

Svojstva datoteka

UDŽBENIK, POGLAVLJE 8

Svojstva datoteka

- Svaka datoteka poseduje ime koje bira korisnik. Poželjno je da ime datoteke ukazuje na:
 - njen konkretan sadržaj
 - na vrstu njenog sadržaja (radi klasifikacije datoteka po njihovom sadržaju)
- Zato su imena datoteka dvodelna npr:
 - **godina1.txt**
- Može da predstavlja ime datoteke, koja sadrži podatke o studentima prve godine studija. Na to ukazuje prvi deo imena **godina1**, dok drugi deo imena **txt** ove datoteke govori da je datoteka tekstualna, odnosno da sadrži samo vidljive ASCII znakove.

Svojstva datoteka

•Rukovanje datotekom obuhvata ne samo rukovanje njenim sadržajem, nego i rukovanje njenim **imenom**. Izmena imena datoteke moguća je kod:

-Stvaranja datoteke

-Editovanja

-Kompilacije

-Kopiranja

Organizacija datoteka

- Datoteke se grupišu u **skupove** datoteka (datoteke sa podacima o studentima pojedinih godina studija istog odseka pripadaju istom skupu).
- Svaki skup datoteka predstavlja **imenik (directory, folder)** koji sadrži imena svih datoteka koje pripadaju datom skupu.
- Radi razlikovanja imenika, svaki od njih poseduje ime koje bira korisnik. Na primer:
–**odsek**
- može da predstavlja ime imenika, koji obuhvata datoteke sa podacima o studentima svih godina studija istog odseka.

Organizacija datoteka

- Razvrstavanjem datoteka uz pomoć imenika nastaje **hijerarhijska organizacija datoteka**, u kojoj su na višem nivou hijerarhije **imenici**, a na nižem nivou se nalaze **datoteke**, čija imena su sadržana u ovim imenicima.
- Hijerarhijsku oznaku ili putanju (**path name**) datoteke obrazuju ime imenika za koji je datoteka vezana i ime datoteke. Delove putanje obično razdvaja znak / (ili znak \):
-**odsek1/godina1.txt**
- predstavlja putanju datoteke, koja sadrži podatke o studentima prve godine studija sa prvog odseka.
- Uobičajeno je da se ime imenika završava znakom /

Organizacija datoteka

- Hijerarhijska organizacija datoteka ima više nivoa, kada jedan imenik sadrži, pored **imena datoteka**, i imena **drugih imenika**, odnosno obuhvata, pored **datoteka**, i **druge imenike**.
- Obuhvaćeni imenici se nalaze na nižem nivou hijerarhije. Na primer, imenik fakultet obuhvata imenike pojedinih odseka.
- Na vrhu hijerarhijske organizacije datoteka se nalazi **korenski** imenik (**root**) koji obično nema ime.

Organizacija datoteka

- U slučaju više nivoa u hijerarhijskoj organizaciji datoteka, putanju datoteke obrazuju **imena imenika** sa svih nivoa hijerarhije (navedena u redosledu od **najvišeg nivoa na dole**) kao i ime datoteke.

