



Operativni sistemi

- Upravljanje datotekama -

Prof. dr Dragan Stojanović

Katedra za računarstvo
Univerzitet u Nišu, Elektronski fakultet



Literatura

- ✿ *Operating Systems: Internals and Design Principles*, edition, W. Stallings, Pearson Education Inc., 7th – 2012, (5th -2005, 6th - 2008, 8th – 2014 , 9th – 2017)

- ✦ <http://williamstallings.com/OperatingSystems/>
- ✦ <http://williamstallings.com/OperatingSystems/OS9e-Student/>

- ✿ **Poglavlje 12: Upravljanje datotekama**



Sistem za upravljanje datotekama

- ❖ **Datoteka** (*file*) je imenovana kolekcija povezanih podataka zapisana na sekundarnoj memoriji (magnetni diskovi, magnetne trake, optički diskovi, USB flash, itd.)
- ❖ **Sistem za upravljanje datotekama** (**Datotečni sistem**, *File management sistem*, *File system*) je deo OS-a zadužen za upravljanje datotekama koji obezbeđuje pogodan interfejs za korisnike datoteka
- ❖ Detalji o organizaciji i implementaciji sistema za upravljanje datotekama nisu bitni za korisnika, ali jesu za projektante OS-a i sistem programere

Sistem za upravljanje datotekama

- ✿ *File system, File management sistem*
- ✿ Omogućuje korisnicima kreiranje datoteke kao kolekcije podataka, koja ima sledeće svojstva:
 - ✦ Dugoročno postojanje
 - ✦ Deljiva je između procesa
 - ✦ Posедује odgovarajuću internu strukturu
- ✿ Obezbeđuje operacije nad datotekama
 - ✦ Kreiranje (*Create*)
 - ✦ Brisanje (*Delete*)
 - ✦ Otvaranje (*Open*)
 - ✦ Zatvaranje (*Close*)
 - ✦ Čitanje (*Read*)
 - ✦ Upis (*Write*)

Struktura datoteke



Datoteka može biti struktuirana na dva različita načina:

1. Sekvenca (niz, *stream*) **bajtova** (Unix, Unix-like OS i Windows)
2. Sekvenca **slogova** (*records*) fiksne ili promenljive dužine
 - **Polje** (*field*) predstavlja elementarni podatak u okviru sloga, koji se karakteriše imenom, dužinom polja i tipom podatka (ime zaposlenog, datum zaposlenja, itd.). Može biti fiksne ili promenljive dužine
 - **Slog** je kolekcija polja koja se tretiraju kao celina od strane aplikacionog programa; ukoliko su polja u okviru sloga promenljive dužine, i slog je promenljive dužine
 - **Datoteka** predstavlja kolekciju povezanih slogova koja se referencira imenom
 - **Baza podataka** predstavlja kolekciju podataka u datotekama koji su eksplicitno povezani i koji se odnose na određeni domen (npr. podaci o fakultetu, profesorima, asistentima, studentima, predmetima, itd.)

Atributi datoteke

- ✿ Skup atributa se razlikuje od jednog do drugog OS
- ✿ Obično su to:
 - ✦ *Ime datoteke*: simboličko ime čitljivo za čoveka
 - ✦ *Identifikator (descriptor)*: jedinstvena oznaka datoteke, obično broj, kojom se identifikuje fajl unutar fajl sistema; nije čitljiv za čoveka
 - ✦ *Tip*: neophodan kod OS-a koji podržavaju različite tipove
 - ✦ *Lokacija*: pokazivač na uređaj i lokaciju datoteke na tom uređaju
 - ✦ *Veličina*: tekuća veličina datoteke u bajtovima ili blokovima
 - Ponekad se kao atribut uključuje max dozvoljena veličina datoteke
 - ✦ *Zaštita*: informacija o kontroli pristupa datoteci kojom se određuje ko može čitati, upisivati, izvršavati, itd. datoteku
 - ✦ *Vreme, datum i identifikacija korisnika*: odnose se na kreiranje, zadnju modifikaciju i zadnje korišćenje; korisno za zaštitu, sigurnost i monitoring korišćenja
- ✿ Informacije o datotekama se čuvaju u
 - ✦ Direktorijumima,
 - ✦ Posebnoj strukturi podataka (*i-čvor*, *FCB*)

Upravljanje datotekama



Sistemi za upravljanje datotekama

- ❖ Sistem za upravljanje datotekama je onaj deo operativnog sistema koji korisnicima i aplikacijama obezbeđuje usluge u pogledu upotrebe datoteka
- ❖ Programer ili korisnik nema potrebe da razvija softver posebne namene za svaku aplikaciju za upravljanje datotekama

Ciljevi sistema za upravljanje datotekama

- ✿ Zadovoljiti potrebe i **zahteve korisnika za upravljanje podacima** (skladištenje podataka i operacije za rad sa datotekama)
- ✿ Garantovati da su podaci u datotekama **validni**
- ✿ Optimizovati **performanse** (propusna moć, vreme odziva)
- ✿ Obezbediti U/I podršku za **razne vrste** uređaja za skladištenje podataka
- ✿ Svesti na najmanju meru ili odstraniti mogućnost **gubitka** ili uništenja podataka
- ✿ Obezbediti standardizovani **skup U/I funkcija** kao interfejsa za korisničke procese
- ✿ Obezbediti U/I podršku za **više korisnika** (kod višekorisničkih sistema)

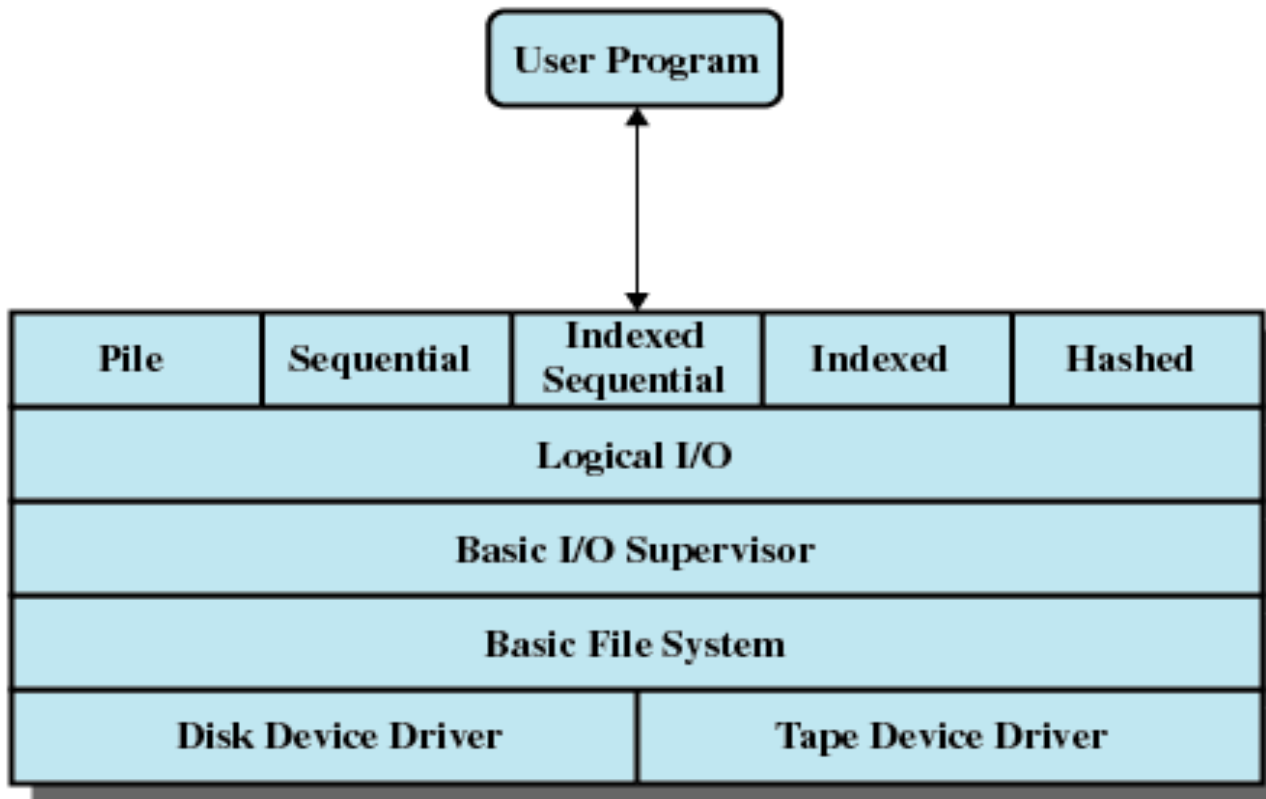


Zahtevi korisnika za upravljanje podacima

1. Svaki korisnik bi trebalo da može da **kreira, briše, čita, upisuje i modifikuje** datoteke
2. Svaki korisnik može da ima **kontrolisani pristup** datotekama drugih korisnika
3. Svaki korisnik može da kontroliše koje **vrste pristupa** korisničkim datotekama su dozvojene
4. Svakom korisniku bi trebalo omogućiti da **restruktura** korisničke datoteke u oblik koji odgovara konkretnom problemu
5. Svakom korisniku bi trebalo omogućiti da **pomera** podatke između datoteka
6. Svakom korisniku bi trebalo omogućiti da **pravi rezervne kopije** i da **obnavlja** korisničke datoteke u slučaju oštećenja
7. Svakom korisniku bi trebalo omogućiti da **pristupa** korisničkim datotekama koristeći **simbolička imena**



Arhitektura sistema datoteka (*File system-a*)





Upravljački programi

Device Drivers

- ✿ Najniži nivo
- ✿ Komunicira direktno sa perifernim uređajem
- ✿ Odgovoran je za startovanje U/I operacija na uređaju
- ✿ Obradjuje kompletiranje U/I zahteva
- ✿ Deo je OS-a

Osnovni sistem datoteka

- ✿ Fizički U/I sloj
- ✿ Primarni interfejs sa okolinom izvan računarskog sistema
- ✿ Radi sa blokovima podataka koji se razmenjuju sa sistemima diska ili trake
- ✿ Bavi se smeštanjem blokova na uređaju sekundarne memorije i baferovanjem tih blokova u glavnoj memoriji
- ✿ Ne razume sadržaj podataka, niti strukturu datoteka koji su uključeni u U/I
- ✿ Deo je OS-a



Osnovni U/I supervizor

- ✿ Odgovoran je za svako iniciranje i završavanje operacije U/I datoteke
- ✿ Održava upravljačke strukture koje su odgovorne za
 - ✦ U/I uređaja
 - ✦ Planiranje diska
 - ✦ Status datoteka
- ✿ Bira uređaj na kome treba da se izvrši U/I datoteke zavisno od selektovane datoteke
- ✿ Bavi se raspoređivanjem diska i trake u cilju optimizovanja performansi
- ✿ Na ovom nivou se dodeljuju U/I baferi i alocira potrebna sekundarna memorija
- ✿ Deo je OS



Logički U/I

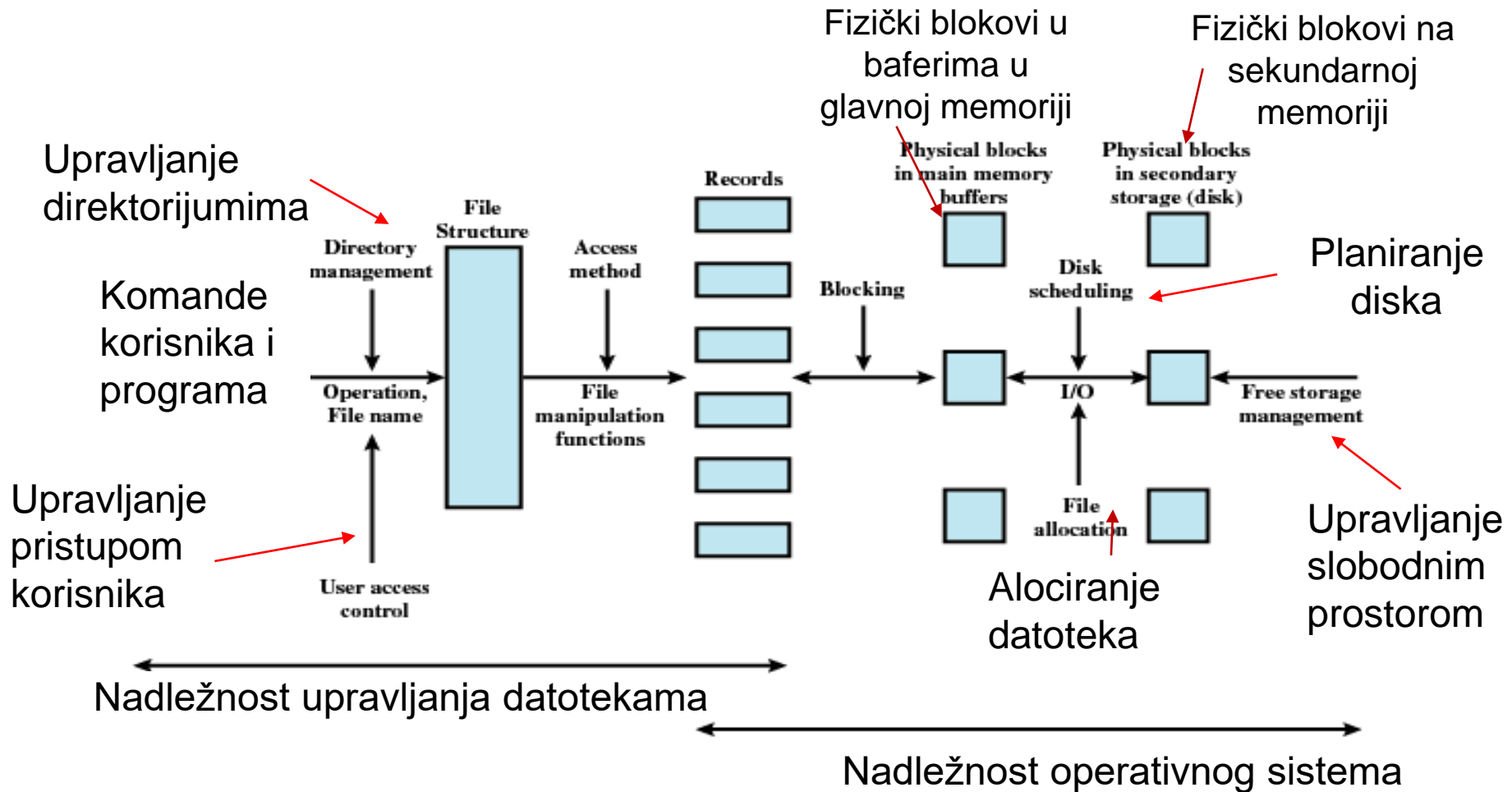
- ✿ Omogućava korisnicima i aplikacijama da pristupaju slogovima (*records*)
- ✿ Obezbeđuje U/I podršku opšte namene za slogove
- ✿ Održava osnovne podatke o datotekama



Metode pristupa

- ❁ Sloj *file system-a* najbliži korisnicima
- ❁ Obezbeđuje standardni interfejs između aplikacija i datotečnog sistema i uređaja na kojima su smeštene datoteke
- ❁ Uzima u obzir strukturu datoteka (serijsku, sekvencijalnu,...)
- ❁ Za različite strukture datoteka obezbeđuje različite metode pristupa i obrade podataka

