Fajl sistemi

Operativni sistemi 2

Institut za matematiku i informatiku

Institut za matematiku i informatiku
Prirodno-matematički fakultet, Kragujevac

Novembar 2010. god.

O čemu će biti reči?

Pregled

Osnovne poželjne osobine fajlova:

1. Permanentno postojanje
2. Deljivost između procesa (*permissions*)
3. Unutrašnja struktura zavisna od aplikacije
4. Globalna struktura u obliku hijerarhije fajlova

Osnovne operacije sa fajlom:

1. Kreiranje
2. Brisanje
3. Otvaranje
4. Zatvaranje
5. Čitanje
6. Upis

Struktura i operacije sa fajlom

1. Polje (*field*) - Osnovni element podataka. Jedno polje - jedna vrednost
2. Zapis (*record*) - Zbirka polja sa kojima aplikacioni program postupa kao sa jedinicom
3. Fajl (*file*) - Zbirka povezanih podataka

Osnovne operacije sa fajlom

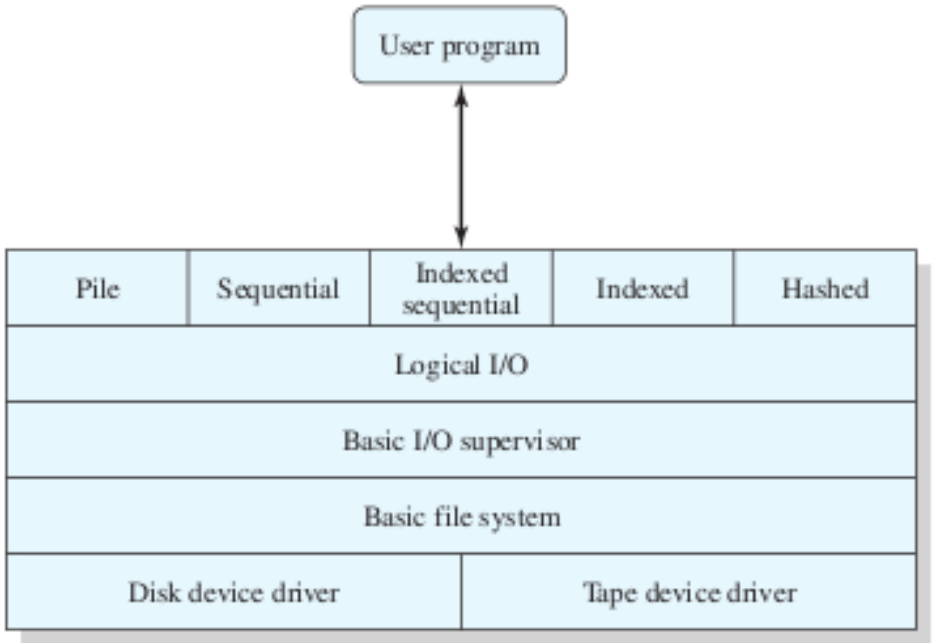
1. Izvuci_sve (sekvencijalna obrada)
2. Izvuci_jedan (transakcije)
3. Izvuci_sledeći
4. Izvuci_prethodni
5. Umetni_jedan (možda mora da se čuva redosled?)
6. Izbriši_jedan
7. Ažuriraj_jedan
8. Izvuci_nekoliko

Sa gledišta korisnika...

1. Kreiranje, brisanje, čitanje, upis, ažuriranje
2. Kontrolisan pristup fajlovima drugog korisnika
3. Kontrola dozvola za sopstvene fajlove
4. Restrukturiranje
5. Pomeranje podataka između fajlova
6. *Backup*
7. Pristup korišćenjem simboličkih imena