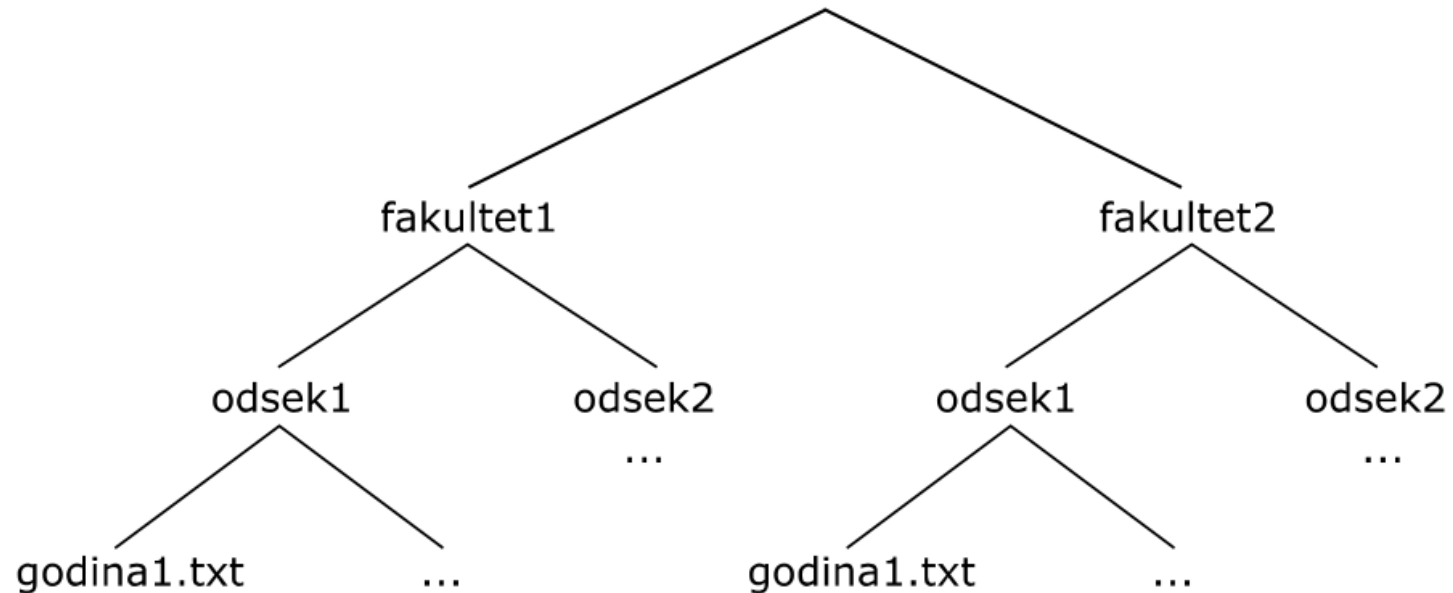
- Na primer:

–/fakultet/odsek1/godina1.txt

- predstavlja putanju datoteke **godina1.txt**, koja pripada imeniku **odsek1**. Ovaj imenik pripada imeniku **fakultet**, a on pripada **korenskom** imeniku **/**.

Organizacija datoteka

- Hijerarhijska organizacija datoteka dozvoljava da postoje **datoteke (imenici)** sa **istim imenima**, pod uslovom da pripadaju **raznim imenicima**.



Organizacija datoteka

- U hijerarhijskoj organizaciji datoteka korenskom imeniku pripadaju imenici **fakultet1** i **fakultet2**.
- Svaki od njih sadrži imenike **odsek1** i **odsek2**. Pri tome, oba imenika sa imenom **odsek1** sadrže datoteku **godina1.txt**.
- Putanje omogućuju razlikovanje istoimenih imenika, odnosno istoimenih datoteka. Tako, putanje:
–/fakultet1/odsek1/ i **/fakultet2/odsek1/**
- omogućuju razlikovanje imenika sa imenom **–odsek1**,
- a putanje:
–/fakultet1/odsek1/godina1.txt
–/fakultet2/odsek1/godina1.txt
- omogućuju razlikovanje datoteka sa imenom **godina1.txt**

Organizacija datoteka

- Zahvaljujući hijerarhijskoj organizaciji datoteka, moguće je rukovanje **skupovima** datoteka.
- Na primer, moguće je kopiranje **celog imenika**, odnosno kopiranje **svih datoteka i imenika, koji mu pripadaju**.
- Navođenje **apsolutne putanje datoteke, sa svim prethodećim imenicima**, je potrebno kad god je moguć **nesporazum**, zbog **datoteka sa istim imenima**, odnosno, zbog **imenika sa istim imenima**.
- Ali, ako postoji mogućnost određivanja nekog imenika kao **radnog (working directory)**, tada se njegova putanja može **podrazumevati** i ne mora se navoditi.

Organizacija datoteka

- Na primer, ako se podrazumeva da je:

- fakultet1/odsek1/**

- radni imenik, tada:

- godina1.txt**

- jednoznačno označava datoteku, koja pripada imeniku **odsek1** iz imenika **fakultet1**.

- Radni imenik omogućuje korišćenje relativnih putanja. Na primer, ako se podrazumeva da je:

- fakultet1/**

- radni imenik, tada:

- odsek1/godina1.txt**

- jednoznačno označava **datoteku**, koja pripada imeniku **fakultet1**.

- **Datoteke koje pripadaju istoj hijerarhijskoj organizaciji obrazuju sistem datoteka.**

Zaštita datoteka

- Za uspešnu upotrebu podataka, trajno pohranjenih u datotekama, neophodna je **zaštita datoteka**.
- Ona obezbeđuje da podaci, sadržani u datoteci, neće biti **izmenjeni** bez znanja i saglasnosti njihovog vlasnika (zabrana prava **pisanja**).
- Takođe, ona obezbeđuje da podatke, sadržane u datoteci jednog korisnika, bez njegove dozvole drugi korisnici **ne mogu da koriste** (zabrana prava **čitanja**).