Funkcije za upravljanje datotekama





Funkcije upravljanja datotekama

- ✿ Identifikuje i locira datoteku nad kojom se zahteva operacija od strane korisnika ili aplikacionog programa
 - ✚ korišćenjem direktorijuma kojim se opisuju lokacije svih datoteka i njihovi atributi
- ✿ Ako je dozvoljeno deljenje datoteka opisuje pristupne privilegije
- ✿ Vrš grupisanje slogova u blokove i obrnuto za pristup datotekama radi čitanja i upisa
- ✿ Vrš dodelu slobodnih blokova datotekama
- ✿ Bavi se raspoređivanjem diska i alociranjem datoteke u cilju optimizovanja performansi
- ✿ Upravlja slobodnim prostorom na disku (blokovi)

Organizacija i pristup datotekama

- ✿ U izboru organizacije datoteke važni su sledeći kriterijumi
 - ✦ Kratko vreme pristupa
 - ✦ Lako ažuriranje
 - ✦ Ekonomičnost smeštanja datoteke
 - ✦ Jednostavno održavanje
 - ✦ Pouzdanost
- ✿ Tipovi organizacija datoteke:
 - ✦ Gomila (*pile*)
 - ✦ Sekvencijalna datoteka
 - ✦ Indeksirano-sekvencijalna (indeks-sekvencijalna) datoteka
 - ✦ Indeksirana datoteka
 - ✦ Direktna (heširana) datoteka

Tipovi organizacije datoteka i pristupa (1)

✿ Gomila

- ✦ Podaci su organizovani u redosledu u kom su stizali, pri čemu slogovi mogu imati različit redosled polja ili različita polja
- ✦ Pristup podacima (slogovima) zahteva iscrpno pretraživanje

✿ Sekvencijalna datoteka

- ✦ Slogovi su fiksnog formata
- ✦ Slogovi mogu imati polje koje predstavlja ključ koje ga jedinstveno identifikuje
- ✦ Novi slogovi dodaju se na kraj datoteke, ili na određeno mesto ukoliko je datoteka sortirana po ključu
- ✦ Pogodan kada je medijum magnetna traka ili disk
- ✦ Loše performanse za interaktivni pristup koji uključuje upite i pretraživanja individualnih slogova



Tipovi organizacije datoteka i pristupa (2)

Indeksna sekvencijalna datoteka

- Organizuje slogove u sekvenci u skladu sa vrednošću ključa
- Za podršku *random* pristupu dodaje se indeksna datoteka pomoću koje se na osnovu ključa pristupa slogovima u blizini traženog sloga
- Indeksna datoteka je sekvencijalna i svaki slog sadrži dva polja: ključ (isti kao u glavnoj datoteci) i pokazivač na glavnu datoteku
- Novi slog se dodaje u tzv. *overflow* datoteku, i na njih ukazuje odgovarajući slog glavne datoteke koji mu neposredno prethodi na osnovu vrednosti ključa
- Moguće je indeksiranje u dva ili više nivoa



Tipovi organizacije datoteka i pristupa (3)



Indeksna datoteka

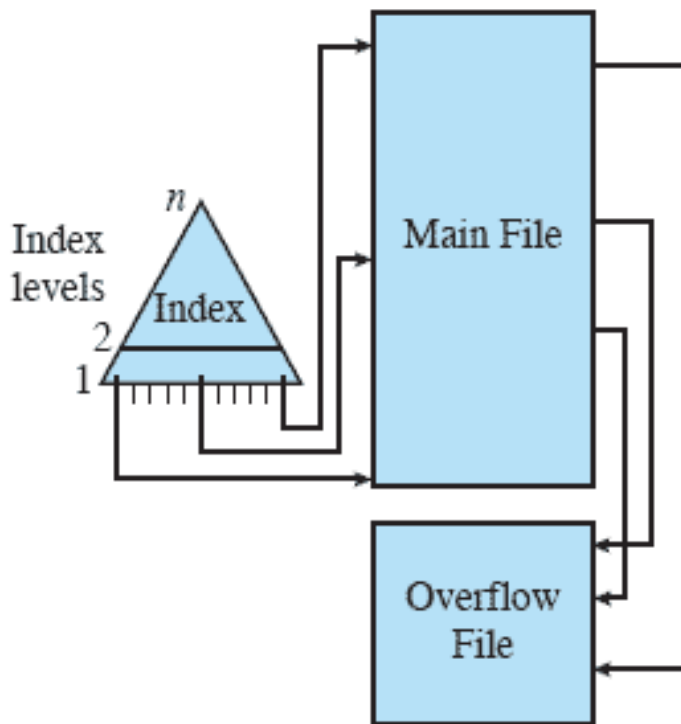
- ❖ Višestruki indeksi za svako polje po kome se vrši pretraživanje
- ❖ Sekvencijalnost ne postoji, slogovima se pristupa isključivo putem indeksa
- ❖ Indeks može biti iscrpni koji sadrži jedan ulaz za svaki slog u datoteci, ili delimični koji ukazuje na grupu slogova u okviru kojih postoji slog sa traženom vrednošću polja
- ❖ Koriste se u aplikacijama u kojima je zahteva brz pristup podacima



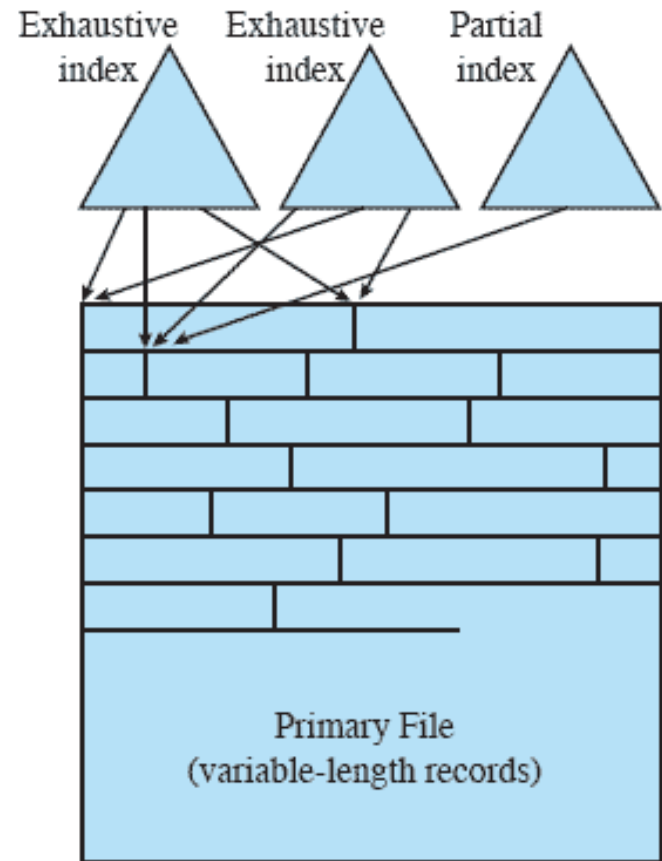
Direktna (random, heširana) datoteka

- ❖ Zasnivaju se na karakteristikama diska kojima se omogućava direktan pristup bilo kom bloku sa poznatom adresom
- ❖ Bajtovi/slogovi se čitaju ili zapisuju u bilo kom poretku
- ❖ Svaki slog ima ključ, a direktan pristup je omogućen na osnovu heširanja po vrednosti ključa
- ❖ Bitno za sisteme baza podataka
- ❖ Pogodan za aplikacije koje zahtevaju brz pristup slogovima fiksne dužine i formata

Indeksno-sekvencijalna i indeksna datoteka



(c) Indexed Sequential File



(d) Indexed File

Performanse

● Ocene performansi 5 osnovnih organizacija datoteka

	Space Attributes		Update Record Size		Retrieval		
File Method	Variable	Fixed	Equal	Greater	Single record	Subset	Exhaustive
Pile	A	B	A	E	E	D	B
Sequential	F	A	D	F	F	D	A
Indexed sequential	F	B	B	D	B	D	B
Indexed	B	C	C	C	A	B	D
Hashed	F	B	B	F	B	F	E

- A = Excellent, well suited to this purpose $\approx O(r)$
 B = Good $\approx O(o \times r)$
 C = Adequate $\approx O(r \log n)$
 D = Requires some extra effort $\approx O(n)$
 E = Possible with extreme effort $\approx O(r \times n)$
 F = Not reasonable for this purpose $\approx O(n^{>1})$

where

- r = size of the result
 o = number of records that overflow
 n = number of records in file



Operacije nad datotekama (1)

1. Kreiranje (*Create*)
 2. Brisanje (*Delete*)
 3. Otvaranje (*Open*)
 4. Zatvaranje (*Close*)
 5. Čitanje (*Read*)
 6. Upis (*Write*)
 7. Dodavanje (*Append*)
 7. Traženje (*Seek*)
 8. Uzimanje atributa (*Get attributes*)
 9. Postavljanje atributa (*Set Attributes*)
 10. Preimenovanje (*Rename*)
- Neke operacije može direktno pozivati krajnji korisnik - *komande*
 - Mogu se pozivati iz programa - *sistemski pozivi*
 - Postoje i drugi sistemski pozivi za rad sa datotekama
 - Detalji u Praktikum

Operacije nad datotekama (2)

✚ Kreiranje

- ✚ Datoteci se dodeljuje ime i neki atributi
- ✚ Datoteka je bez podataka
- ✚ Evidentira se u sistemske tablice

✚ Brisanje

- ✚ Oslobađa se prostor koji datoteka zauzima
- ✚ Uklanja se iz sistemskih tablica

✚ Otvaranje

- ✚ Pre upotrebe proces mora otvoriti datoteku
- ✚ Omogućava se sistemu da pristupa atributima datoteke
- ✚ Kreira se upravljački blok procesa (File Control Block - FCB)- struktura podataka u memoriji preko koje sistem brže pristupa datoteci
- ✚ OS ograničava broj otvorenih datoteka

✚ Zatvaranje

- ✚ Datoteka se zatvara kada više nije potrebna
- ✚ Oslobađa se prostor u internim tablicama

Operacije nad datotekama (3)

✚ Čitanje

- ✚ Čitaju se podaci iz datoteke, obično od tekuće pozicije
- ✚ Specificira se koliko bajtova/slogova treba pročitati i gde smestiti pročitane podatke

✚ Upis

- ✚ U datoteku se upisuju podaci, obično od tekuće pozicije
- ✚ Specificira se koliko bajtova/slogova treba zapisati i podaci koje treba zapisati

✚ Dodavanje

- ✚ Vrsta upisa, ali na kraj datoteke

✚ Traženje (*seek*)

- ✚ Kod random datoteka specificira se pozicija odakle će početi upis ili čitanje

✚ Pribavljanje atributa

- ✚ Čitaju se atributi datoteke

✚ Postavljanje atributa

- ✚ Postavljaju se atributi za koje je zadužen vlasnik
- ✚ Npr. zaštita

✚ Preimenovanje

- ✚ Menja se tekuće ime datoteke



Direktorijum (adresar, folder)

- ✦ **Direktorijum** je datoteka čiji je vlasnik OS i koji služi za organizovanje datoteka u logičke celine
- ✦ Sadrži informacije o datotekama
 - ✦ Atribute
 - ✦ Lokacija
 - ✦ Vlasništvo
 - ✦ Ostali atributi
- ✦ Obezbeđuje preslikavanje između imena datoteke koje je poznato korisnicima i aplikacijama i same datoteke koja je smeštena na nekom perifernom uređaju (disku, CD, DVD, USB Flash, ...)
- ✦ Direktorijumi mogu biti različito organizovani
 - ✦ U 1 nivou
 - ✦ U 2 nivoa
 - ✦ U obliku stabla
 - ✦ U obliku grafa

Informacije koje sadrži direktorijum

- ✿ Osnovne informacije
 - ✦ **Ime datoteke** – jedinstveno ime u okviru direktorijuma, izabrano od strane kreatora (korisnika, programa)
 - ✦ **Tip datoteke** – tekstualna, binarna, izvršna, itd.
 - ✦ **Organizacija datoteke** – u sistemima koji podržavaju različite organizacije
- ✿ Adresne informacije
 - ✦ **Volumen** - označava uređaj na kome je smeštena datoteka
 - ✦ **Početna adresa** – početna adresa datoteke na sekundarnoj memoriji (npr. cilindar, staza, broj bloka na disku)
 - ✦ **Veličina datoteke** – trenutna veličina datoteke u bajtovima ili blokovima
 - ✦ **Maksimalna veličina datoteke**
- ✿ Informacija o kontroli pristupa
 - ✦ **Vlasnik datoteke** – Korisnik koji ima pravo upravljanja datotekom; dodeljuje/oduzima prava ostalim korisnicima
 - ✦ **Informacije o pristupu** – najjednostavnije korisničko ime i lozinka svakog autorizovanog korisnika datoteke
 - ✦ **Dozvoljene akcije** – čitanje, upis, izvršavanje, menjanje, itd.



Informacije koje sadrži direktorijum (nastavak)

✚ Informacije o korišćenju

- ✚ Datum kreiranja
- ✚ Identitet kreatora datoteke (obično, ali ne i obavezno tekući vlasnik)
- ✚ Datum poslednjeg čitanja datoteke
- ✚ Identitet poslednjeg čitaoca datoteke
- ✚ Datum poslednjeg modifikovanja datoteke (upis, brisanje, ažuriranje)
- ✚ Identitet korisnika koji je izvršio poslednju modifikaciju datoteke
- ✚ Datum poslednjeg backup-a datoteke
- ✚ Tekuće korišćenje – informacije o tekućim aktivnostima nad datotekom, procesima koji su otvorili tu datoteku, da li je datoteka modifikovana u memoriji, ali još ne i na disku, itd.



