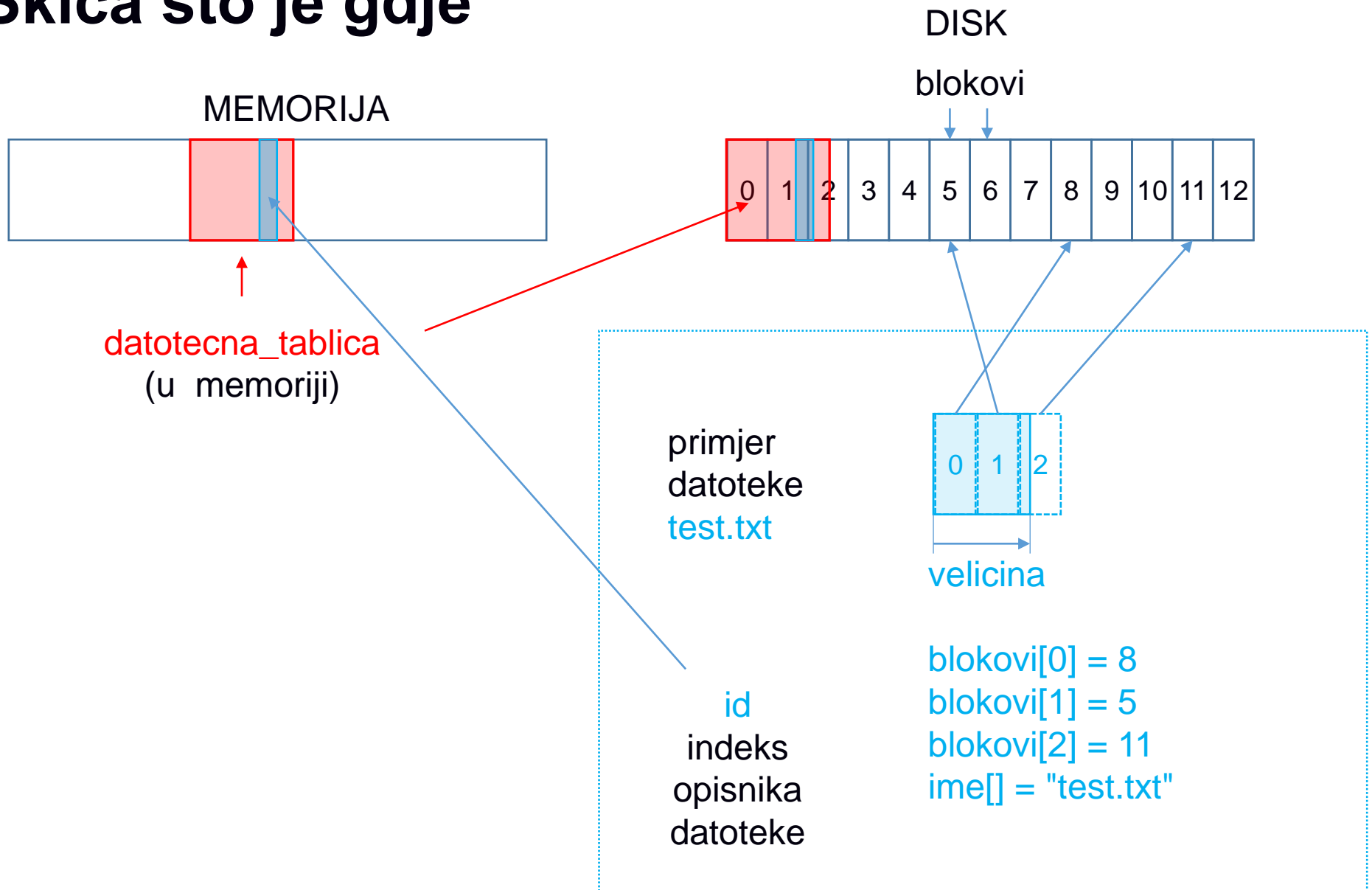


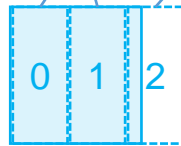
OS-PMF

dodatek za LAB4

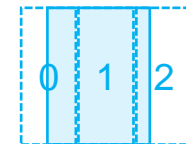
Skica što je gdje



Kopiranje iz datoteke



početno stanje



dat.veličina

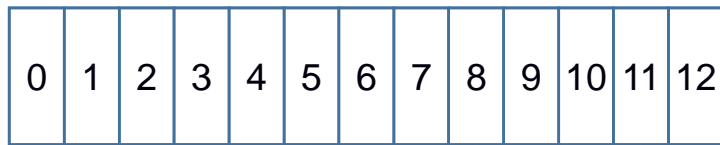
dat.kazaljka

još

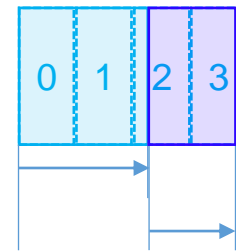
čitaj datoteku (id, p, koliko)

```
{
    buffer[VEL_BLOKA]
    dat = opisnik datoteke na mjestu id
    još = koliko
    blok_dat = dat.kazaljka / VEL_BLOKA
    dok je (još > 0 //još se može čitati
        && dat.kazaljka < dat.veličina //ima još u datoteci
        && dat.blokovi[blok_dat] >= 0) //postoji taj blok datoteke
    {
        dohvati blok diska dat.blokovi[blok_dat] u buffer
        kopirati = min(još, dat.veličina - dat.kazaljka,
            VEL_BLOKA - dat.kazaljka % VEL_BLOKA)
        kopiraj kopirati bajtova iz buffer + dat.kazaljka % VEL_BLOKA u p
        p += kopirati
        dat.kazaljka += kopirati
        još -= kopirati
        blok_dat++
    }
    vrati koliko - još
}
```

Kopiranje u datoteku



početno stanje



dat.veličina

još

piši u datoteku (id, p, koliko)

```
{
    buffer[VEL_BLOKA]
    dat = opisnik datoteke na mjestu id
    još = koliko
    blok_dat = dat.kazaljka / VEL_BLOKA
    max_vel_datoteke = VEL_BLOKA * BR_BLOKOVA
    dok je (još > 0 && dat.kazaljka < max_vel_datoteke)
    {
        ako je dat.blokovi[blok_dat] == -1 {
            nađi slobodni blok i ažuriraj dat.blokovi
            ako nema stani i izađi iz petlje
        }
        inače {
            dohvati blok diska dat.blokovi[blok_dat] u buffer
        }
        kopirati = min(još, VEL_BLOKA - dat.kazaljka % VEL_BLOKA)
        kopiraj kopirati bajtova iz p u buffer + dat.kazaljka % VEL_BLOKA
        zapiši blok na disk (buffer => dat.blokovi[blok_dat])
        p += kopirati
        dat.kazaljka += kopirati
        još -= kopirati
        blok_dat++
    }
    dat.veličina = max(dat.kazaljka, prethodna veličina datoteke)
    //ili dat.veličina = dat.kazaljka
    //te osloboditi blokove od 'blok_dat' na dalje
    vrati koliko - još
}
```