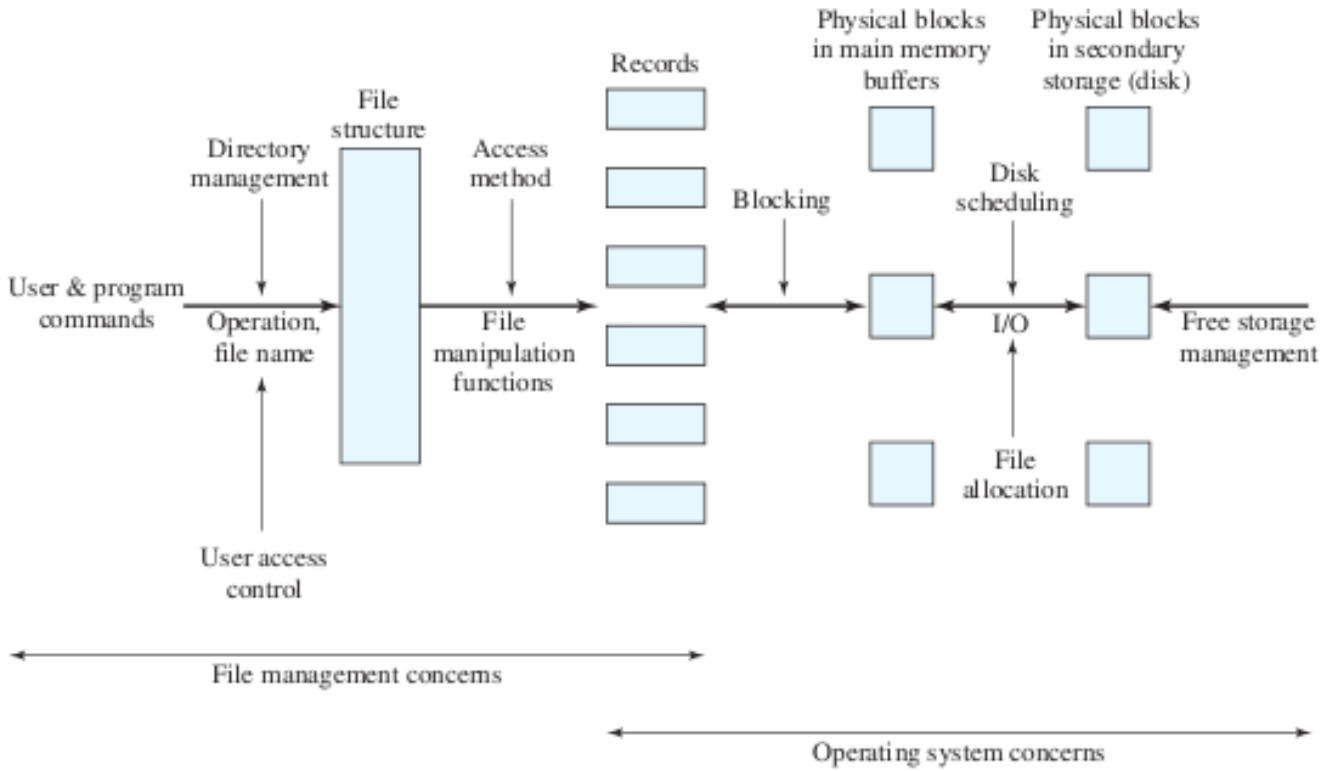
Arhitektura fajl sistema

1. Drajveri uređaja - komuniciraju direktno sa hardverom
2. Osnovni fajl sistem/fizički U/I - razmena i baferovanje *blokova*. Deo je OS-a
3. Logički U/I - radi sa *record*-ovima, a ne blokovima.



Arhitektura fajl sistema
Funkcije za fajl menadžment

1. Korisnik sagledava fajl sistem preko zapisa (*records*)
2. Sistem radi sa fajlovima preko *blokova*
3. Ova dva pristupa se sustiču na nivou zapisa



Organizacija fajlova i pristup

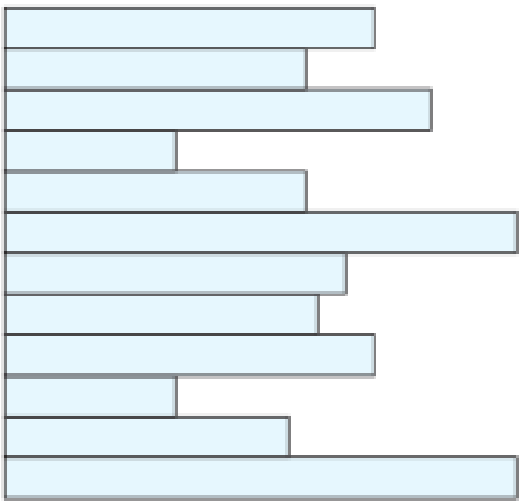
1. Logičko strukturiranje zapisa (*records*), ne fizičko!
2. Kratko vreme pristupa
3. Lakoća ažuriranja
4. Ekonomičnost (bez nepotrebnog taćenja prostora na *storage*-u)
5. Jednostavno održavanje
6. Pouzdanost

Prioritet kriterijuma

Na primer, na CD-ROM-u je potpuno nebitna lakoća ažuriranja, jer je ISO-9660 fajl sistem *read-only*, pa se i ne razmatra.