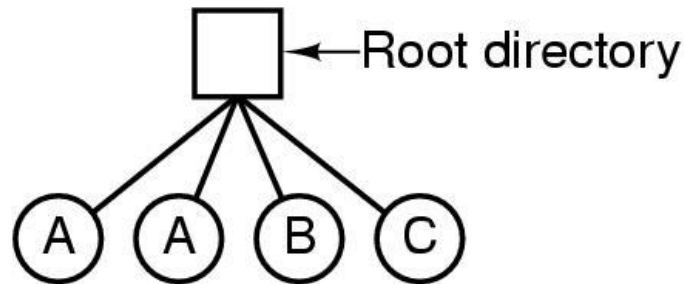
Operacije nad direktorijumima

Tipovi operacija nad direktorijumima

- Traženje
- Kreiranje datoteke
- Brisanje datoteke
- Listanje sadržaja direktorijuma
- Ažuriranje direktorijuma

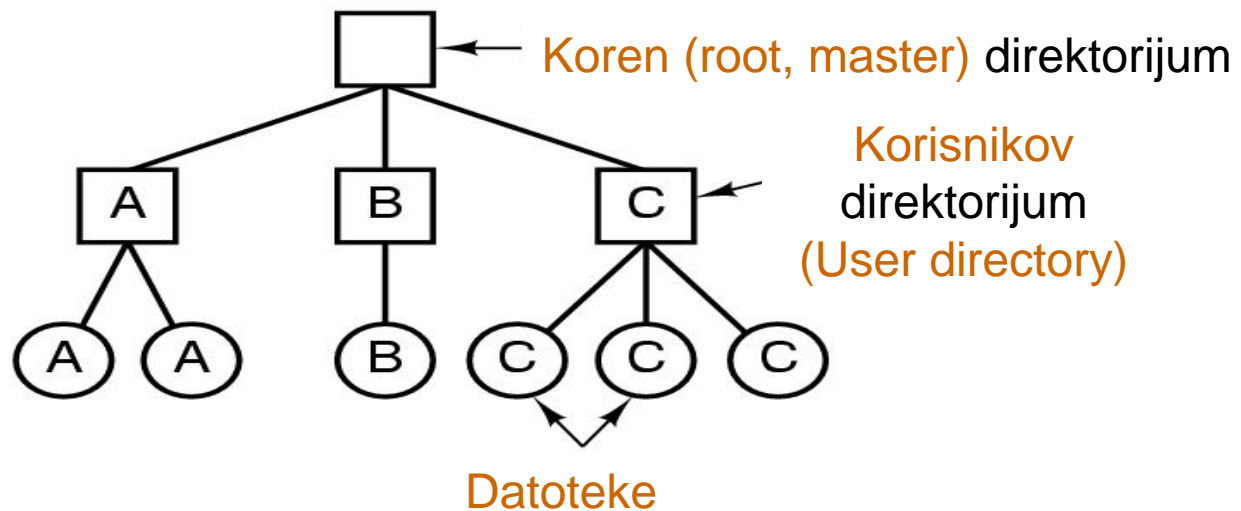
Direktorijum u 1 nivou

- ✿ Lista ulaza (*entry*), po jedan za svaku datoteku
- ✿ Sekvencijalna datoteka gde se ime datoteke koristi kao ključ
- ✿ Ne obezbeđuje pomoć u organizovanju datoteka
- ✿ Korisnik mora voditi računa da sve datoteke imaju različita imena



Direktorijum u 2 nivoa

- Po jedan direktorijum za svakog korisnika plus glavni (master) direktorijum
- Master direktorijum sadrži za svakog korisnika
 - adresu i informacije za kontrolu pristupa
- Svaki korisnički direktorijum je prosta lista datoteka tog korisnika
- Još uvek ne obezbeđuje pomoć u organizovanju datoteka



Direktorijumi strukture stabla ili hijerarhijski direktorijumi

- ✿ *Master direktorijum (koren)* sa korisničkim direktorijumima ispod njega
- ✿ Svaki korisnički direktorijum može imati kao ulaze poddirektorijume i datoteke
- ✿ Datoteke se mogu locirati praćenjem puta od korena direktorijuma, preko poddirektorijuma do datoteke
 - ✦ To je *ime puta (path name) datoteke*
- ✿ Može postojati više datoteka sa istim imenom sve dok oni imaju različita imena puta
- ✿ Tekući direktorijum je *radni direktorijum (working)*
- ✿ Datoteke se referenciraju relativno u odnosu na radni direktorijum
 - ✦ To je *relativno ime puta*

Ime puta



Ime puta



apsolutno

- UNIX `/usr/ast/mailbox`
- Windows `\usr\ast\mailbox`
- MULTICS `>usr>ast>mailbox`



relativno

- UNIX `ast/mailbox`
- Windows `ast\mailbox`
- MULTICS `>usr>ast>mailbox`



Apsolutno ime puta uvek počinje od korena



Relativno ime puta počinje od *radnog (tekućeg)* direktorijuma

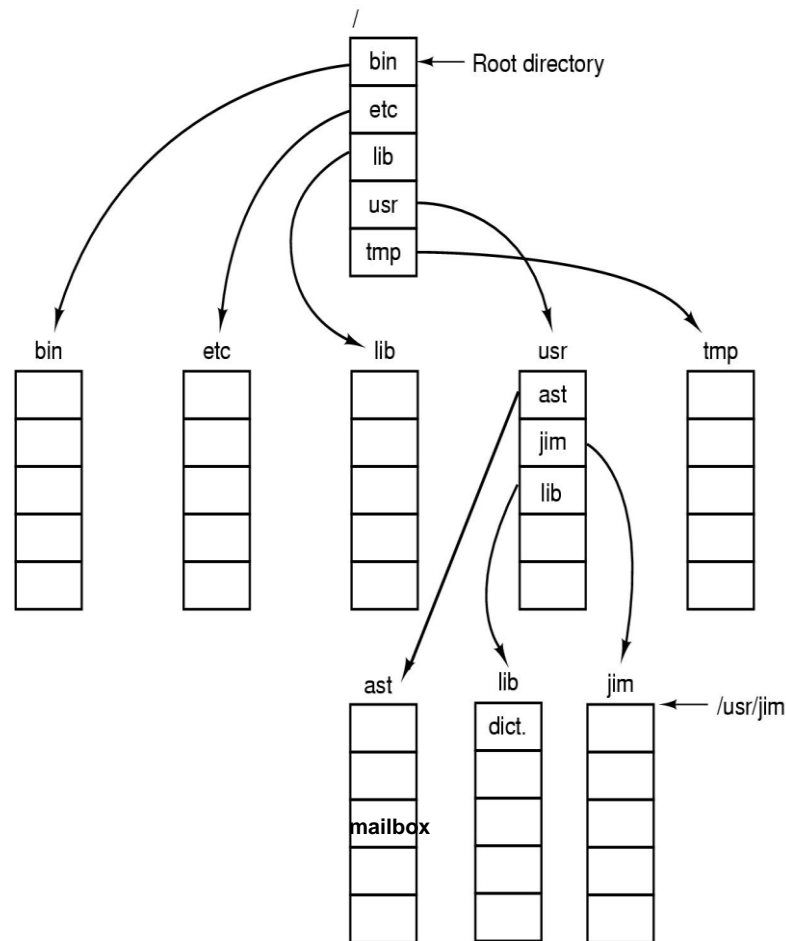


Svaki proces ima sopstveni radni direktorijum



U svakom direktorijumu postoje dva specijalna ulaza

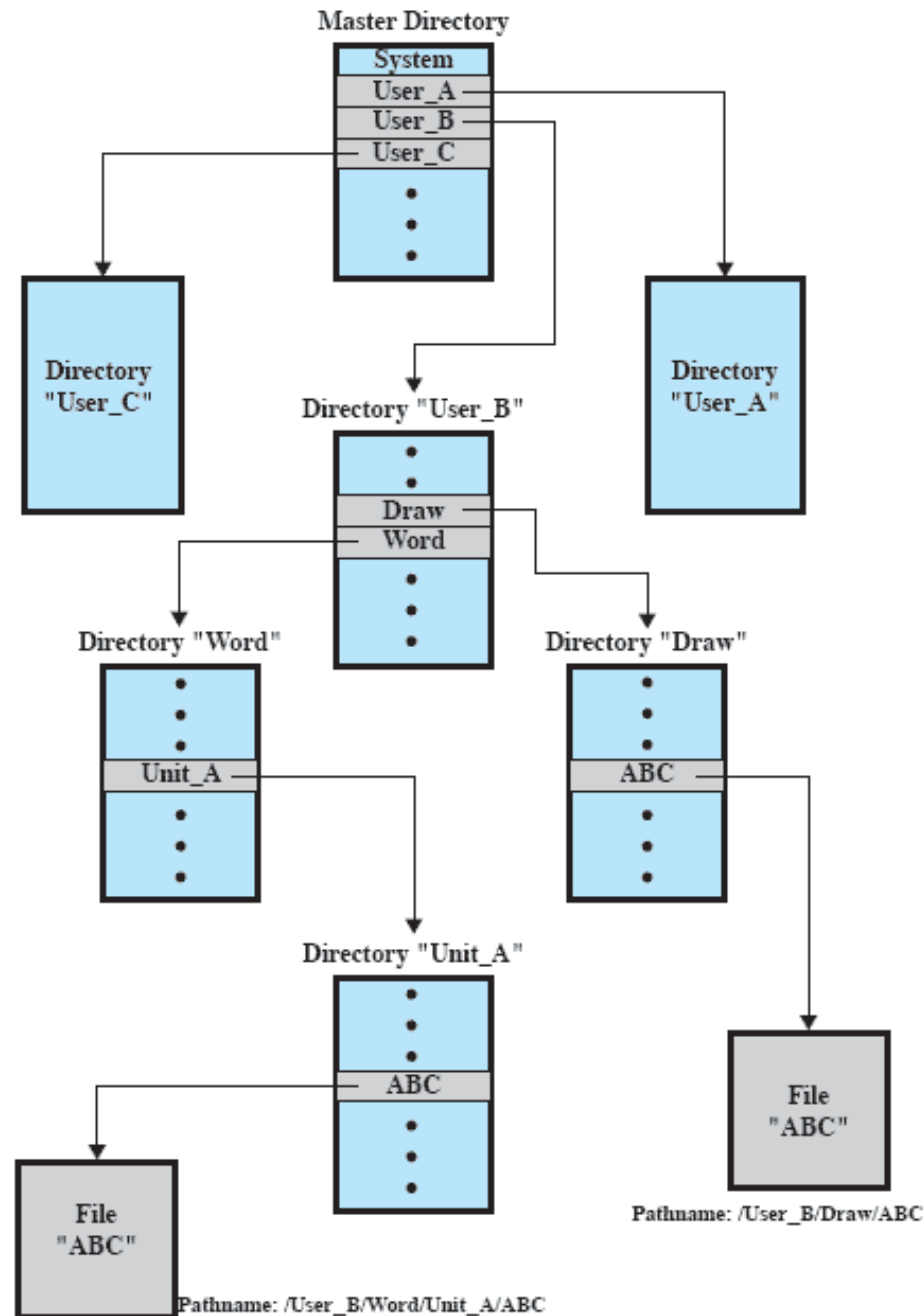
- . tekući direktorijum
- .. njegov roditeljski direktorijum



Stablo direktorijuma u UNIX-u



Primer direktorijuma strukture stabla



Operacije direktorijuma (1)

1. Kreiranje (*Create*)
 2. Brisanje (*Delete*)
 3. Otvaranje direktorijuma (*Opendir*)
 4. Zatvaranje direktorijuma (*Closedir*)
 5. Čitanje direktorijuma (*Readdir*)
 6. Preimenovanje (*Rename*)
 6. Povezivanje (*Link*)
 7. Razvezivanje (*Unlink*)
- Korisnik može direktno pozivati ove funkcije – *komande*
 - Operacije se mogu pozivati iz programa - *sistemski pozivi*
 - Postoje i drugi sistemski pozivi za rad sa direktorijumima (*Praktikum*)



Operacije nad direktorijumima (2)

- ✚ Kreiranje (**Create**)
 - ✚ Kreira direktorijum
 - ✚ direktorijum ima samo . i ..
 - ✚ Unosi se u sistemske tablice
- ✚ Brisanje (**Delete**)
 - ✚ Briše direktorijum
 - ✚ Može se brisati samo prazan direktorijum
 - ✚ Oslobađa se prostor koji direktorijum drži
 - ✚ Uklanja se iz sistemskih tablica
- ✚ Otvaranje (**Opendir**)
 - ✚ Pre upotrebe proces mora otvoriti direktorijum
 - ✚ Omogućava se sistemu da pristupa atributima direktorijuma
 - ✚ OS ograničava broj otvorenih direktorijuma
- ✚ Zatvaranje (**Closedir**)
 - ✚ direktorijum se zatvara kada više nije potreban
 - ✚ Oslobađa se prostor u internim tablicama
- ✚ Čitanje (**Readdir**)
 - ✚ Čita se jedan ulaz iz direktorijuma
- ✚ Preimenovanje (**Rename**)
 - ✚ Menja se tekuće ime direktorijuma
- ✚ Povezivanje (**Link**)
 - ✚ Omogućava da se isti datoteka javlja u više direktorijuma
- ✚ Razvezivanje (**Unlink**)
 - ✚ Uklanja se link ulaz direktorijuma



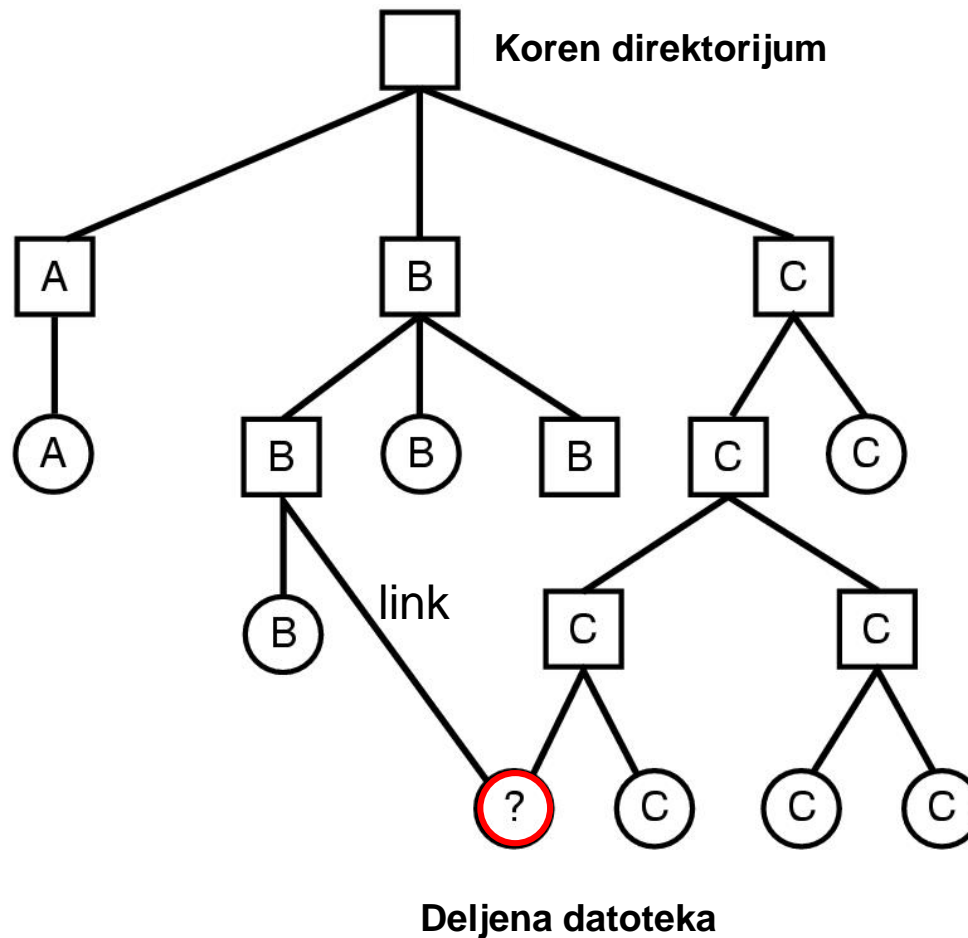
Deljenje datoteka

File Sharing

- U višekorisničkom sistemu, dozvoljeno je da se datoteke dele među korisnicima
 - Da se zajednički koriste
- Dva su nova problema
 - Prava pristupa (*Access rights*)
 - Upravljanje simultanim pristupom



Primer file system-a koji sadrži deljenu datoteku



Implementacija deljenih datoteka

- Deljena datoteka se može implementirati na više načina. Niže su prikazane dve metode:

Metoda 1 (*hard link*)

