Operativni sistemi

dr.sc. Amer Hasanović

Fajl sistemi

- Fajl sistemi imaju zadatak da:
  - organiziraju i pohranjuju podatke;
    - količina podataka za pohranu može biti veća od virtuelnog adresnog prostora.
  - omogućavaju dijeljenje podataka između aplikacija i korisnika;
    - podaci moraju biti dostupni i nakon terminacije procesa unutar kojeg su generirani;
    - proizvoljan broj procesa treba da ima istovremeni pristup podacima;
  - čuvaju podatke nakon gašenja sistema.
    - podaci u fajl sistemu trebaju uvijek biti u konsistentom stanju, čak i u slučaju iznenadnog nestanka napajanja.

Mediji za trajnu pohranu podataka

- Diskovi:
  - magnetni (hard disk);
  - SSD (solid state disk).
- Organiziraju podatke u lineranoj sekvenci blokova bita, tzv sektori, koji su fiksne dužine (npr 512B).
- Podržavaju dvije ključne operacije:
  - pročitaj *k*-ti sektor, transfer se vrši sa diska u RAM;
  - snimi *k*-ti sektor, transfer se vrši iz RAM-a na disk.

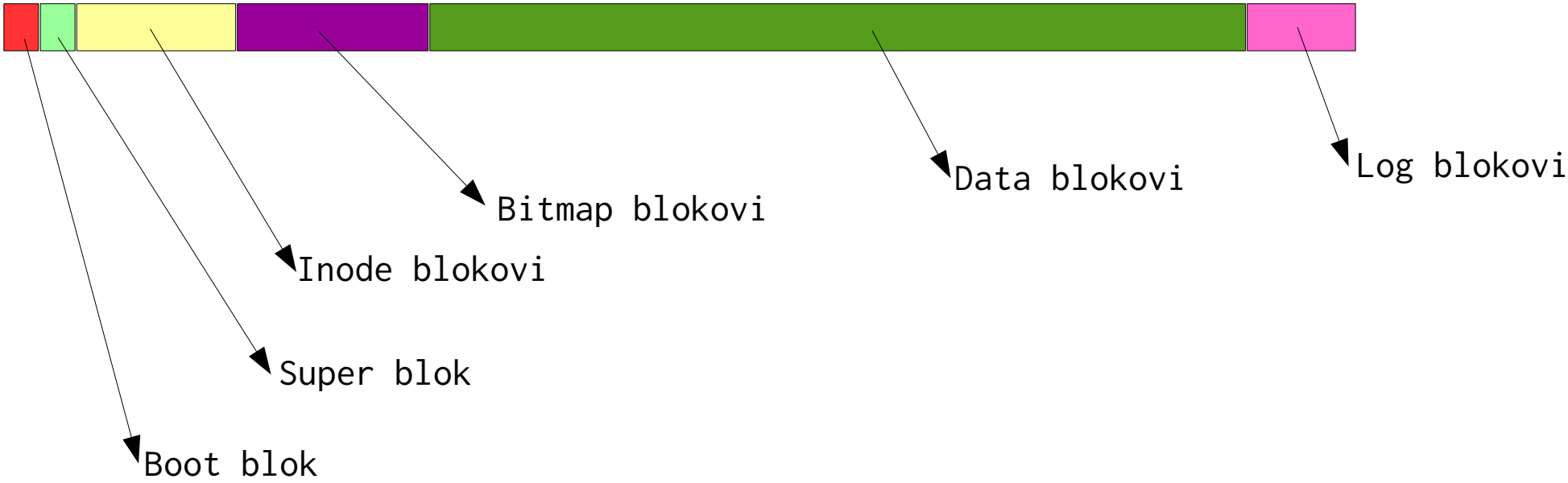
- Neka pitanja vezana za trajnu pohranu podataka:
  - Kako pronaći potrebni podatak na mediju?
  - Kako onemogućiti korisnika da mijenja tuđe podatke?
  - Kako pronaći slobodne blokove na mediju? itd..
- Podaci se na mediju organizuju u fajl sistem:
  - osnovna jedinica za pohranu podataka u sistemu zove se **fajl**;
  - radi lakšeg pristupa za korisnike, fajlovi se organizuju u **direktorije**, koji mogu sadržavati:
    - fajlove;
    - druge direktorije, čime se formira struktura stabla.
  - fajlovi i direktoriji se imenuju, krajnji korisnik koristi ova imena da bi pomoću posebnih sistemskih poziva došao do podataka pohranjenih u fajlovima.