Gomila (hrpa) - *pile*

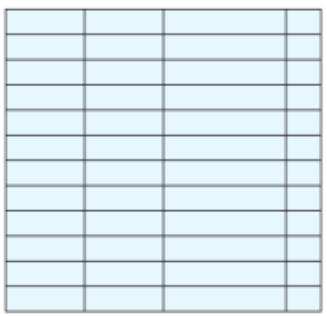
- Najjednostavniji oblik organizacije
- Podaci se pakuju u redosledu u kome stižu
- Pošto polja ne stižu redom, svako treba da sadrži samoopis
- U potrazi za zapisom pretražuje se ceo fajl
- Koristi se kada se često menja veličina i struktura fajla



Variable-length records
Variable set of fields
Chronological order

Sekvencijalni fajl

- Fiksni format, zapisi su iste dužine u određenom redosledu
- Kako je struktura poznata, u fajl idu samo vrednosti polja
- Obično prvo polje u zapisu je polje ključa
- Koristi se u paketnim aplikacijama, pretraživanje loše
- Ako ceo fajl može da se prenese u RAM, pretraživanje efikasnije
- Kod dodavanja zapisa, zapis se stavlja u poseban fajl *pile* tipa - fajl transakcija, a zatim periodično unose
- Alternativa je organizacija u obliku povezane liste



Fixed-length records
Fixed set of fields in fixed order
Sequential order based on key field

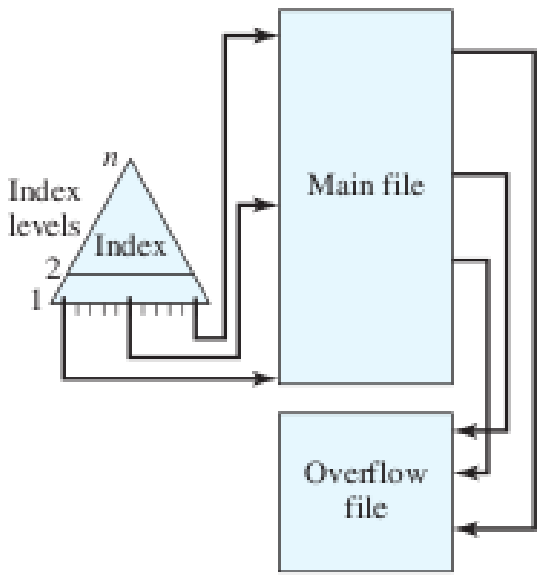
Indeksiran sekvencijalni fajl

- Dodaju se dva svojstva: indeks fajl i fajl prekoračenja (*overflow file*)
- Indeks obezbeđuje brzo stizanje u susedstvo traženog zapisa
- Fajl prekoračenja je sličan dnevniku (*journal*) koji ide uz sekvencijalni fajl, ali je integrisan
- Zapis se dodaje u fajl prekoračenja i ažuriraju se pokazivači
- Da bi se sekvencijalno obradio ceo fajl ide se redom dok se ne dođe do pokazivača na prekoračenje, nastavlja se sa obradom prekoračenja dok se ne dođe do NULL pokazivača, a zatim se vraća gde se stalo sa glavnim fajlom
- Da bi se još ubrzao pristup, indeks može da se podigne na više nivoa

Indeksiran sekvencijalni fajl
Nastavak

Primer (Performanse)

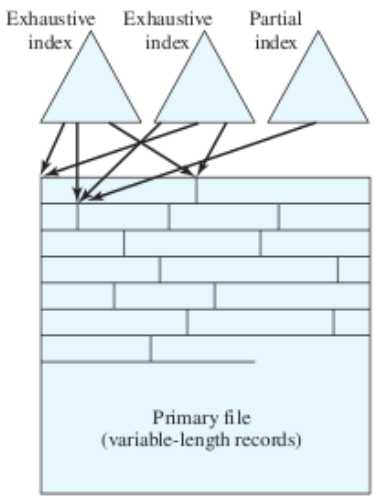
Fajl od 1000000 zapisa. Da bi se pronašao pojedinačni zapis, u proseku je potrebno 500000 pristupa. Međutim, ako postoji indeks fajl od 1000 stavki, sada je potrebno 500 pristupa indeksu i još 500 pristupa fajlu da bi se došlo do željenog zapisa.