- Podaci o blokovima koji su dodeljeni datoteci se ne nalaze u direktorijumu, već u posebnoj strukturi uz datoteku, a u direktorijumu je pokazivač na tu strukturu
- Za deljenu datoteku se u direktorijumu korisnika kome je dozvoljen pristup formira klasičan ulaz za datoteku, a u originalnom ulazu se inkrementira *brojač linkova*
 - ❏ moguće kod sistema sa i-čvorom (npr. Unix)

Metoda 2 (*simboličko linkovanje*)

- U direktorijumu korisnika kome je dozvoljen pristup se kreira nova datoteka tipa LINK
 - ❏ Nova datoteka sadrži samo put do datoteke koja je linkovana
 - ❏ Kada se čita takva datoteka, OS prepoznaje da je tipa LINK, uzima put do datoteke i čita originalnu datoteku



Prava pristupa (1)

- ✿ Nije dozvoljeno znati da datoteka postoji (*None*)
 - ✦ Korisnik ne sme znati da datoteka postoji
 - ✦ Korisniku nije dopušteno da čita direktorijum koji sadrži datoteku
- ✿ Dozvoljeno je znati da datoteka postoji, ali je za pristup potrebna dozvola (*Knowledge*)
 - ✦ Korisnik jedino može utvrditi da datoteka postoji i ko je vlasnik
 - ✦ Mora od vlasnika tražiti dozvolu pristupa

Prava pristupa (2)

✿ Izvršenje (*Execution*)

- ✚ Korisnik može loadovati i izvršavati program, ali ga ne može kopirati

✿ Čitanje (*Reading*)

- ✚ Korisnik može čitati datoteku u bilo koje svrhe, uključujući kopiranje i izvršenje

✿ Dodavanje (*Appending*)

- ✚ Korisnik može dodavati podatke datoteci, obično na kraj datoteke, ali ne može modifikovati ili brisati sadržaj datoteke

Prava pristupa (3)

✿ Ažuriranje (*Updating*)

- ✦ Korisnik može modifikovati, brisati i dodavati podatke datoteci. Ovo uključuje kreiranje datoteke, ponovni upis i brisanje svih ili dela podataka

✿ Izmena zaštite (*Changing protection*)

- ✦ Korisnik može promeniti pristupna prava koja je dodelio drugim korisnicima

✿ Brisanje (*Deletion*)

- ✦ Korisnik može obrisati datoteku

Prava pristupa (4)

✚ Vlasnici

- ✚ Imaju sva prethodno navedena prava pristupa
- ✚ Mogu prenositi prava drugim korisnicima koristeći sledeće klase korisnika
 - Specifični korisnik (*specific user*)
 - Pojedini korisnici koji imaju svoj **userID**
 - Grupe korisnika (*user groups*)
 - Skup korisnika koji imaju zajednički identifikator **groupID**
 - Svi (*public*)

Istovremeni pristup

- ✿ Ako je dozvoljeno dodavanje ili ažuriranje datoteke većem broju korisnika tada OS mora obezbediti nesmetani simultani pristup
 - ✦ Prosto rešenje: dozvoliti korisniku da zaključa celu datoteku kada je ažurira
 - ✦ Finije rešenje: dozvoliti korisniku da zaključa slog koji ažurira
 - ✦ Svodi se na sinhronizacioni problem čitaoci-pisci
- ✿ Za deljene datoteke treba rešiti uzajamno isključivanje i samrtni zagrljaj

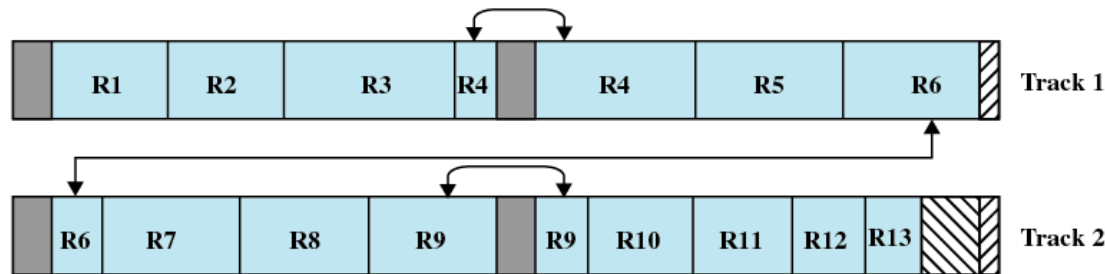
Blokiranje slogova

- ✿ Organizovanje slogova u blok radi U/I
- ✿ Tri metode blokiranja
 - ✚ Fiksno blokiranje – u blok je smešten celi broj slogova fiksne veličine
 - Interna fragmentacija
 - ✚ Blokiranje varijabilne dužine sa prebacivanjem – slogovi varijabilne dužine su smešteni u blok sa mogućim prebacivanjem ostatka poslednjeg sloga u bloku u sledeći blok
 - ✚ Blokiranje varijabilne dužine bez prebacivanja - slogovi su varijabilne dužine, ali nije moguće prebacivanje slogova između blokova

Blokiranje slogova - primer



Fixed Blocking



Variable Blocking: Spanned



Variable Blocking: Unspanned

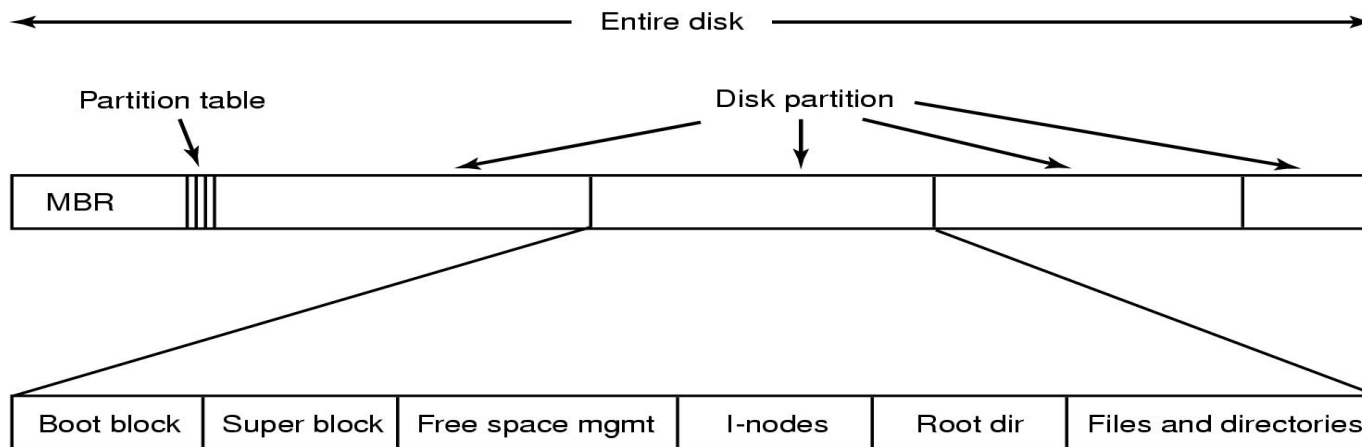
- Data
- Gaps due to hardware design
- Waste due to block fit to track size
- Waste due to record fit to block size
- Waste due to block size constraint from fixed record size



Strukture podataka file system-a

- File system na disku i u memoriji održava određene strukture podataka neophodne za podršku upravljanju datotekama
- Na disku** se čuvaju informacije za inicijalno punjenje (bootovanje) OS-a, ukupan broj blokova, broj i lokacije slobodnih blokova, struktura direktorijuma, kao i same datoteke i direktorijumi
- U memoriji** se čuvaju informacije neophodne za upravljanje file system-om i za poboljšanje performansi (keširanje)

Struktura diska

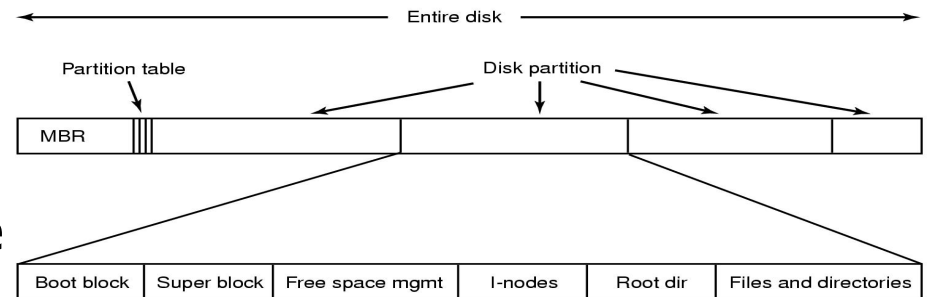


Tanenbaum, 2014

- ✚ File system se memoriše na disku
- ✚ Na slici je prikazana moguća struktura (*layout*) diska
 - ▣ svaki OS ima sopstveni *layout*
- ✚ Mnogi diskovi su podeljeni na 1 ili više particija
- ✚ Sektor 0 diska se zove *MBR (Master Boot Record)*
 - ▣ koristi se za *boot-ovanje* sistema (inicijalno load-ovanje operativnog sistema)
- ✚ Na kraju MBR-a se nalazi *Tabela particija (Partition Table)*
 - ▣ čuva adresu početka i kraja svake particije
- ✚ Jedna particija je označena kao aktivna – sadrži OS

Struktura particije

- ✚ Boot block (Unix), Partition boot sector (NTFS)
 - ▣ Sadrži punilac (*bootstrep loader*)
- ✚ Super blok (Unix), Master File Table (NTFS)
 - ▣ Sadrži sve ključne parametre o file system-u
 - Broj blokova u particiji
 - Veličina bloka
 - Broj slobodnih blokova
 - Pokazivač na listu slobodnih blokova (bitmapa slobodnih blokova)
 - Slobodne FCB-e
 - Pokazivač na listu FCB
 - ▣ Učitava se u memoriju kada se računar boot-uje ili kada se prvi put pristupa file system-u
- ✚ Informacija o slobodnom prostoru za smeštanje podataka
 - ▣ Bit mapa ili lista pokazivača (adresa) na slobodne blokove
- ✚ i-čvorovi (*i-node*) (Unix), FCB
 - ▣ Po jedan za svaku datoteku
 - ▣ Sadrži attribute datoteke
- ✚ Koren direktorijum (root)
- ✚ Sačuvane datoteke i direktorijumi





Strukture podataka file system-a u glavnoj memoriji

✿ Tabela particija u memoriji

- ✦ Sadrži informacije o svim mount-ovanim particijama

✿ Struktura direktorijuma u memoriji

- ✦ Sadrži informacije o direktorijumima koji su skoro korišćeni
- ✦ Pokazivač na Tabelu particija gde je direktorijum mount-ovan

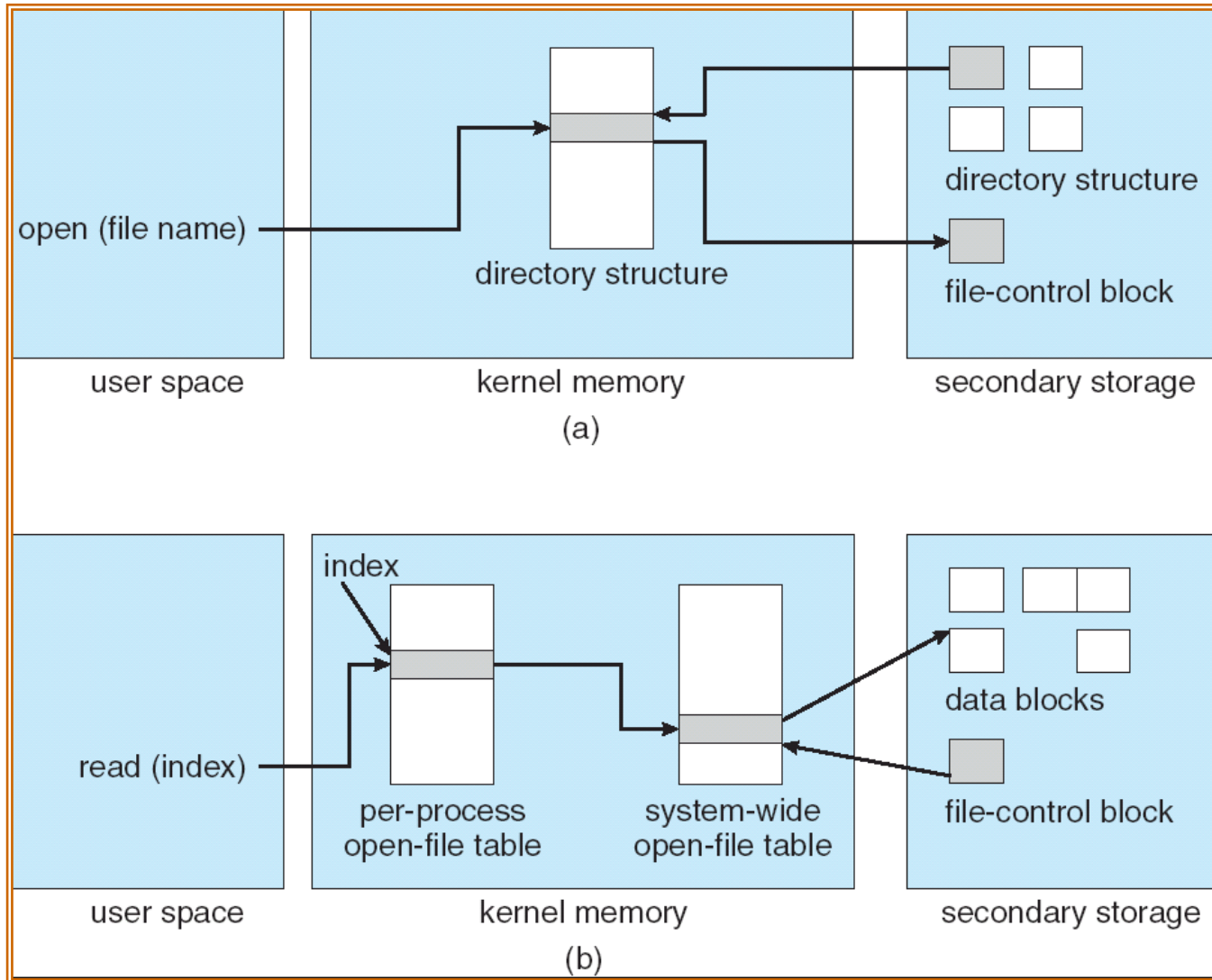
✿ Tabela otvorenih datoteka sistema (1 za ceo sistem)

- ✦ Sadrži kopije FCB-ova svih otvorenih datoteka, kao i druge sistemske informacije

✿ Tabela otvorenih datoteka procesa (po 1 za svaki proces)

- ✦ Sadrži pokazivač na odgovarajući ulaz Tabela otvorenih datoteka sistema, kao i druge informacije

Strukture podataka file system-a u glavnoj memoriji (2)



