {
```

```
    for file in "$DIRECTORY"/*; do
```

```
        if [ -f "$file" ]; then
```

```
            extension="{file##*}."
```

```
            new_name="slika$count.$extension"
```

```
            mv "$file" "$DIRECTORY/$new_name"
```

```
            count=$((count + 1))
```

```
        fi
```

```
    done
```

```
}
```

```
rename_files
```

```
# Organizacija datoteka u poddirektorije
```

```
organize_files() {
```

```
    for file in "$DIRECTORY"/*; do
```

```
        if [ -f "$file" ]; then
```

```
            year=$(date -r "$file" +%Y)
```

```
            month=$(date -r "$file" +%m)
```

```
            destination="$DIRECTORY/$year/$month"
```

```
            mkdir -p "$destination"
```

```
            mv "$file" "$destination"
```

```
        fi
```

```
    done
```

```
}
```

```
organize_files
```