(c) Indexed sequential file

Indeksni fajl

- Ponekad je potrebna pretraga po atributu koji nije ključ
- Za svako polje od intresa dodaje se poseban indeks
- Napušta se koncept sekvencijalnosti i ključa
- Iscrpan indeks sadrži po stavku za svaki zapis
- Delimičan indeks sadrži stavke za zapise u kojima postoji polje od interesa
- Moguće su čak i različite dužine zapisa



(d) Indexed file

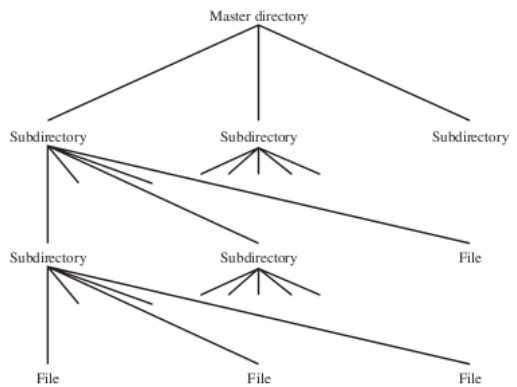
Direktni (heširani) fajl

- Koristi sposobnost diskova da direktno pristupe svakom bloku s poznatom adresom
- NE postoji koncept sekvencijalnog uređivanja
- Koristi se heširanje vrednosti ključa
- Primenjuje se gde su zapisi fiksne dužine, gde se traži brz pristup i gde se pristupa samo po jednom zapisu istovremeno

Direktorijumi

Direktorijum sadrži informacije o fajlovima, kao što su atributi, lokacija i vlasništvo. Operacije:

1. Pretraživanje
2. Kreiranje fajlova
3. Brisanje fajlova
4. Listanje direktorijuma (*find*)
5. Ažuriranje direktorijuma (promena atributa fajlova)
6. Jedno rešenje je sekvencijalni fajl, ali ako su liste velike, koristi se *hash*-irana struktura



Deljenje fajlova
Prava pristupa

1. Za vlasnika, grupu i ostale
2. Prava čitanja, pisanja i izvršavanja

	pristupna prava za fajlove	pristupna prava za direktorijume
read (r)	korisnik može pročitati sadržaj fajla, odnosno može prikazivati fajl na ekranu, štampati ga ili kopirati;	korisnik može pročitati sadržaj direktorijuma, što znači i da korisnik može da izvrši komandu ls. **Napomena:** za prikazivanje detaljnog listinga direktorijuma (ls -l) neophodno je i x pravo nad direktorijumom
write (w)	korisnik može modifikovati sadržaj fajla. **Napomena:** može obrisati fajl samo ako mu je dato pravo upisa nad roditeljskim direktorijumom;	korisnik može modifikovati sadržaj direktorijuma, odnosno dodavati nove fajlove i brisati postojeće, kreirati i brisati poddirektorijume. **Napomena:** može obrisati direktorijum samo ako mu je dato pravo upisa nad roditeljskim direktorijumom;
execute (x)	korisnik može izvršavati fajl, pod uslovom da se radi o shell programu ili o fajlu u binarnom izvršnom formatu;	korisnik se može pozicionirati na direktorijum (komandom cd), može prikazivati dugački listing (ls -l) sadržaja i pretraživati direktorijum (find).