Zaštita datoteka

- Na ovaj način uvedeno pravo pisanja i pravo čitanja datoteke omogućuju da se za svakog korisnika jednostavno ustanovi koja vrsta rukovanja datotekom mu je **dozvoljena**, a koja **ne**.
- Tako, korisniku, koji **ne poseduje pravo pisanja** datoteke, nisu dozvoljena rukovanja datotekom, koja izazivaju **izmenu** njenog sadržaja.
- Ili, korisniku, koji **ne poseduje pravo čitanja** datoteke, nisu dozvoljena rukovanja datotekom, koja zahtevaju **preuzimanje** njenog sadržaja.

Zaštita datoteka

- Za izvršne datoteke uskraćivanje prava **čitanja** je **prestrogo**, jer sprečava ne samo neovlašćeno uzimanje tuđeg izvršnog programa, nego i njegovo **izvršavanje (execution)**.
- Zato je uputno, radi izvršnih datoteka, uvesti posebno **pravo izvršavanja** programa, sadržanih u izvršnim datotekama.
- Zahvaljujući posedovanju ovog prava, korisnik može da pokrene **izvršavanje** programa, sadržanog u izvršnoj datoteci, **i onda kada nema pravo njenog čitanja**.

Zaštita datoteka

- Pravo **čitanja**, pravo **pisanja** i pravo **izvršavanja** datoteke predstavljaju tri **prava pristupa datotekama (file access control)**, na osnovu kojih se za svakog korisnika utvrđuje koje vrste rukovanja datotekom su mu dopuštene.
- Da se za svaku datoteku ne bi evidentirala prava pristupa za svakog korisnika pojedinačno, uputno je sve korisnike razvrstati u **kategorije** i za svaku od njih vezati pomenuta prava pristupa.
- Iskustvo pokazuje da su dovoljne **tri kategorije** korisnika. Jednoj pripada **vlasnik** datoteke, drugoj njegovi **saradnici**, a trećoj **ostali korisnici**.

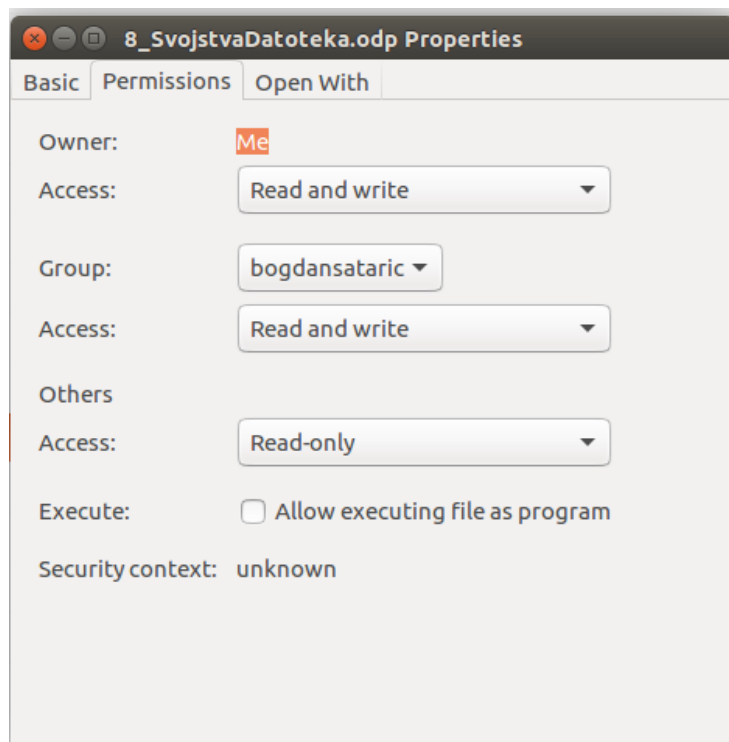
Matrica zaštite

- Ima **tri kolone** (po jednu za svaku kategoriju korisnika) i onoliko redova koliko ima datoteka.
- U preseku svakog reda i svake kolone matrice zaštite navode se prava pristupa datoteci iz posmatranog reda za korisnike koji pripadaju kategoriji iz posmatrane kolone.