Fajlovi

- Fajlovi su logičke jedinice podataka koje sadrže informacije proizvedene unutar procesa:
  - za krajnjeg korisnika apstrahuju komunikaciju sa medijima za pohranu podataka;
  - organizirani i upravljeni od strane OS-a.
    - OS tretira fajl kao nestruktuiranu kontinualnu sekvencu bajta, koja se može mijenjati, proširivati, brisati i trajno pohranjivati na disku.
    - OS mora voditi računa u kojim sektorima se nalaze podaci koje pripadaju određenom fajlu, kao i o slobodnim sektorima na mediju.
- `xv6` implementira fajl sistem u sedam slojeva funkcija i struktura
  - Svaki sloj ima specifične zadatke i namjenu.

xv6 Fajl Sistem (FS) organizacija

xv6 fajl sistem sekcije



Superblok

- informacije o počecima i krajevima ostalih sekcija FS-a i ukupnoj veličini FS-a.

Inode blokovi

- Niz inode struktura:
  - jedan inode predstavlja fajl ili direktorij u FS-u;
  - bilježi veličinu fajla i sektore u kojima su pohranjeni podaci iz fajla;

Bitmap blokovi

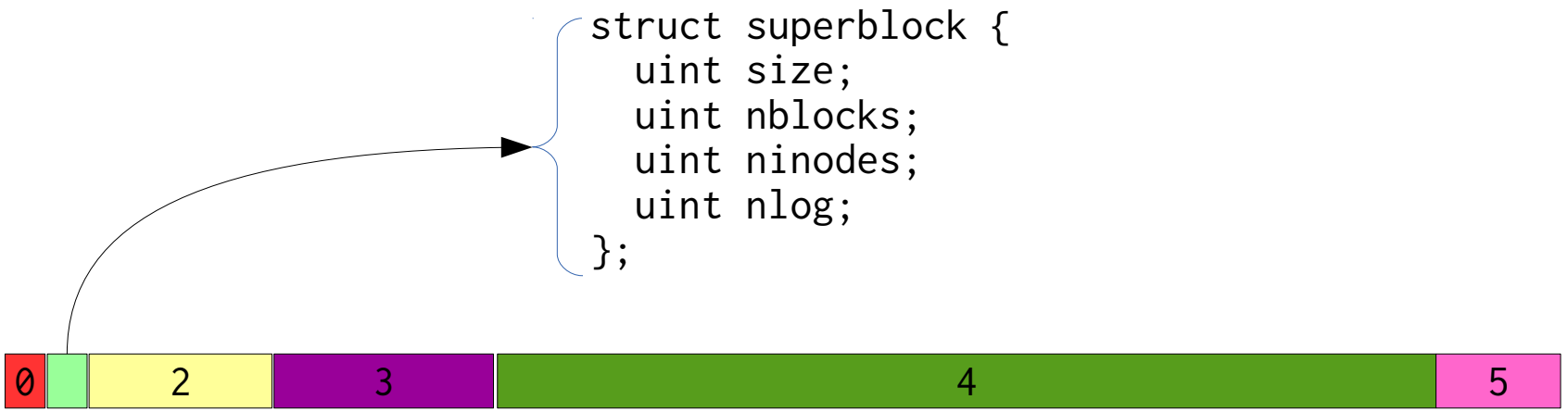
- Informacije o zauzetim i slobodnim sektorima;
  - po jedan bit za jedan sektor

Data blokovi

- sektori sa podacima koji pripadaju različitim fajlovima.