Silberschatz, 2012

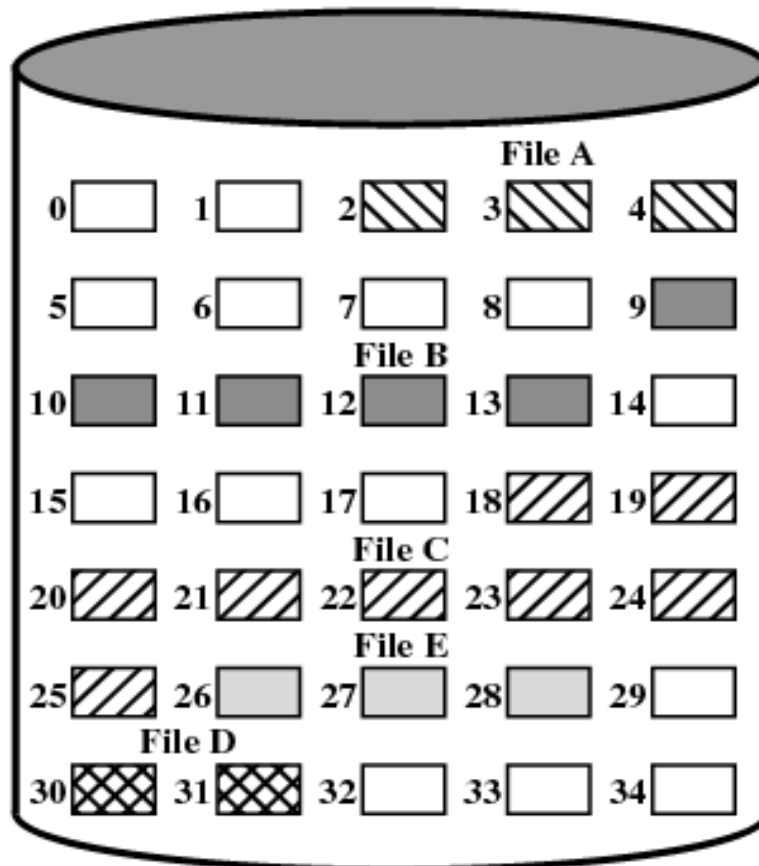
Upravljanje sekundarnom memorijom

- ✱ Datoteke se sastoje od kolekcije blokova i file system vodi evidenciju o datotekama na sekundarnoj memoriji
 - ✱ Alokacija (dodeljivanje) datoteke
 - ✱ Upravljanje slobodnim prostorom na sekundarnoj memoriji
- ✱ **Dodeljivanje datoteke** – dodela blokova datoteci na osnovu određene strategije i evidencija dodeljenih blokova
 - ✱ **Dodeljivanje unapred** zahteva da se maksimalna veličina datoteke specificira u vreme kreiranja datoteke, dok **dinamičko dodeljivanje** dodeljuje datoteci blokove po potrebi
 - ✱ Datoteci se može dodeliti jedinstvena **kontinualna** grupa blokova, **više grupa kontinualnih** blokova različite veličine, ili **pojedinačni** blokovi. Ukoliko se dodeljivanje datoteke obavlja u varijabilnim grupama kontinualnih blokova, primenjuju se strategije prvo, najbolje i sledeće poklapanje (*first, best, next fit*)
- ✱ Za dodeljivanje datoteke se koriste metode:
 - ✱ **Kontinualna dodela**
 - ✱ **Nekontinualna dodela**
 - Ulančavanje blokova
 - Indeksiranje blokova

Kontinualna dodela

- ✿ Najprostija šema dodele (npr. IBM VM/CMS OS)
- ✿ Svakom fajlu se dodeljuje potreban broj susednih blokova na disku
- ✿ Prednosti:
 - ▣ Jednostavna evidencija dodeljenog prostora
 - U adresaru se za svaku datoteku pamti adresa prvog dodeljenog bloka i broj dodeljenih blokova (dužina)
 - ▣ Odlične performanse čitanja
 - Vreme traženja je minimalno
 - Potrebno je samo postaviti glavu diska na prvi blok
 - Nema pomeranja glave kod čitanja sledećeg bloka
 - Ceo fajl se može pročitati jednom operacijom
 - ▣ Jednostavan sekvencijalni i direktni pristup
 - Sekvencijalni – iz adresdirektorijum čita adresa prvog bloka, a zatim se nakon čitanja prvog bloka, adresa uvećava za 1, itd.
 - Direktni pristup – blok i datoteke koji počinje od bloka b se nalazi na adresi $b+i$
- ✿ Nedostaci
 - ▣ Eksterna fragmentacija
 - ▣ Datoteke ne mogu da rastu
 - ▣ Povremeno je potrebno vršiti kompakciju
- ✿ Primena: CD-ROM

Primer kontinualne dodele (1)

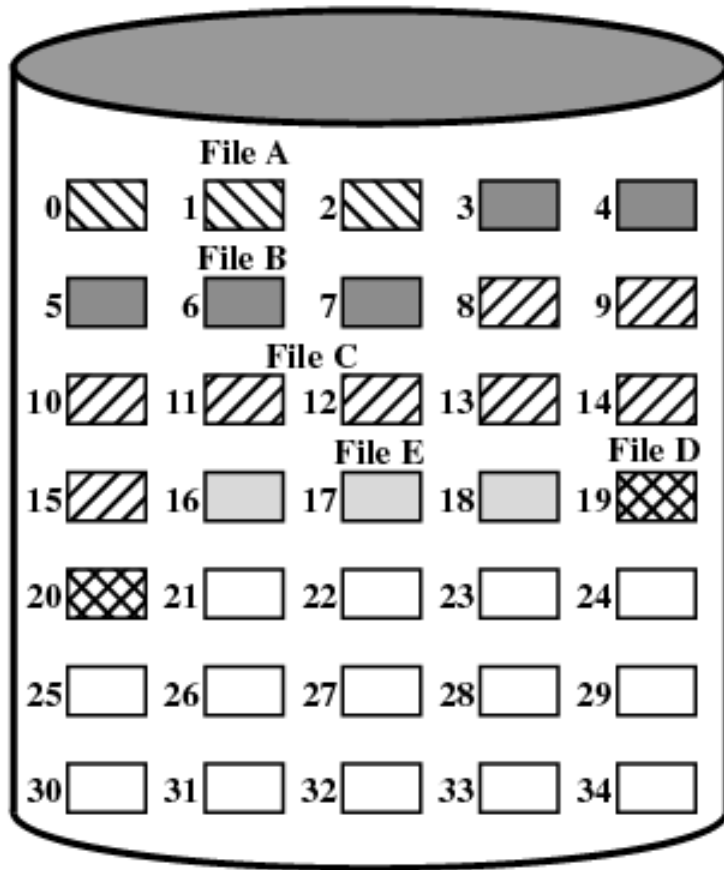


File Allocation Table

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

Primer kontinualne dodele (2)

- nakon kompakcije



File Allocation Table

File Name	Start Block	Length
File A	0	3
File B	3	5
File C	8	8
File D	19	2
File E	16	3

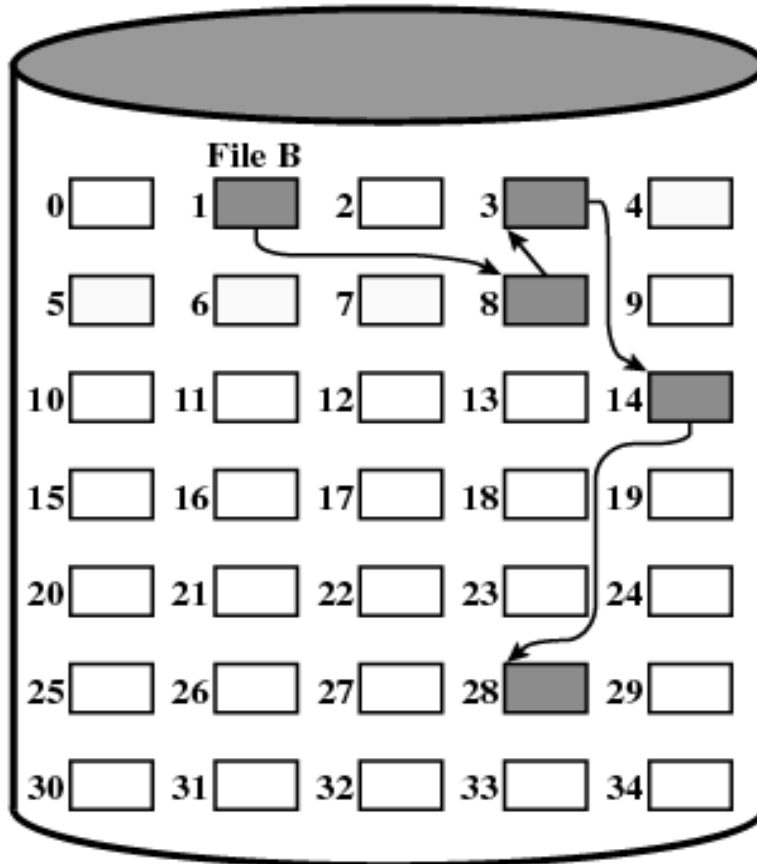
Dodela ulančavanjem (1)

- ❖ Blokovi dodeljeni datoteci su povezani u lančanu listu
- ❖ Prva reč svakog bloka se koristi za čuvanje adrese na sledeći blok datoteke
- ❖ Ostali deo bloka su podaci
- ❖ U direktorijumu se čuva samo adresa prvog bloka datoteke
- ❖ Svojstva:
 - ❑ Nema eksterne fragmentacije
 - ❑ Brz sekvencijalni pristup
 - ❑ Neefikasan random pristup: svodi se na sekvencijalni pristup, odnosno na obilazak lančane liste blokova startujući od prvog bloka

blok



Primer dodele ulančavanjem (2)

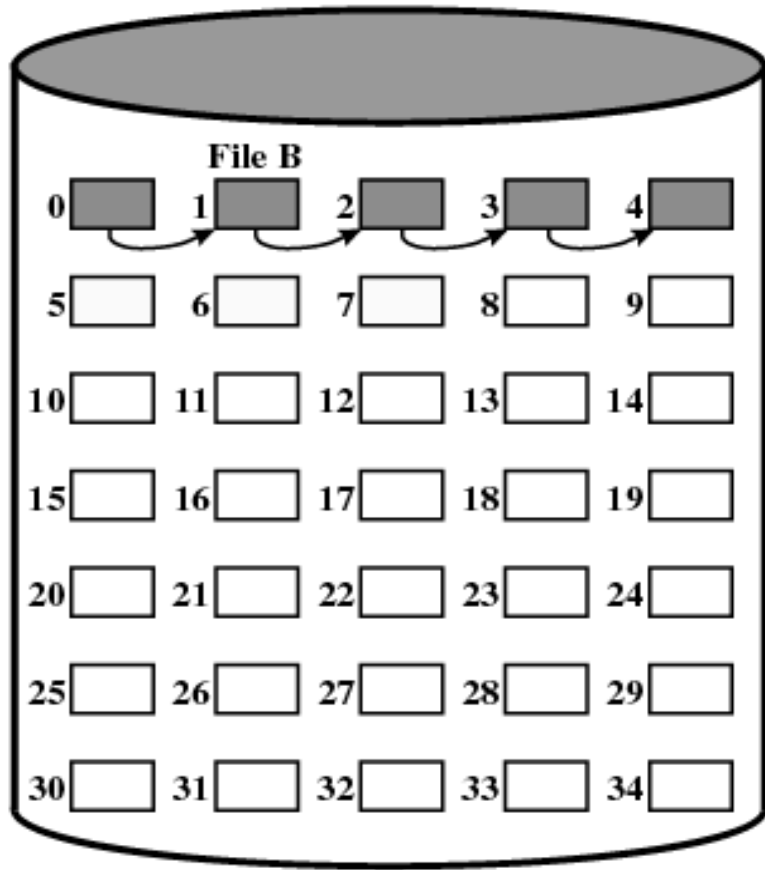


File Allocation Table

File Name	Start Block	Length
...
File B	1	5
...

Primer dodele ulančavanjem (2)

- nakon integrisanja



File Allocation Table

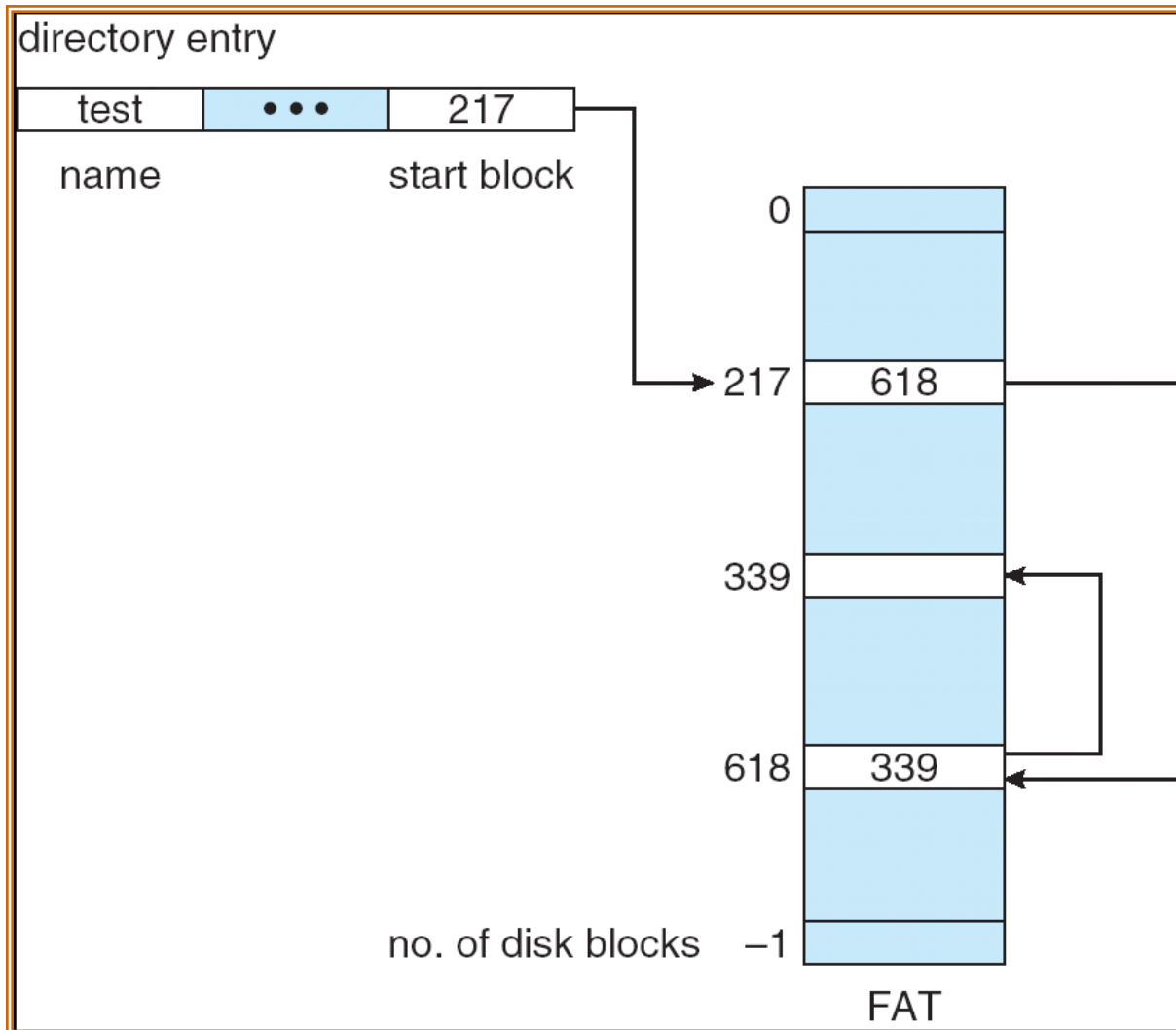
File Name	Start Block	Length
...
File B	0	5
...