	vlasnik	saradnik	ostali
datoteka_1.bin	pisanje čitanje izvršavanje	- čitanje izvršavanje	- - izvršavanje
datoteka_2.bin	- čitanje izvršavanje	- - izvršavanje	- - izvršavanje

datoteka_n.bin	pisanje čitanje -	- čitanje -	- - -

Matrica zaštite



```
bogdansataric@bogdansataric-CELSIUS-M740:~/Desktop/Grive/Predmeti/OS$ ls -la
total 4744
drwxrwxr-x 13 bogdansataric bogdansataric 4096 apr 19 11:28 .
drwxrwxr-x 8 bogdansataric bogdansataric 4096 dec 25 11:12 ..
drwxrwxr-x 5 bogdansataric bogdansataric 4096 okt 6 2016 2012-2013
drwxrwxr-x 6 bogdansataric bogdansataric 4096 okt 17 2016 2013-2014
drwxrwxr-x 8 bogdansataric bogdansataric 4096 okt 17 2016 2014-2015
drwxrwxr-x 11 bogdansataric bogdansataric 4096 okt 12 2016 2015-2016
drwxrwxr-x 10 bogdansataric bogdansataric 4096 jan 29 16:39 2016-2017
drwxrwxr-x 9 bogdansataric bogdansataric 4096 apr 18 13:33 2017-2018
drwxrwxr-x 6 bogdansataric bogdansataric 4096 mar 14 2017 Autotest alati i vezbe
-rw-r--r-- 1 bogdansataric bogdansataric 124883 mar 8 10:20 c++11_podesavanje_kompajlera.odt
-rw-rw-r-- 1 bogdansataric bogdansataric 351266 mar 8 10:20 c++11_podesavanje_kompajlera.pdf
drwx----- 11 bogdansataric bogdansataric 4096 dec 14 11:16 collbry_3.6.1-2
-rwxrwxrwx 1 bogdansataric bogdansataric 169 feb 19 12:28 cropar.sh
drwxrwxr-x 2 bogdansataric bogdansataric 4096 feb 7 2017 Knjige
-rw-rw-r-- 1 bogdansataric bogdansataric 108 apr 19 11:28 .-lock.OS_evidencija.docx#
-rw-rw-r-- 1 bogdansataric bogdansataric 1959058 feb 19 12:17 OS2017-2x1.pdf
-rw-rw-r-- 1 bogdansataric bogdansataric 1852907 feb 20 2017 OS2017.pdf
-rw-rw-r-- 1 bogdansataric bogdansataric 75658 feb 19 12:29 OS_cesta_pitanja_o_predmetu-2x1.pdf
-rw-rw-r-- 1 bogdansataric bogdansataric 27627 feb 19 11:55 OS_cesta_pitanja_o_predmetu.odt
-rw----- 1 bogdansataric bogdansataric 131872 feb 19 11:58 OS_cesta_pitanja_o_predmetu.pdf
-rw-rw-r-- 1 bogdansataric bogdansataric 11005 apr 19 11:28 OS_evidencija.docx
-rw-rw-r-- 1 bogdansataric bogdansataric 18197 mar 28 11:38 Pravila_za_studente.odt
-rw-rw-r-- 1 bogdansataric bogdansataric 72508 mar 28 11:42 Pravila_za_studente.pdf
drwxrwxr-x 2 bogdansataric bogdansataric 4096 apr 19 11:31 Predavanja
-rw-rw-r-- 1 bogdansataric bogdansataric 27306 mar 28 11:28 Uputstvo_za_asistente.odt
-rw-rw-r-- 1 bogdansataric bogdansataric 89744 mar 28 11:42 Uputstvo_za_asistente.pdf
-rw----- 1 bogdansataric bogdansataric 15715 okt 6 2016 uputstvo_za_c++_help.odt
drwxrwxr-x 5 bogdansataric bogdansataric 4096 dec 28 2016 Zadaci_za_testove
bogdansataric@bogdansataric-CELSIUS-M740:~/Desktop/Grive/Predmeti/OS$
```

Matrica zaštite

- Prava pristupa iz matrice zaštite se mogu vezati za:
 - datoteke i čuvati u **deskriptorima** datoteka
 - vezati za korisnike
- U prvom slučaju redovi matrice zaštite su raspoređeni po **deskriptorima** raznih datoteka, a u drugom slučaju elemente kolona matrice zaštite čuvaju **pojedini korisnici**.
- Za uspeh izloženog koncepta zaštite datoteka neophodno je **onemogućiti neovlašćeno menjanje matrice zaštite**.
- Jedino **vlasnik** datoteke sme da zadaje i menja prava pristupa sebi, svojim saradnicima i ostalim korisnicima. Zato je potrebno znati za svaku datoteku **ko je njen vlasnik**.