Log blokovi

- sektori sa podacima koje pripadaju jednoj transakciji
  - pisanja u FS grupišu se u transakcije, radi:
    - rekonstrukcije konsistentnog FS u slučaju iznenadnog gubitka napajanja;
    - optimizacije pisanja na medij.



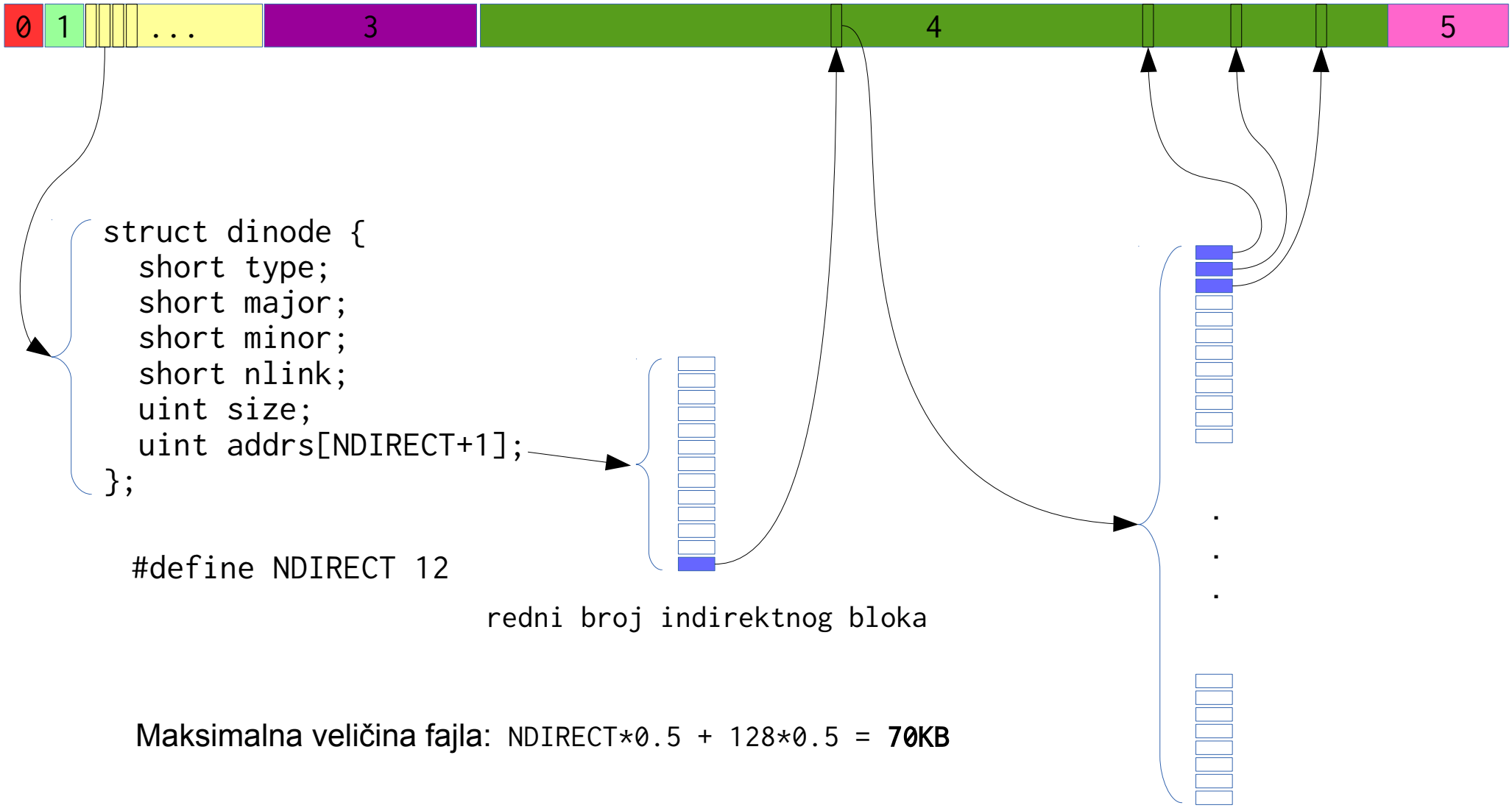
- broj sektora u sekciji 2:
  - `(ninodes * sizeof(struct dinode)) / 512`
- broj sektora u sekciji 3:
  - `nblocks / 512`
- broj sektora u sekciji 4:
  - `nblocks`
- broj sektora u sekciji 5:
  - `nlog`
- ukupno sektora:
  - `size`


```
struct dinode {
  short type;
  short major;
  short minor;
  short nlink;
  uint size;
  uint addrs[NDIRECT+1];
};
```