Dodela ulančavanjem korišćenjem tabele u memoriji

Fizički blok	FAT	
0		
1		
2	10	
3	11	
4	7	← Fajl A počinje ovde
5		
6	3	← Fajl B počinje ovde
7	2	
8		
9		
10	12	
11	14	
12	-1	
13		
14	-1	
15		← Blok koji se ne koristi

- Uvodi se **Tabela dodele datoteke (FAT - File Allocation Table)**
- U FAT-u se održava lančana lista dodele (primer: MS-DOS, Windows 95/98/Me, OS/2)
- U direktorijumu se čuva samo adresa prvog bloka datoteke
- Svojstva:
 - Ceo blok se koristi za podatke
 - Brži random pristup
- Nedostatak:
 - Cela tabela mora biti u memoriji
 - Za disk od 20GB sa veličinom bloka 1KB, FAT ima 20 miliona ulaza
 - Svaki ulaz minimalno 3B ili 4B
 - Za FAT neophodno 60 do 80 MB

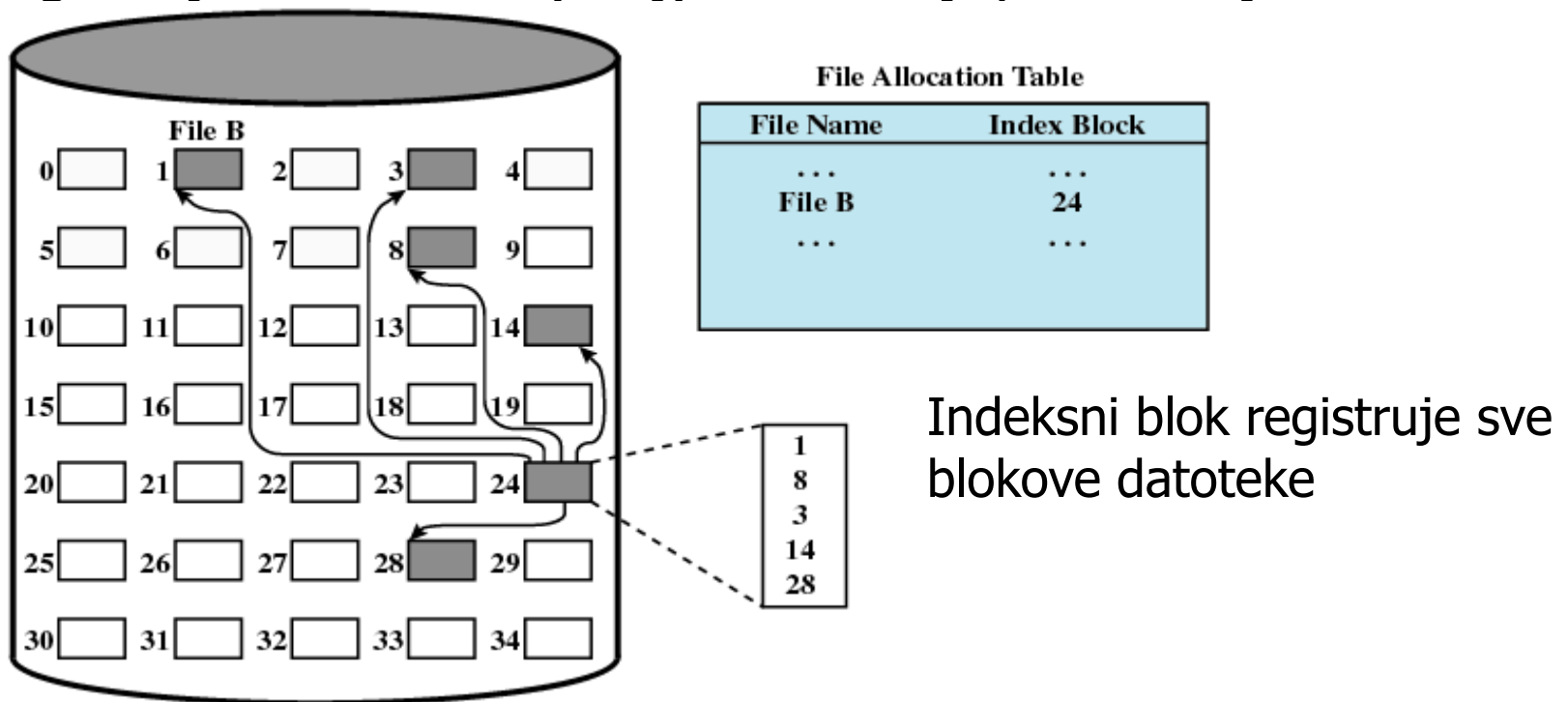
FAT primer



Silberschatz, 2012

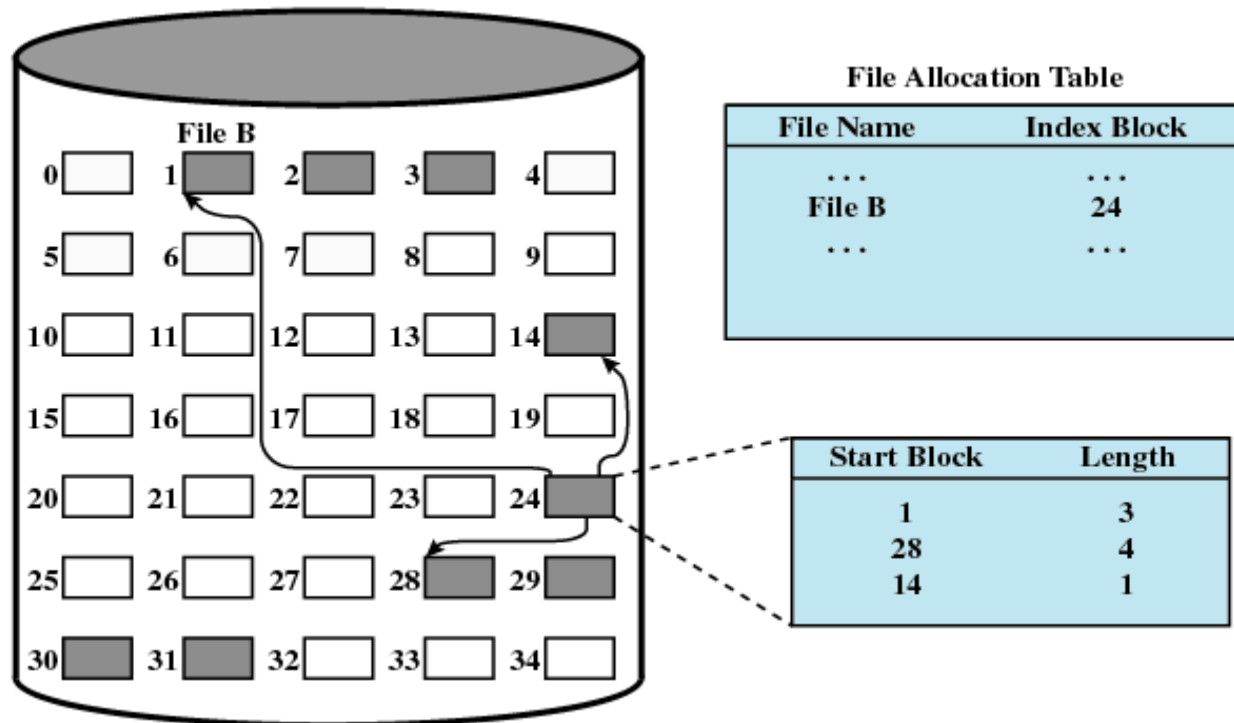
Dodela indeksiranjem

- Svakoj datoteci se dodeljuje **indeksni blok** (*index block*)
- U *indeksnom bloku* su smeštene adrese blokova (grupa blokova) datoteke
- U okviru ulaza direktorijuma (*File Allocation Table*) registruje se adresa (broj) bloka koji predstavlja indeksni



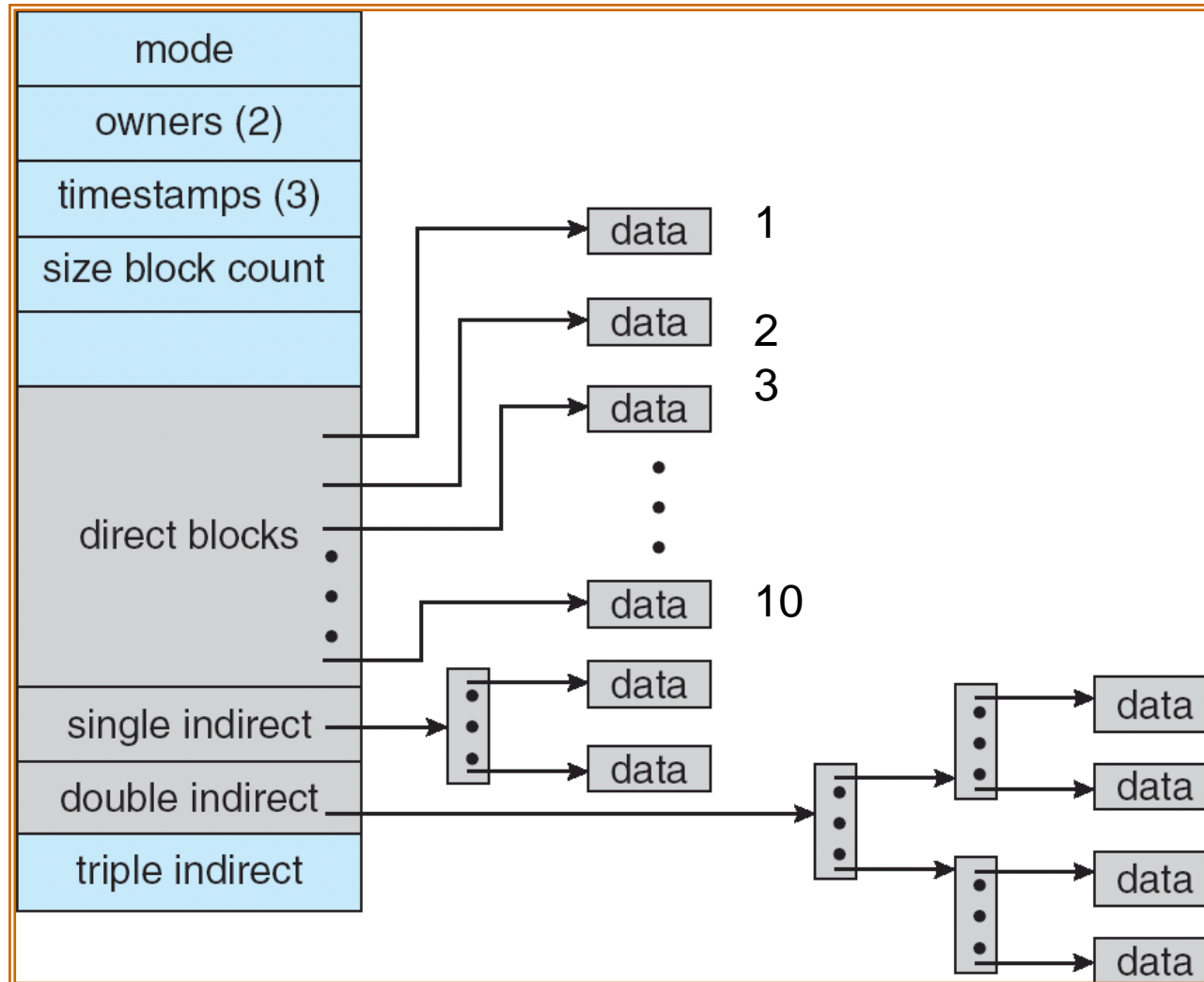
Varijanta dodele indeksiranjem

- ✱ Datoteci se dodeljuju blokovi u grupama blokova različite veličine, pa se u indeksnom bloku registruju te varijabilne grupe blokova datoteke



Dodela blokova u UNIX

i-čvor
(*i-node*)





Upravljanje slobodnim prostorom na disku

- ✿ Današnji OS koriste blok (klaster) kao jedinicu zauzimanja/oslobađanja
- ✿ Ostaje pitanje kolika je veličina bloka poželjna?
 - ✦ Umnožak veličine sektora
 - ✦ 2^n KB
- ✿ Primenjene tehnike
 - ✦ Bit tabela (bit vektor)
 - ✦ Ulančavanje slobodnih blokova (delova sastavljenih od grupe sukcesivnih blokova)
 - ✦ Indeksiranje
 - ✦ Lista slobodnih blokova



Bit tabela (bit vektor)

Bit vektor ili bit tabela

- ❖ Svaki blok na disku je predstavljen 1 bitom
- ❖ Traženje slobodnog prostora od n blokova se svodi na nalaženje sekvence od n nula u bit vektoru
- ❖ Bit vektor se čuva u slobodnim blokovima na disku
- ❖ U memoriju se prenosi blok po blok

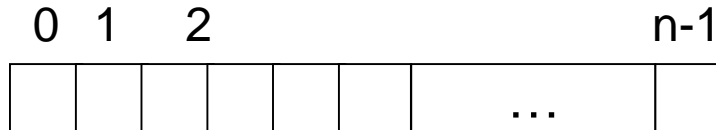
Bit tabela zahteva ekstra prostor - primer:

veličina bloka = 512 B = 2^9 B

veličina diska = 16 GB = 2^{34} B

$$n = 2^{34}/2^9 = 2^{25} \text{ bitova} = 4 \text{ MB (veličina bit tabele)}$$

- Mnogi file system-i održavaju pomoćnu strukturu podataka koja na osnovu logičke podele bit vektora na podopsege, u toj strukturi registruje za svaki podopseg broj slobodnih blokova i maksimalni broj kontinualnih slobodnih blokova u podopsegu.



$$\text{bit}[i] = \begin{cases} 0 \Rightarrow \text{blok}[i] \text{ slobodan} \\ 1 \Rightarrow \text{blok}[i] \text{ zauzet} \end{cases}$$

Upravljanje datotekama

Operativni sistemi

Ulančavanje slobodnog prostora

❁ Lančana lista (slobodna lista)

- ❁ Ulančavanje svih slobodnih disk blokova ili grupa blokova
 - Pokazivač na prvi blok (grupu blokova) ove liste se drži na specijalnoj lokaciji na disku i kešira u memoriji
- ❁ Ovaj metod je pogodan za sve metode alokacije datoteka
- ❁ Nedostaci ove metode
 - Kontinualni prostor željene veličine se ne može lako naći,
 - Ukoliko se alociranje vrši u pojedinačnim blokovima neophodan je višestruki pristup disku da bi se alocirao odgovarajući broj blokova
 - Datoteci se dodeljuje prvi slobodan blok (grupa blokova) iz slobodne liste
 - Ukoliko je alociranje u grupama kontinualnih blokova, prostor diska bi bio fragmentiran, sa grupama blokova uglavnom veličine jednog bloka