Matrica zaštite

- To se postiže tako što svoju aktivnost svaki korisnik započinje svojim **predstavljanjem (login)**.
- U toku predstavljanja korisnik predočava svoje **ime** i navodi dokaz da je on osoba za koju se predstavlja, za šta je, najčešće, dovoljna **lozinka**.
- Predočeno ime i navedena lozinka se porede sa **spiskom imena** i **spiskom** za njih vezanih **lozinki** registrovanih korisnika.
- Predstavljanje je uspešno, ako se u spiskovima imena i lozinki registrovanih korisnika pronađu **predočeno ime i navedena lozinka**.

Matrica zaštite

- Predstavljanje korisnika se zasniva na pretpostavci da su njihova **imena javna**, ali da su im **lozinke tajne**.
- Zato je i spisak imena registrovanih korisnika **javan**, a spisak lozinki registrovanih korisnika **tajan**.
- Jedina dva slučaja, u kojima ima smisla dozvoliti korisnicima posredan pristup **spisku lozinki**, su:
 - radi njihovog **predstavljanja**
 - radi **izmene** njihove lozinke

Matrica zaštite

- Za **predstavljanje** korisnika uvodi se **posebna operacija**, koja omogućuje samo proveru da li se zadani par (ime, lozinka) može pronaći u spiskovima imena i lozinki registrovanih korisnika.
- Slično, za **izmenu** lozinki uvodi se **posebna operacija**, koja omogućuje samo promenu lozinke onome **ko zna postojeću** lozinku.
- **Sva druga rukovanja** spiskovima imena i lozinki registrovanih korisnika, kao što su **ubacivanje** u ove spiskove novih parova (ime, lozinka), ili njihovo **izbacivanje** iz ovih spiskova, nalaze se u nadležnosti poverljive osobe, koja se naziva **administrator (superuser)**.

UID i GID

- Da bi se pojednostavila provera korisničkih prava pristupa, uputno je, umesto imena korisnika, uvesti njegovu **numeričku** oznaku.
- Radi klasifikacije korisnika zgodno je da ovu numeričku oznaku obrazuju **dva redna broja**.
- Prvi od njih označava **korisnika**, a drugi od njih označava **grupu** kojoj korisnik pripada.
- Podrazumeva se da su svi korisnici iz **iste grupe međusobno saradnici**.
- Prema tome, **redni broj** korisnika (**UID, User IDentification**) jednoznačno određuje vlasnika.
- Saradnici vlasnika su svi korisnici koji imaju **isti redni broj grupe (GID, Group IDentification)** kao i vlasnik.
- **Posebna grupa** se rezerviše za **administratore**.

Matrica zaštite

- Da se provera ne bi obavljala prilikom svakog pristupa datoteci, umesno je takvu proveru obaviti samo **pre prvog pristupa**.
- To je zadatak operacije **otvaranja datoteke**, koja **prethodi** operacijama, kao što su **pisanje** ili **čitanje** datoteke.
- Pomoću operacije otvaranja se saopštava i na koji način korisnik namerava da koristi datoteku.
- Ako je njegova namera **u skladu** sa njegovim pravima, otvaranje datoteke je **uspešno**, a pristup datoteci je dozvoljen, ali samo **u granicama** iskazanih namera.
- Iza operacija pisanja ili čitanja datoteke sledi operacija **zatvaranja datoteke**.

Matrica zaštite

- **Numerička oznaka vlasnika** datoteke i **prava pristupa korisnika** iz pojedinih klasa predstavljaju **atribute datoteke**.
- Zaštita datoteka uvodi pojam **sigurnost (security)** koji se odnosi na **uspešnost zaštite** od neovlašćenog korišćenja **ne samo datoteka**, nego i **ostalih delova računara** kojima upravlja operativni sistem.
- Sigurnost se bavi načinima prepoznavanja ili **identifikacije korisnika (authentication)**, kao i načinima provere njihovih **prava pristupa (authorization)**.
- Operativni sistem nudi mehanizme sigurnosti pomoću kojih mogu da se ostvare **različite politike sigurnosti**.

ext2fs

PRAKTIČAN PRIMER FAJL SISTEMA

Blokovi

ext2 smatra da se ono na čemu se zasniva sastoji od *sektora* (mi smo ih zvali blokovi na našem lažnom disku) koji se grupišu u *blokove*.