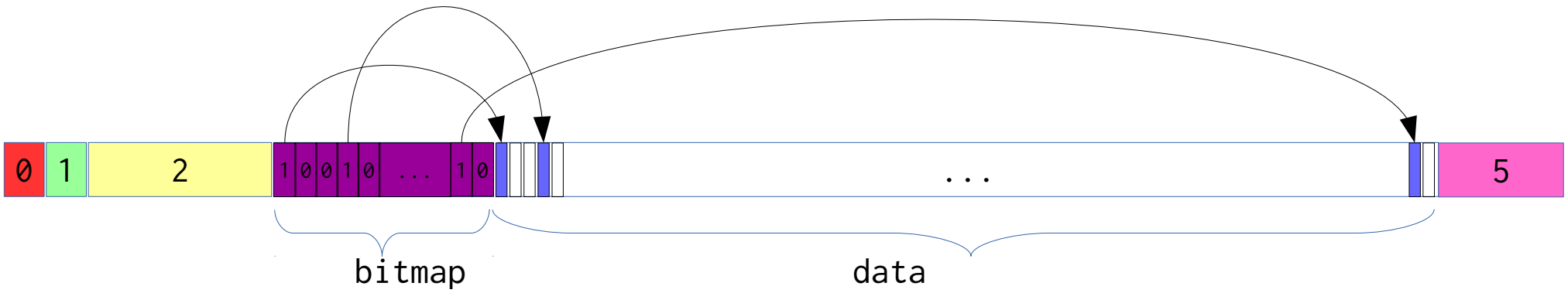
Redni brojevi blokova koje koristi fajl asociran sa inode-om

```
#define NDIRECT 12
```

```
#define T_DIR  1   // Directory
#define T_FILE 2   // File
#define T_DEV  3   // Device
```



Maksimalna veličina fajla: `NDIRECT*0.5 + 128*0.5 =` **`70KB`**


0
1
2
1 0 0 1 0 ... 1 0
...
5
bitmap
data

Direktoriji

- Formiraju infrastrukturu za imenovanje fajlova
  - Asociraju korisnički definirana imena fajlova ili direktorija sa inode-ima u FS-u
  - Direktorij je i sam inode tipa `T_DIR`, čiji je sadržaj sekvenca parova (*hard links*):
    - `(korisničko ime, broj inode-a)`
  - Korisničko ime za inode mora biti unikatno u datom direktoriju:

```
#define DIRSIZ 14

struct dirent {
  ushort inum;
  char name[DIRSIZ];
};
```

- Direktoriji organizuju FS u strukturu stabla:
  - Na vrhu stabla je tzv root inode, rednog broja `1` i imena “`/`”
- Stablo se u potrazi za određenim inode-om, može referencirati apsolutnom stazom spram root inode-a, npr:
    - `/home/amer/test.txt`
  - ili, relativnom stazom spram inodea asociranim sa poljem `cwd` strukture `proc` procesa iz kojeg se vrši referenciranje, npr:
    - `amer/test.txt`
      - Pretpostavka da `cwd` polje trenutnog procesa pokazuje na inode asociran sa imenom “`/home`”

- Bilo koji direktorij sadrži minimalno dva para:
  - (“.”, inode broj od tog direktorija);
  - (“..”, inode broj od direktorija koji sadrži taj direktorij).
- Jedan fajl može biti asociran u fajl sistemu pod istim ili različitim imenom, ali u različitim direktorijima FS-a
  - polje `nlink` inodea datog fajla predstavlja broj hard link-ova u FS sa datim fajlom.