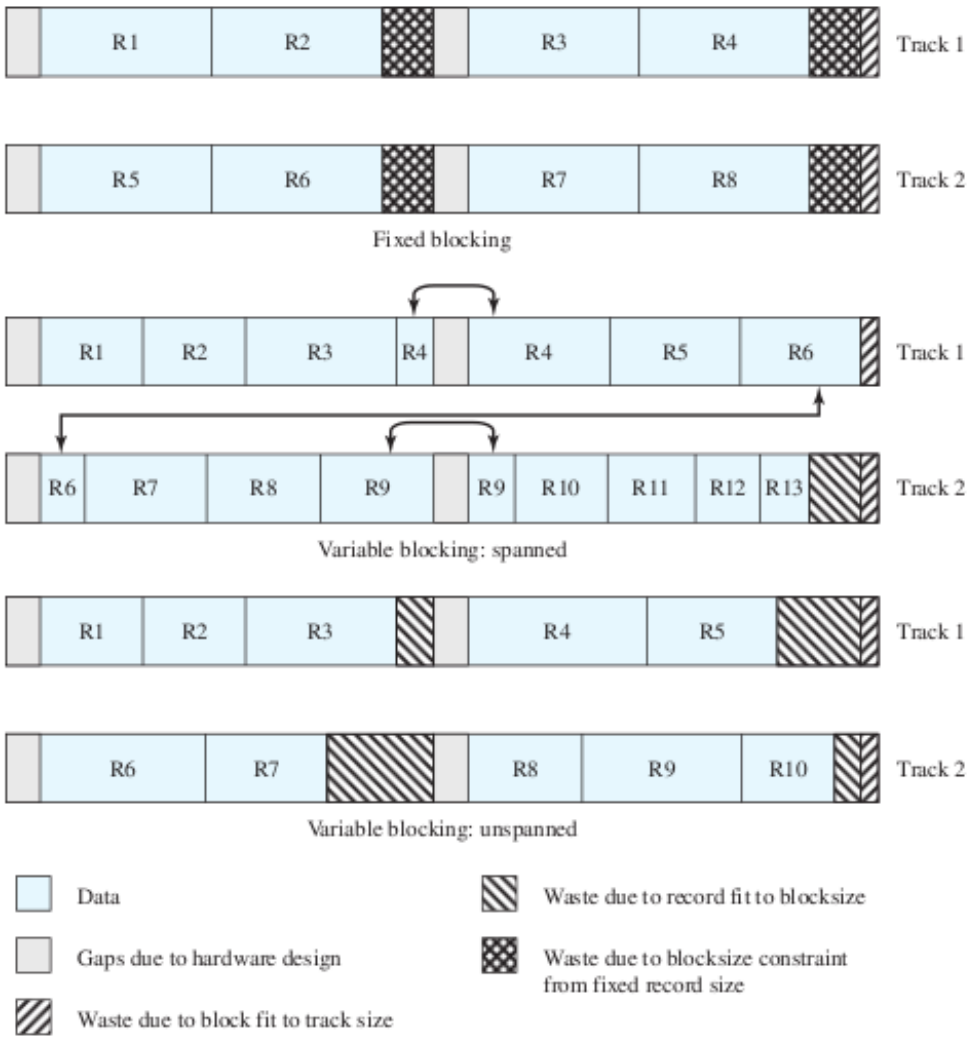
Blokovanje fajlova

- U većini sistema, blokovi su fiksne dužine
- Veći blok - više zapisa se transferuje u jednoj U/I operaciji
- Manji blok - rentabilniji prenos ako se zahteva manji broj zapisa

Vrste blokovanja

1. Fiksno blokovanje - unutrašnja fragmentacija
2. Blokovanje promenljive dužine sa premošćenjem - nema neiskorišćenog prostora
3. Blokovanje promenljive dužine bez premošćenja - postoji neiskorišćen prostor

Blokovanje fajlova



Upravljanje sekundarnim *storage*-om

Dodlejivanje fajlova

1. Kada se kreira fajl, da li se maksimalan prostor za nju traži odmah?
2. Prostor se dodeljuje fajlu kao jedna ili više susednih jedinica, tj. delova. Koju veličinu dela upotrebiti? Krajnje mere su **jedan blok** i **ceo fajl**.
3. Koja struktura se koristi za vođenje evidencije o dodeljenim delovima?

Dodeljivanje unapred ili dinamičko dodeljivanje?

Ponekad je veličina fajla koji treba da se kreira poznata unapred (kopiranje, kompajliranje), ali u većini upotreba nije.

Veličina dela

Dve krajnosti: Ceo fajl u jednom delu i po jedan blok istovremeno. Veliki promenljivi delovi daju bolje performanse, manje su tabele alokacije. Mali delovi obezbeđuju bolju fleksibilnost.

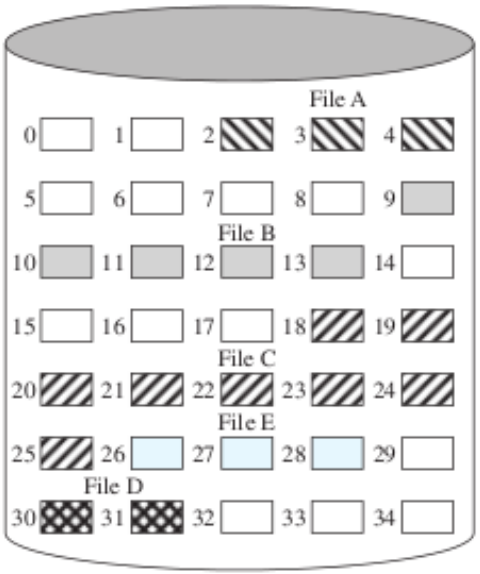
Dodeljivanje fajlova

Kod sistema sa delovima promenljive veličine

1. Prvi odgovarajući - Izabrati prvu neiskorišćenu grupu blokova dovoljne veličine.
2. Najboji odgovarajući - Izabrati najmanju neiskorišćenu grupu blokova
3. Najbliži odgovarajući - Izabrati najbližu neiskorišćenu grupu blokova