Jedan blok u ex2fs je par Kb koji mora biti stepen broja 2. Na mom disku to je 4kb.

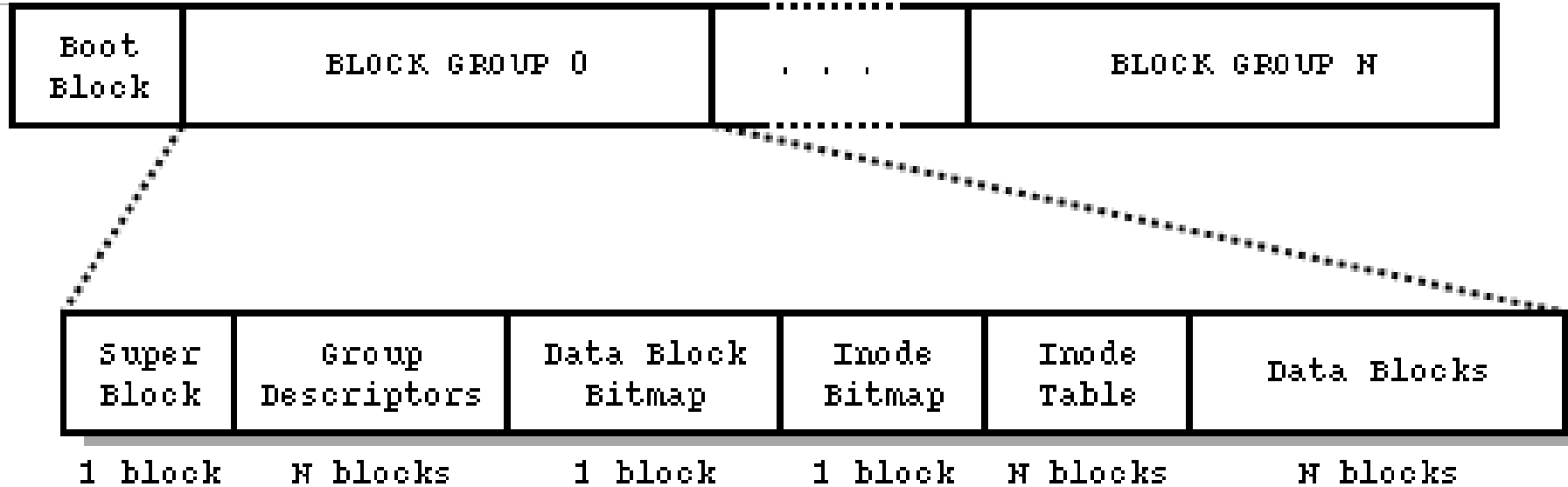
Blokovi se grupišu u *grupe* ne bi li se smanjila fragmentacija. Maksimalna veličina grupe je 8 x veličina bloka u blokovima, tj. jedna grupa na mom disku je $8 \times 4096 = 32\,768$ blokova. Ovo je neophodno zato što bitmapa, koja čuva šta je slobodno a šta nije zauzima samo jedan blok.

Superblok

Superblok sadrži sve informacije o tome kako je podešen fajl sistem. Primarno, to se nalazi 1024 bajtova daleko od početka diska, ali zato što je *apsolutno neophodno* da se ovi podaci ne izgube (inače nema šanse da se podaci sa particije vrate) superblok se replicira.

U starim verzijama ext2fs stavlja se kopija na početak svake grupe blokova. To gubi dosta prostora, te se danas tipično samo stavlja tek u poneku grupu, tipično stepene 3, 5, ili 7.

Grupa i superblok



Inode

Inoda (index node) je fundamentalan koncept za ext2 fajl sistem: sve o čemu fajl sistem vodi računa (fajl, direktorijum, link...) ima indeksni čvor, inodu. Taj čvor sadrži sve metapodatke o fajlu (tu se nalaze podaci o vlasniku, dozvolama, itd.) kao i, vrlo bitno, pokazivač ka blokovima sistema gde se fajlovi nalaze.

Sve inode se nalaze u inode tabelama, sa po jednom takvom tabelom po grupi blokova.

