

### 3. domaća zadaća: TCP poslužitelj i klijent

#### Git i Make

U svoj Git repozitorij dodati poddirektorij `zadatak3` u kojem se treba nalaziti odgovarajuća `[Mm]akefile` datoteka s definiranim sljedećim ciljevima (pravilima):

- `default`ni cilj je kreiranje izvršnih datoteka `tcpserver` i `tcpklijent`,
- `make clean` - obriše izvršne programe i sve ostale nepotrebne datoteke.

Nakon poziva `make` kao rezultat se moraju dobiti izvršni programi `tcpserver` i `tcpklijent`.

#### Program `tcpserver`

`tcpserver [-p port]`

`-p port` naziv servisa ili broj porta (*default* je 1234)

Poslužiteljski program `tcpserver` čeka dolazne konekcije na definiranom TCP *portu*. Pret-postavljeni broj *porta* je 1234. Zahtjev od klijenta je oblika:

- prva 4 okteta su pomak (*offset*) od početka tražene datoteke u „*network byte order*”,
- zatim slijedi naziv datoteke koji završava sa znakom `\0` („*null-terminated string*”).

Kao odgovor, `tcpserver` vraća klijentu 1 oktet koji predstavlja status:

- `0x00` i sadržaj tražene datoteke počevši od pomaka *offset*,
- `0x01` i proizvoljnu poruku; ako tražena datoteka ne postoji,
- `0x02` i proizvoljnu poruku; ako nema prava za čitanje datoteke,
- `0x03` i proizvoljnu poruku; za ostale greške.

Program `tcpserver` nakon toga zatvara konekciju s klijentom i čeka sljedeću konekciju.

`tcpserver` zahtjeve poslužuje slijedno; ne treba raditi konkurentni poslužitelj, to će biti sljedeća vježba.

`tcpserver` smije dohvaćati samo datoteke iz trenutnog direktorija iz kojeg je pozvan poslužitelj. Ako klijent traži nepostojeću datoteku ili datoteke iz nekog drugog direktorija (tj. u nazivu tražene datoteke se nalazi znak `/`) poslužitelj vraća klijentu kod `0x01` (kao da datoteka ne postoji) i poruku o grešci te zatvara TCP konekciju s klijentom. Ako klijent traži datoteku za koju poslužitelj nema prava za čitanje, poslužitelj vraća klijentu kod `0x02` i poruku o grešci te zatvara TCP konekciju s klijentom.

#### Program `tcpklijent`

`tcpklijent [-s server] [-p port] [-c] filename`

`-s server` naziv ili IP adresa servera (*default* je 127.0.0.1)

`-p port` naziv servisa ili broj porta (*default* je 1234)

`-c` nastavlja se dohvaćanje djelomično prenesene datoteke; pošalje serveru *offset* s veličinom lokalne datoteke

`filename` naziv datoteke koja se traži

Dohvaćeni podaci spremaju se u `filename`, a ako se koristi opcija `-c` dodaju se na kraj datoteke `filename`.

Ako se program pozove bez opcije `-c`, a u direktoriju već postoji datoteka `filename`, program ispisuje poruku korisniku na standardni izlaz za greške i završava s radom uz status različit od 0.

Ako se program pozove s opcijom `-c`, a korisnik nema dozvolu pisanja u datoteku `filename`, program ispisuje poruku na `stderr` i završava s radom uz status različit od 0.

Ako se program pozove s opcijom `-c`, a datoteka `filename` ne postoji u trenutnom direktoriju, prema poslužitelju se šalje zahtjev s *offsetom* 0.

Kad primi traženu datoteku, odnosno kad poslužitelj raskine konekciju s klijentom, program `tcpklijent` završava s radom uz status jednak prvom primljenom oktetu od poslužitelja (0 - kad poslužitelj šalje traženu datoteku, 1 kad tražena datoteka ne postoji, 2 kad nema prava za čitanje, te 3 ako je došlo do neke greške).

### Primjeri pozivanja:

Pokrenemo poslužiteljski program `tcpserver` na *portu* 4444 računala s IP adresom 10.0.1.2:

```
$ ./tcpserver -p 4444 &
```

Klijentski program traži od poslužitelja datoteku `mala`:

```
$ ./tcpklijent -p 4444 -s 10.0.1.2 mala
```

Za provjeru ispravnog prijenosa većih, binarnih datoteka prenosimo datoteku `/usr/bin/dockerd`. Kako poslužitelj dozvoljava prijenos samo onih datoteka koje se nalaze u trenutnom direktoriju, program `tcpserver` pozivamo iz direktorija `/usr/bin`:

```
$ cd /usr/bin
```

```
$ /home/student/zadatak3/tcpserver -p 1234 &
```

Klijentski program pozivamo iz HOME direktorija:

```
$ cd
```

```
$ ./tcpklijent -p 1234 dockerd
```

Programom `cmp` usporedimo sadržaj te dvije datoteke:

```
$ cmp /usr/bin/dockerd dockerd
```

```
$ echo $?
```

```
0
```

Programom `cmp` provjeravamo ispravnost primljenih podataka. Ako se datoteke razlikuju, program `cmp` će ispisati poruku i postaviti izlazni status 1, a ako nema razlika neće se ispisati nikakva poruka i izlazni status će biti 0. (Tekstualne datoteke možemo uspoređivati i programom `diff`).

Provjeru ispravnog prijenosa djelomično prenesene datoteke možemo obaviti tako da u editoru „obrišemo” dio (tekstualne) datoteke ili kopiramo dio već prenesene datoteke korištenjem programa `dd`.

Na primjer, kopiranje prvih 100000 okteta iz datoteke `dockerd` u datoteku `dio` (u HOME direktoriju):

```
$ dd if=/usr/bin/dockerd of=dio bs=1 count=100000
```

```
$ mv dio dockerd
```

te zatim pozovemo `tcpkljent` s opcijom `-c` i na kraju usporedimo sadržaj datoteka:

```
$ ./tcpkljent -p 1234 -c dockerd
$ cmp /usr/bin/dockerd dockerd
$ echo $?
0
```

### Poruke o greškama

Ako se programi pozovu s nedefiniranim opcijama ili ako se `tcpkljent` pozove bez naziva tražene datoteke, na standardni izlaz za greške (*stderr*) treba ispisati poruku:

```
Usage: ./tcpkljent [-s server] [-p port] [-c] filename
```

odnosno:

```
Usage: ./tcpserver [-p port]
```

te završiti s radom uz izlazni status različit od 0.

Programi moraju provjeravati povratne vrijednosti funkcija i u slučaju problema ispisati poruku na standardni izlaz za greške (*stderr*) te završiti s radom uz izlazni status različit od 0:

```
./tcpserver: opis greske (na primjer korištenjem err)
```