Modeli dodeljivanja fajlova
Susedno dodeljivanje

- Skup susednih blokova se dodeljuje fajlu u trenutku kreiranja
- Za svaki fajl treba da se zna samo početni blok i dužina
- Zgodno za sekvencijalne fajlove
- Spoljašnja fragmentacija - defragmentacija je rešenje



File Allocation Table

File Name	Start Block	Length
File A	2	3
File B	9	5
File C	18	8
File D	30	2
File E	26	3

Figure 12.7 **Contiguous File Allocation**

Modeli dodeljivanja fajlova
Ulančano dodeljivanje

- Na nivou pojedinačnih blokova
- Nema spoljašnje fragmentacije
- Nema saglasnosti sa principom lokalnosti, pa se periodično defragmentira

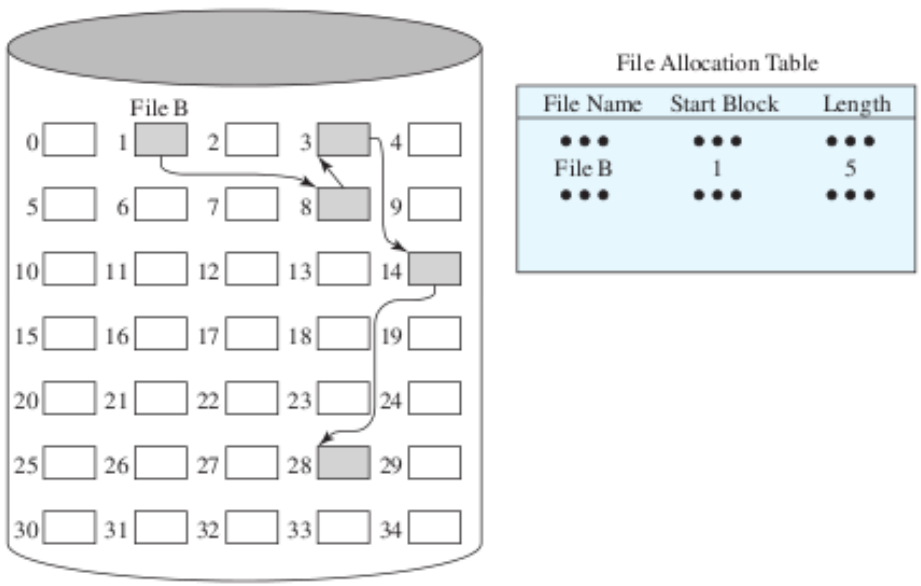


Figure 12.9 **Chained Allocation**

Modeli dodeljivanja fajlova
Indeksno dodeljivanje

- Tabela sadrži poseban indeks u jednom nivou za svaki fajl
- Dodljeivanje može da bude ili na bazi blokova fiksne veličine ili delova promenljive veličine
- Ako su delovi promenljive veličine, defragmentacija smanjuje tabelu