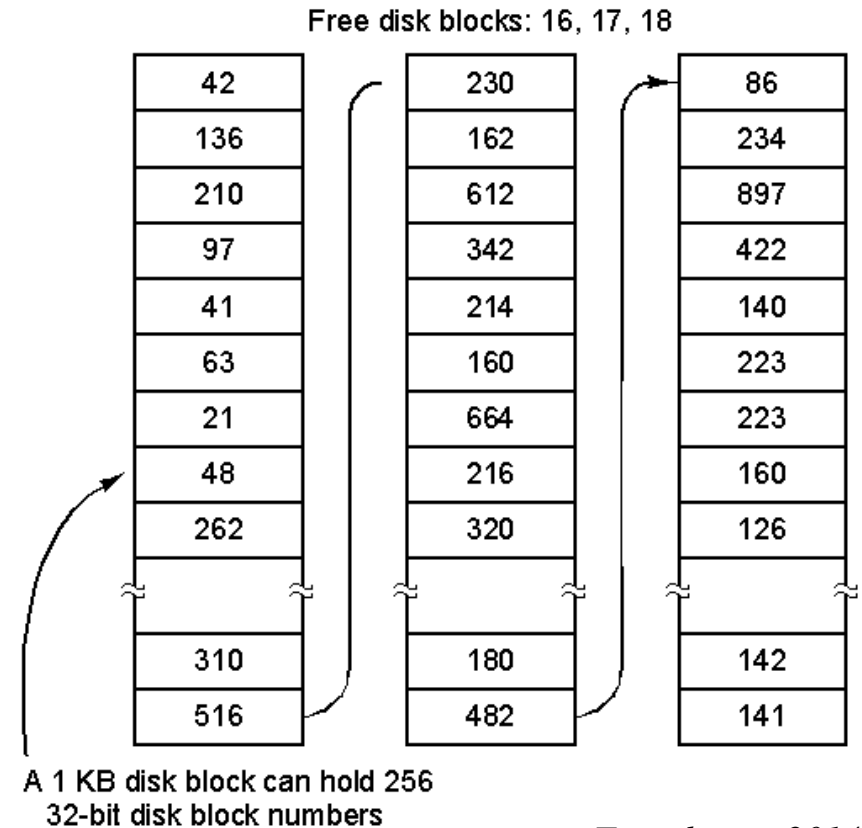
Indeksiranje i lista slobodnih blokova

Indeksiranje

- Slobodni prostor se tretira kao datoteka i slobodni blokovi (grupe blokova) se indeksiraju

Lista slobodnih blokova

- Adrese (brojevi) slobodnih blokova (veličine 3-4 B) se registruju u određenim blokovima na disku koji su ulančani
- Deo ove liste (jedan ili dva bloka) se drži u glavnoj memoriji i po potrebi snima na disk (kada se blok napuni) ili učitava sa diska (kada se slobodni blokovi dodele) po principu magacina ili FIFO reda



(a)

Tanenbaum, 2014

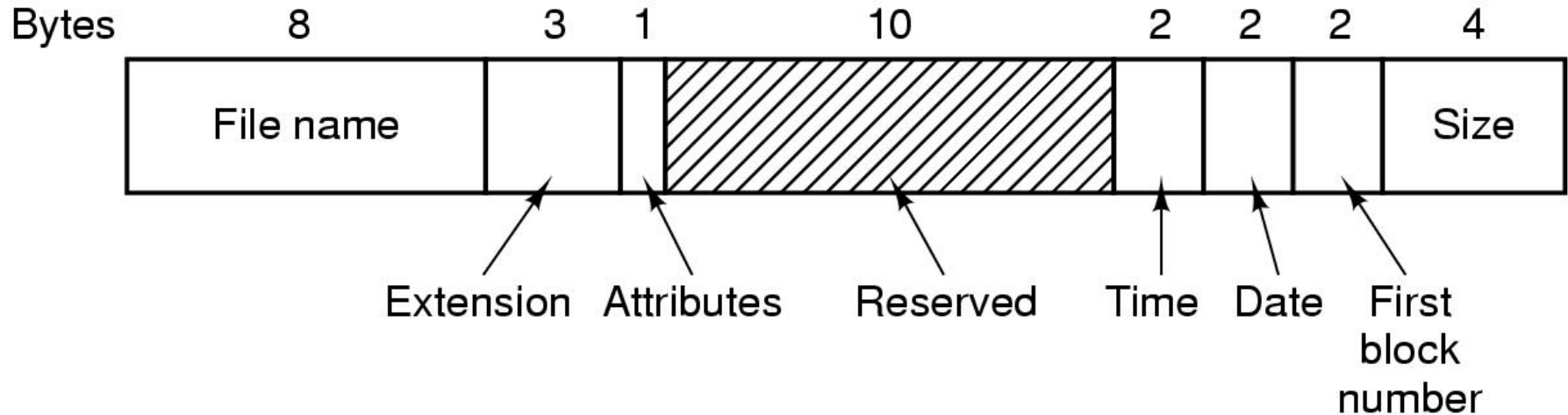


Primeri datotečnih sistema

- ✚ Windows (FAT i NTFS)
- ✚ Unix
- ✚ Linux

MS-DOS file system (1)

Stavka direktorijuma



- ❁ FAT file system dolazi u 3 verzije za MS-DOS
 - ❏ FAT-12: adresa bloka na disku 12 bitova
 - ❏ FAT-16: adresa bloka na disku 16 bitova
 - ❏ FAT-32: adresa dis bloka na disku ka 32 bitova
- ❁ Veličina bloka na disku je umnožak 512 B (može biti različita u svakoj particiji)

Tanenbaum, 2014

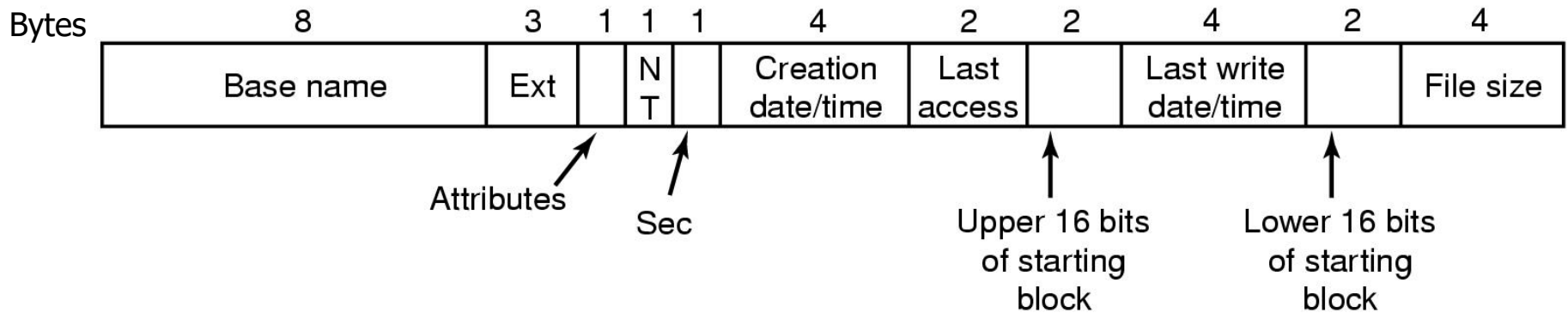
MS-DOS file system (2)

Block size	FAT-12	FAT-16	FAT-32
0.5 KB	2 MB		
1 KB	4 MB		
2 KB	8 MB	128 MB	
4 KB	16 MB	256 MB	1 TB
8 KB		512 MB	2 TB
16 KB		1024 MB	2 TB
32 KB		2048 MB	2 TB

- ✿ Maksimalne veličine particija za različite veličine blokova
- ✿ Prazne stavke označavaju zabranjene kombinacije



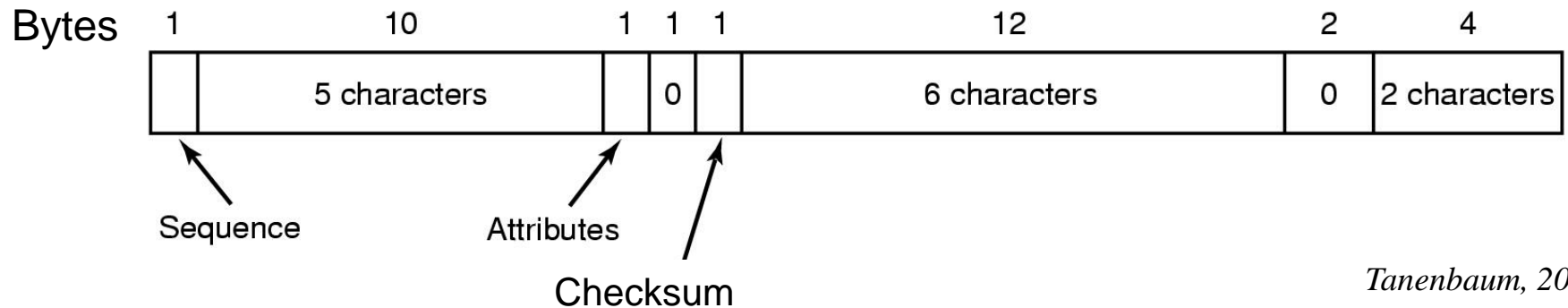
Windows 98 file system (1)



Tanenbaum, 2014

Proširen MS-DOS ulaz direktorijuma korišćen u Windows 98

Windows 98 file system (2)



Ulaz direktorijuma za duga imena u Windows 98

Windows 98 file system (3)

Bytes	68	d o g										A	0	C	K									0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
	3	o v e										A	0	C	K	t h e				l a				0	z y																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
	2	w n f o										A	0	C	K	x j u m p				0	s																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									
	1	T h e q										A	0	C	K	u i c k				b				0	r o																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
	T	H E Q U I ~ 1										A	N	T	S	Creation time			Last acc		Upp		Last write			Low		Size																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		

Tanenbaum, 2014

Primer kako se dugačko ime pamti u Windows 98

FAT

Format FAT diska

Master Boot Record	File Allocation Table #1	File Allocation Table #2	Root Directory	All Other Data ... The Rest of the Disk
--------------------------	--------------------------------	--------------------------------	-------------------	---

Sadržaj ulaza (*entry*) u FAT za različite verzije FAT

FAT12	FAT16	FAT32	Description
0x000	0x0000	0x?0000000	Free Cluster
0x001	0x0001	0x?0000001	Reserved Cluster
0x002 - 0xFE7	0x0002 - 0xFFEF	0x?0000002 - 0x?FFFFFFEF	Used cluster; value points to next cluster
0xFF0 - 0xFF6	0xFFFF0 - 0xFFFF6	0x?FFFFFFF0 - 0x?FFFFFFF6	Reserved values
0xFF7	0xFFFF7	0x?FFFFFFF7	Bad cluster
0xFF8 - 0xFFF	0xFFFF8 - 0xFFFFF	0x?FFFFFFF8 - 0x?FFFFFFF	Last cluster in file



Primer entry-a u direktorijumu FAT 16





NTFS Windows file system

- ✿ NTFS je file system razvijen za Windows NT
- ✿ Koristi se od Windows 2000 u svim kasnijim verzijama (Windows Vista, Windows 7)
- ✿ Ključne karakteristike NTFS-a
 - ✦ Ima sposobnost oporavka nakon pada sistema i greški na disku
 - ✦ Sigurnost
 - ✦ Podrška za veoma velike diskove i datoteke
 - ✦ Višestruki tokovi podataka (sadržaj datoteke se tretira kao tok bajtova, pa je moguće definisati višestruke tokove podataka nad istom datotekom)
 - ✦ Sposobnost generalnog indeksiranja



NTFS struktura volumena i datoteke

✧ Sektor

- ✦ Najmanja fizička jedinica memorisanja na disku (skoro uvek 512B)

✧ Klaster (*cluster*)

- ✦ Jedan ili više susednih sektora (1, 2, 4, 8, 16, ...), maksimalno 2^{16} B = 64 KB.
- ✦ Maksimalna veličina datoteke 2^{32} klastera što iznosi maksimalno 2^{48} B)

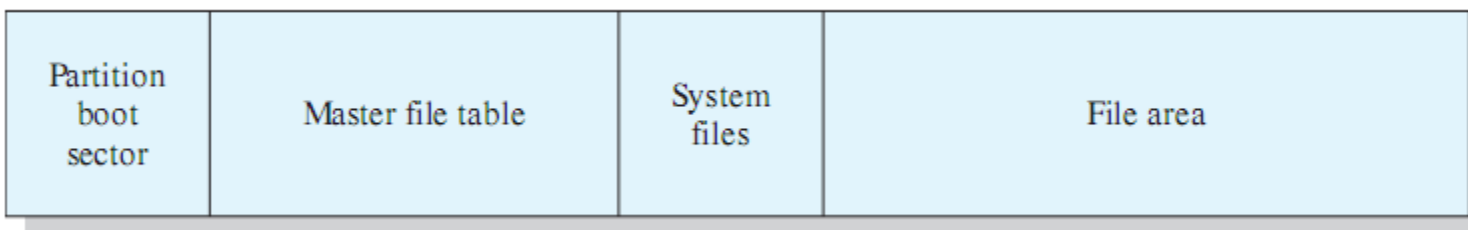
✧ Volumen

- ✦ Logički deo diska (particija diska)
- ✦ Maksimalna veličina 2^{64} B



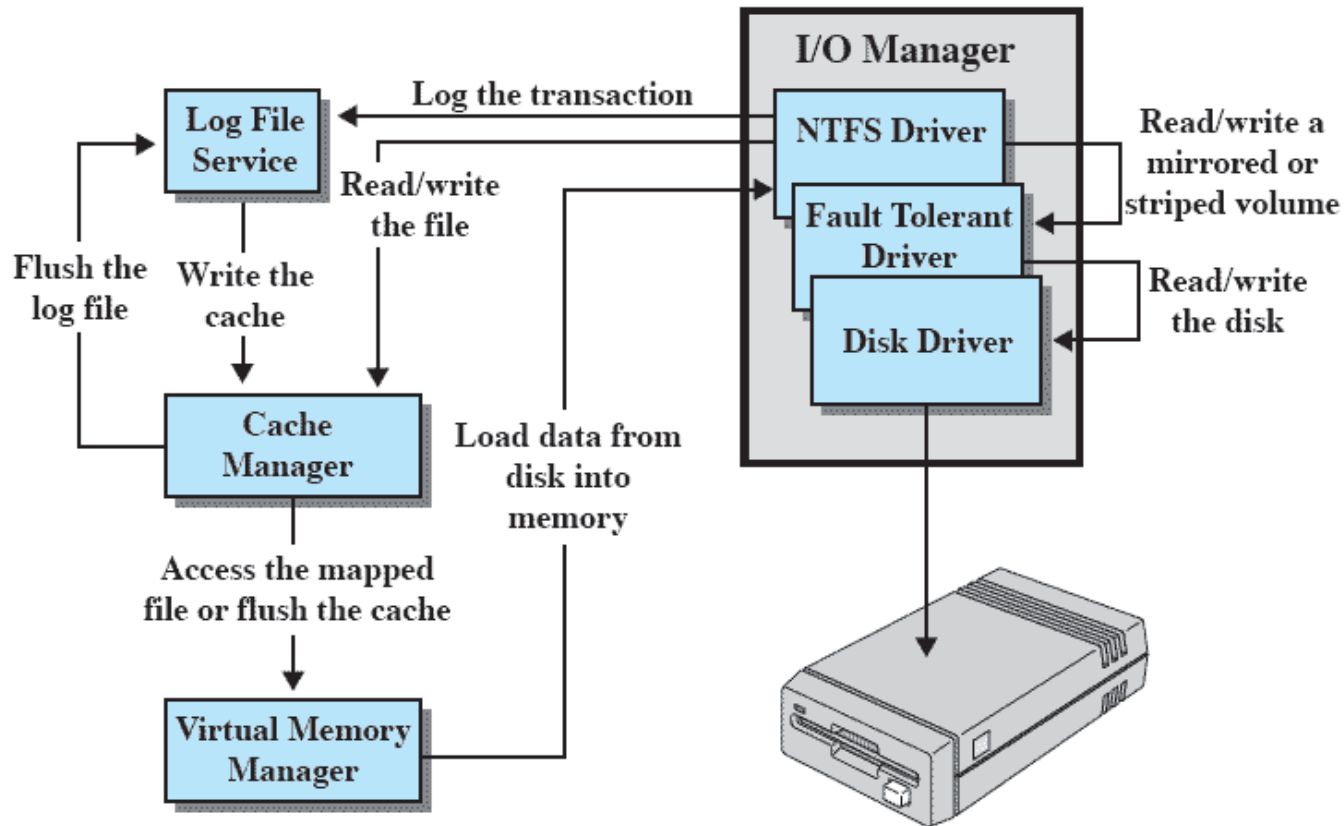
NTFS struktura volumena i datoteke

- ✿ Svaki element na volumenu predstavlja datoteku, i svaka datoteka se sastoji od kolekcije atributa.
 - ✦ Čak i podaci koji čine sadržaj datoteke se tretiraju kao atribut.
 - ✦ Layout NTFS volumena