Fajl sistem pokazivači i inoda

Postavlja se pitanje, kako da znamo gde se sve na disku nalazi naš fajl? ext2 ima jednostavan pristup. Prvo čuva 12 pokazivača (broja bloka) sa tkzv. direktnim podacima. Ako fajl stane tu, dobro. Ako ne, ima i pokazivač ka jednostruko indirektnom bloku.

Taj pokazivač pokazuje na blok koji u sebi sadrži sekvencu pokazivača ka blokovima sa podacima.

Ako i dalje nema mesta? Onda imamo pokazivač ka *dvostruko* indirektnom bloku koji u sebi sadrži pokazivače ka blokovima koji u sebi imaju pokazivače ka blokovima podataka.

Fajl sistem pokazivači i inoda

Ako *i dalje* nema mesta? Onda imamo *trostruko* indirektnu adresu. To je pokazivač na blok koji sadrži listu pokazivača koji pokazuju na blokove koji sadrže liste pokazivača koje pokazuju na blokove koje imaju u sebi podatke.

Ovo znači da ima maksimalna veličina fajla a to je za sistem sa 32-bitnim pokazivačima na blok:

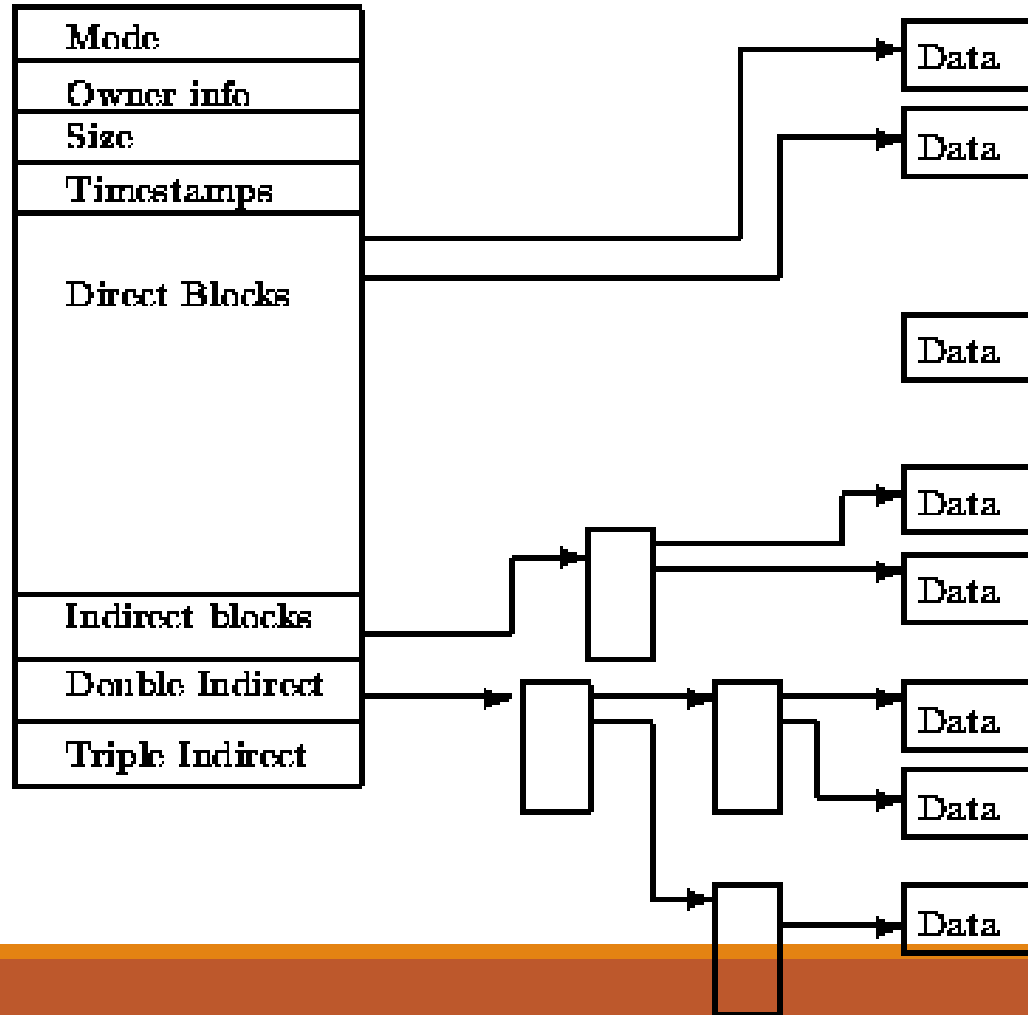
$12 * bs + (bs/4)*bs + (bs/4)*(bs/4)*bs + (bs/4)*(bs/4)*(bs/4)*bs$ bajtova,
gde je bs veličina bloka, tako da za 4kb blok maksimalna veličina bi bila oko 4402345721856 bajtova, odn. oko 4100GB.