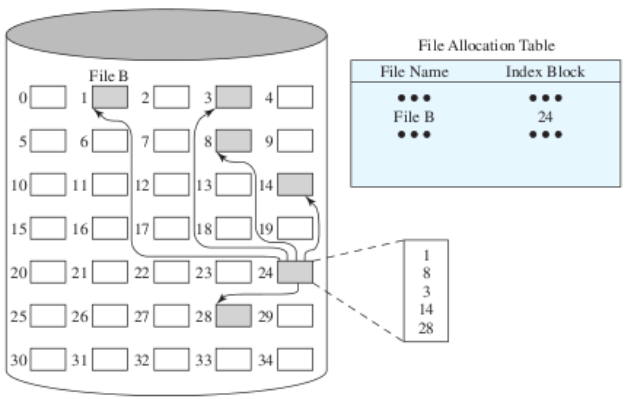


Figure 12.11 Indexed Allocation with Block Portions

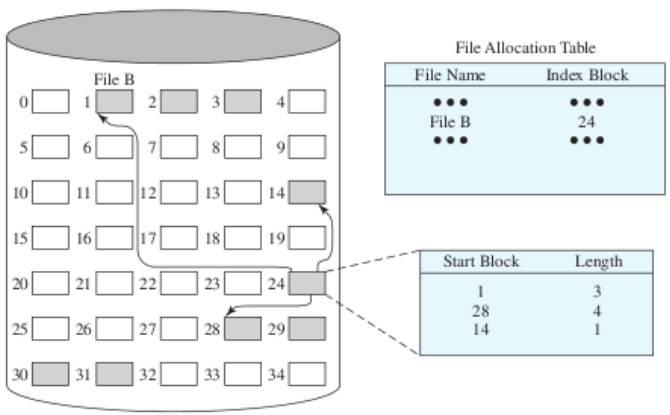


Figure 12.12 Indexed Allocation with Variable-Length Portions

Modeli dodeljivanja fajlova

Poređenje pristupa

	Contiguous	Chained	Indexed	
Preallocation?	Necessary	Possible	Possible	
Fixed or variable size portions?	Variable	Fixed blocks	Fixed blocks	Variable
Portion size	Large	Small	Small	Medium
Allocation frequency	Once	Low to high	High	Low
Time to allocate	Medium	Long	Short	Medium
File allocation table size	One entry	One entry	Large	Medium

Upravljanje slobodnim prostorom

Tabela bitova

Vektor koji sadrži po jedan bit za svaki blok. Disk od 16GB sa blokom od 512b, tabela blokova zauzima oko 4MB RAM-a, što je previše.

Ulančani slobodni delovi

Korišćenjem pokazivača i vrednosti dužine svakog slobodnog dela. Problem je ažuriranje pokazivača i kada ima mnogo fragmentiranih delova.

Indeksiranje

Slično kao kod fajlova, samo što bi indeks trebalo da sadrži promenljive delove zbog performansi

Lista slobodnih blokova

Svakom bloku se dodeljuje broj, npr 32-bitni. Zadovoljavajući metod. Prostor na disku za tabelu je manji od 1% ukupnog prostora. U RAM-u se drži mali deo liste u obliku potisnog steka ili FIFO reda.

Fajl menadžment u sistemu UNIX

Vrste fajlova

1. Regularni
2. Direktorijum
3. Specijalni (`/dev`)
4. Imenovani tokovi (`mkfifo`)
5. Linkovi
6. Simbolički linkovi

Indeksni čvor

Upravljačka struktura koja sadrži ključne informacije koje su potrebne OS-u za dati fajl. Više imena fajlova može da se pridruži jednom čvoru.

Fajl menadžment u sistemu UNIX

Dodeljivanje fajlova

1. Dodeljivanje je dinamičko, po blokovima, bez predviđanja
2. Indeksni čvor sadrži 13 3-bajtnih pokazivača

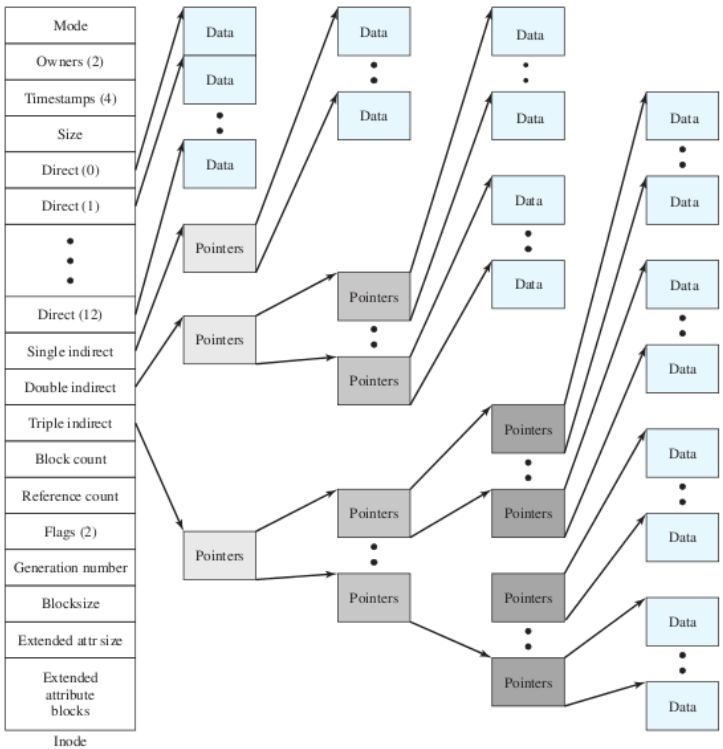


Figure 12.14 Structure of FreeBSD inode and File

Fajl menadžment u sistemu Linux

Linux *Virtual File System*

- Jednoobrazni interfejs raznorodnih fajl sistema prema korisničkim procesima
- Neki fajl sistemi kao FAT32 čak ne pamte direktorijum kao fajl
- Tipovi u okviru OO arhitekture su: **Superblok** (monirani fs), **Indeksni čvor** (fajl), **Dentry** (zapis u direktorijumu) i **Fajl** (otovreni fajl pridružen procesu)

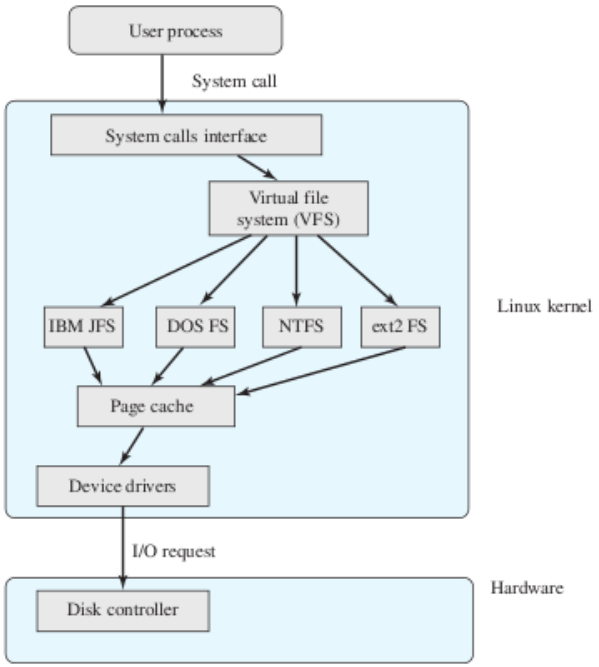
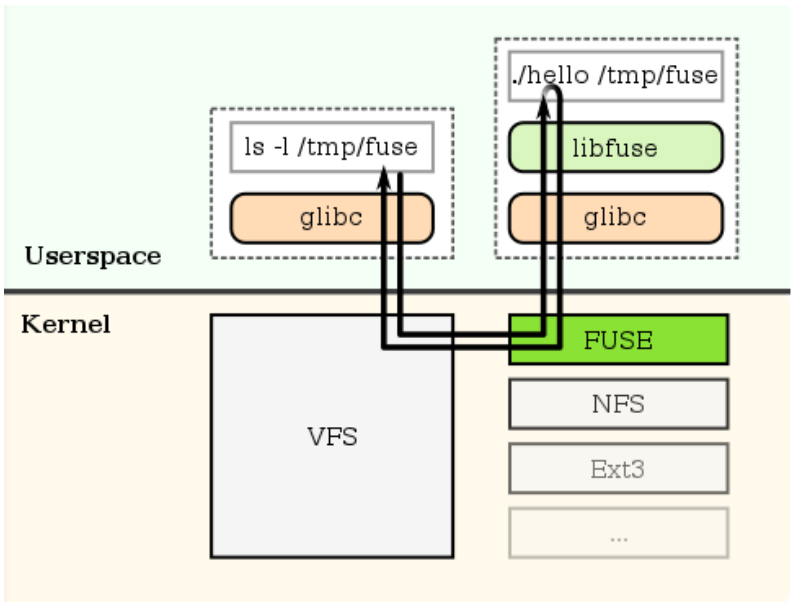


Figure 12.17 **Linux Virtual File System Context**

Fajl menadžment u sistemu Linux

- FUSE, GVFS - File System in User Space (most ka SFTP, WebDAV, FTP, SMB,...)
- http://goo.gl/wc9lue
- `tune2fs` komanda
- ACL (*Access Control Lists*) - http://goo.gl/znP8uO



Windows NTFS fajl sistem

1. Obnovljivost - transakcije
2. Bezbednost
3. Veliki diskovi i veliki fajlovi
4. Opšte sredstvo za indeksiranje - relaciona baza podataka atributa