Veličina fajla u praksi

U praksi, fajl je ograničen on-disk veličinom bloka (koja se razlikuje od ext2 bloka i iznosi 512 bajtova) i time što je sistem 32-bitni, te je maksimalan broj on-disk blokova u $2^{32} - 1$, što znači da je stvarni limit na veličinu fajla $(2^{32} - 1) * 512$, odn. 2TB

Inoda

ext2 inode



Direktorijumi i linkovi

Što se tiče ext2 sve je fajl.

Samo što određeni fajlovi imaju posebno interpretiran sadržaj, npr:

Linkovi su 'pokazivač' ka drugom fajlu

Ovo se odnosi na meke, simbolične linkove. Tvrdi linkovi su u stvari stavka u direktorijumu koja se odnosi na istu inod-u kao i neki drugi fajl.

Direktorijumi su lista pokazivača ka fajlovima

Da li se ovo koristi?

Tehnički, koriste se modernije verzije, ext3 i ext4 koje su u velikoj meri kompatibilne sa tim kako ext2 radi.

Ext je od 'extensible' tako da je oduvek bila ideja da fajl sistem može da se proširi.

Glavna promena koja je ubačena u ext3 je vođenje žurnala diska.

Žurnal fajl sistema—motivacija

Za određene operacije diska neželjen prekid (recimo gubljenjem struje) je *spektakularno* loša ideja. Ako brišemo fajl na disku očekujemo sledeće operacije u ext2:

1. Uklonimo zapis o fajlu iz sadržaja direktorijuma tog fajla.
2. Ako nema više tvrdih linkova na nju, oslobodimo inode tog fajla podešavanjem bitmape.
3. Oslobodimo sve blokove na koje pokazujemo tako što modifikujemo bitmape i metapodatke svih grupa gde se nalaze blokovi tog fajla.

Žurnal fajl sistema—motivacija

Za veliki fajl ovo su potencijalno *stotine* upisa na disk, a fajl sistem je samo konzistentan sam sa sobom na *početku* svih tih upisa i na *kraju*. Prekid bilo gde izaziva, npr. prostor koji ne koristi nijedan fajl, a koji je tehnički i dalje zauzet, ili inode koji se vodi kao zauzet a kome ne možete da priđete ili hiljadu drugih problema.

To znači da gubitak struje dovodi do potencijalnog oštećenja diska što moramo da (pokušavamo) da otklonimo posebnim komandama kao što je fsck. Ovo nije idealno.

Žurnal fajl sistema

Rešenje ovog problema jeste da postoji deo fajl sistema (žurnal) alociran i organizovan kao cirkularni bafer koji sadrži zapis svih promena koje hoćemo da uradimo i koji se povremeno prazni tako što se logovane operacije izvrše na disku.

Ako nestane struje u sred procesa, prilikom sledećeg aktiviranja fajl sistema (Unix to zove 'mount'-ovanje fajl sistema), se primeti da ima neizvršenih žurnal operacija, i postara se da su ili sve te operacije izvršene (te je sistem konzistentan) ili se od cele operacije odustalo.

Konzistentnost žurnala

Ovo nije savršeno rešenje. Prvi problem je: Šta ako nestane struje *baš* kada pišemo žurnal?

Rešenje je da se izračuna i prvo upiše kontrola vrednost (checksum) cele transakcije, tako da ako pukne upis pre kraja, to se može preko kontrolne vrednosti ustvrditi i ta transakcija ignorisati u žurnalu.

Konzistentnost žurnala

Drugi, mnogo veći problem, jeste interakcija između keširanja upisa na disk i žurnala. Žurnal se bazira na ideji da znamo redosled operacija upisa i da nećemo, recimo, početi da upisujemo podatke pre nego alociramo prostor kroz metapodatke.

I fajl sistem i sami disk uređaji često menjaju redosled operacija upisa zbog performansi. I mi smo to radili!

Ovo je rizik koji neki fajl sistemi, ext4, na primer, rešavaju uvođenjem barijera: tačka gde se keš nasilno spusti na sam disk.