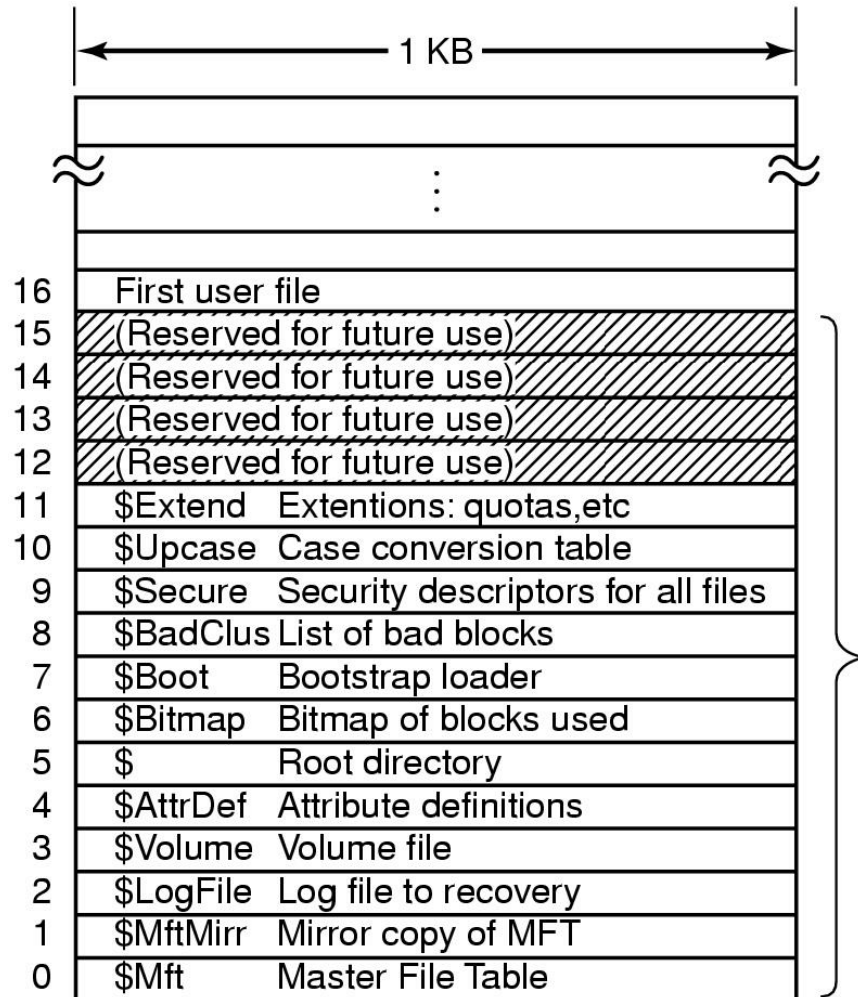


Komponente Windows NTFS-a

- Podrška za oporavljanje strukture fajl sistema



Master File Table - MFT (1)



Tanenbaum, 2008

Upravljanje datotekama

Operativni sistemi

- MFT je datoteka u vidu tabele slogova i može imati maksimalno 2^{48} slogova, pri čemu je svaki slog veličine 1024 B
- Svaki slog opisuje jedan fajl na volumenu, uključujući i MFT koja se tretira kao datoteka
- Prvih 16 slogova je namenjeno datotekama koje opisuju fajl sistem
- Svaki MFT slog se sastoji od sekvence atributa opisanih parom (*header*, vrednost)
- Prvi slog opisuje MFT datoteku i gde su locirani klasteri MFT datoteke (adresa prvog klastera MFT datoteke se nalazi u partition boot sector-u)
- Datoteka sa root direktorijumom je opisana u slogu 5
- Slog 6 opisuje datoteku sa bitmapom kojom se registruju slobodni i zauzeti blokovi (klasteri)



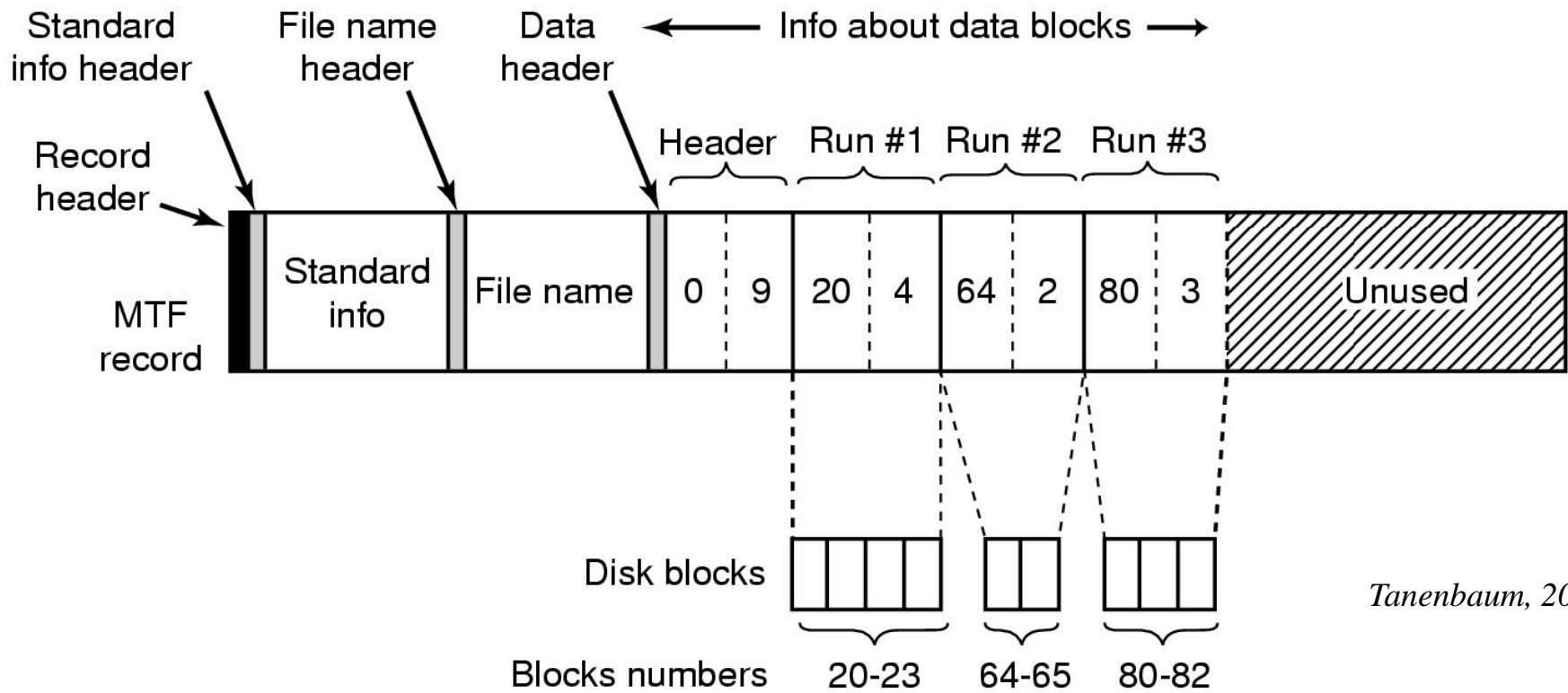
Atributi MFT sloga

(osenčani plavo su obavezni)

Attribute Type	Description
Standard information	Includes access attributes (read-only, read/write, etc.); time stamps, including when the file was created or last modified; and how many directories point to the file (link count).
Attribute list	A list of attributes that make up the file and the file reference of the MFT file record in which each attribute is located. Used when all attributes do not fit into a single MFT file record.
File name	A file or directory must have one or more names.
Security descriptor	Specifies who owns the file and who can access it.
Data	The contents of the file. A file has one default unnamed data attribute and may have one or more named data attributes.
Index root	Used to implement folders.
Index allocation	Used to implement folders.
Volume information	Includes volume-related information, such as the version and name of the volume.
Bitmap	Provides a map representing records in use on the MFT or folder.

MFT slog za datoteku (1)

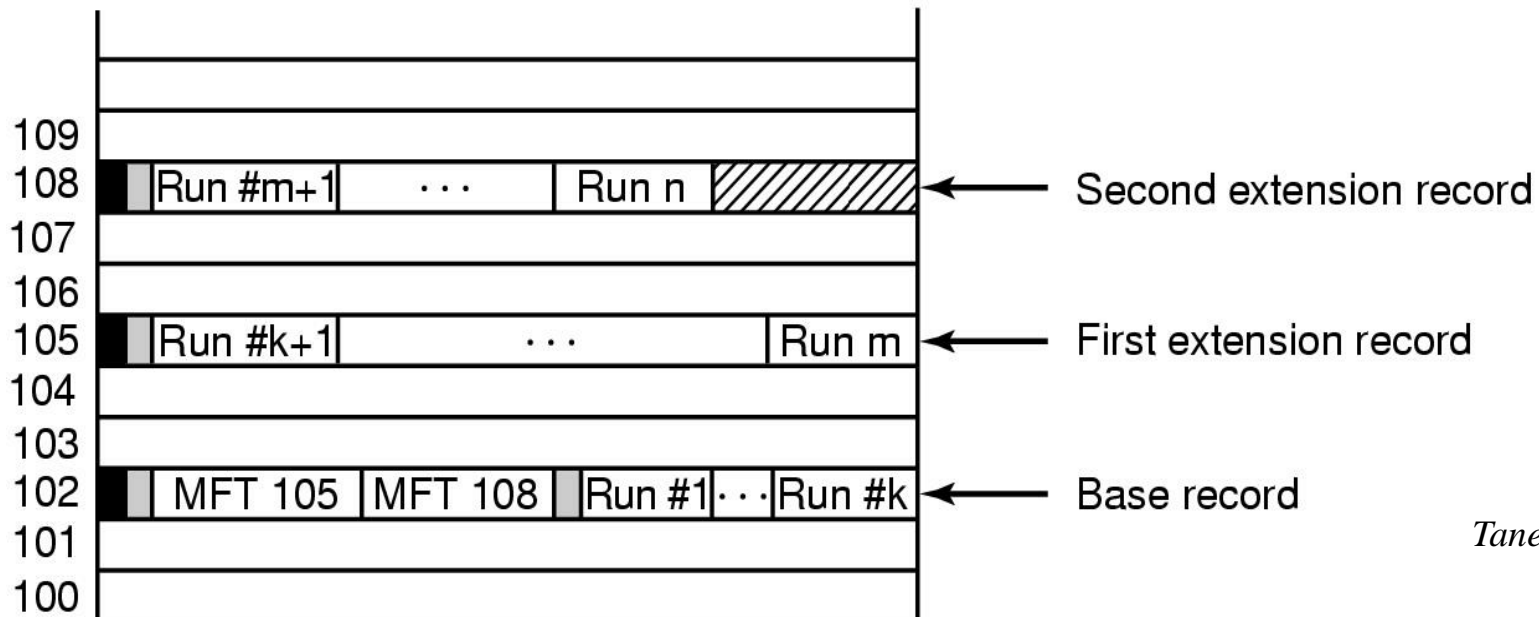
- MFT slog za datoteku od 9 blokova smeštenih na disku u tri grupe sukcesivnih blokova



Tanenbaum, 2014

MFT slog za datoteku (2)

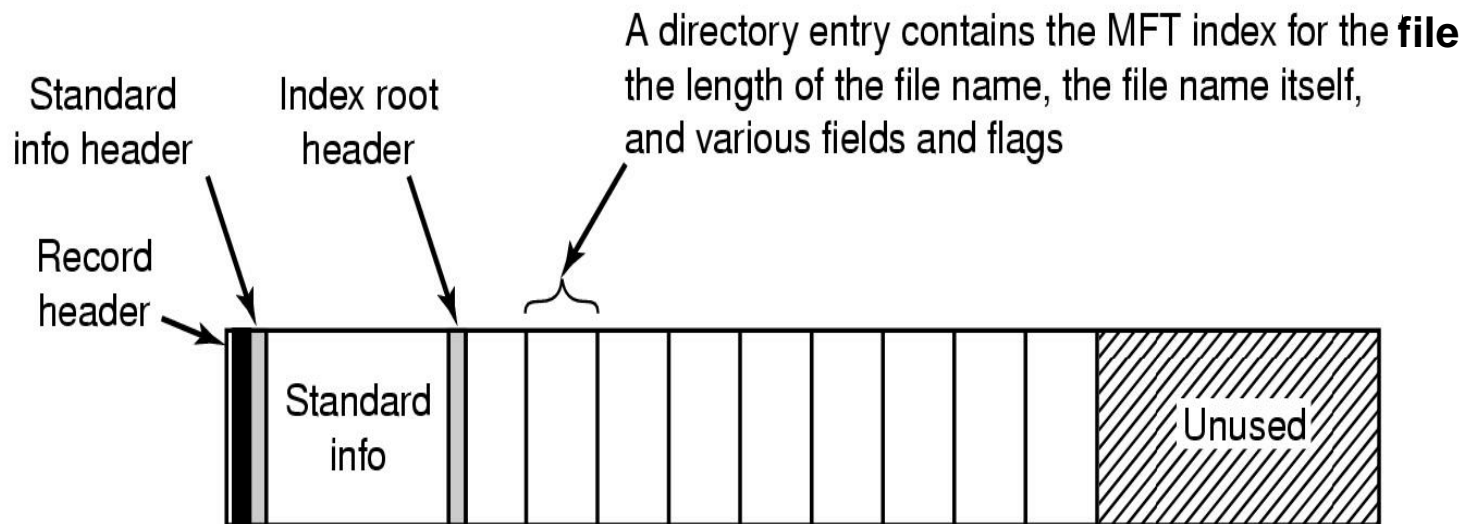
- Datoteka za koju je neophodno 3 MFT sloga za registrovanje blokova



Tanenbaum, 2014

MFT slog za folder

✶ MFT slog za mali folder

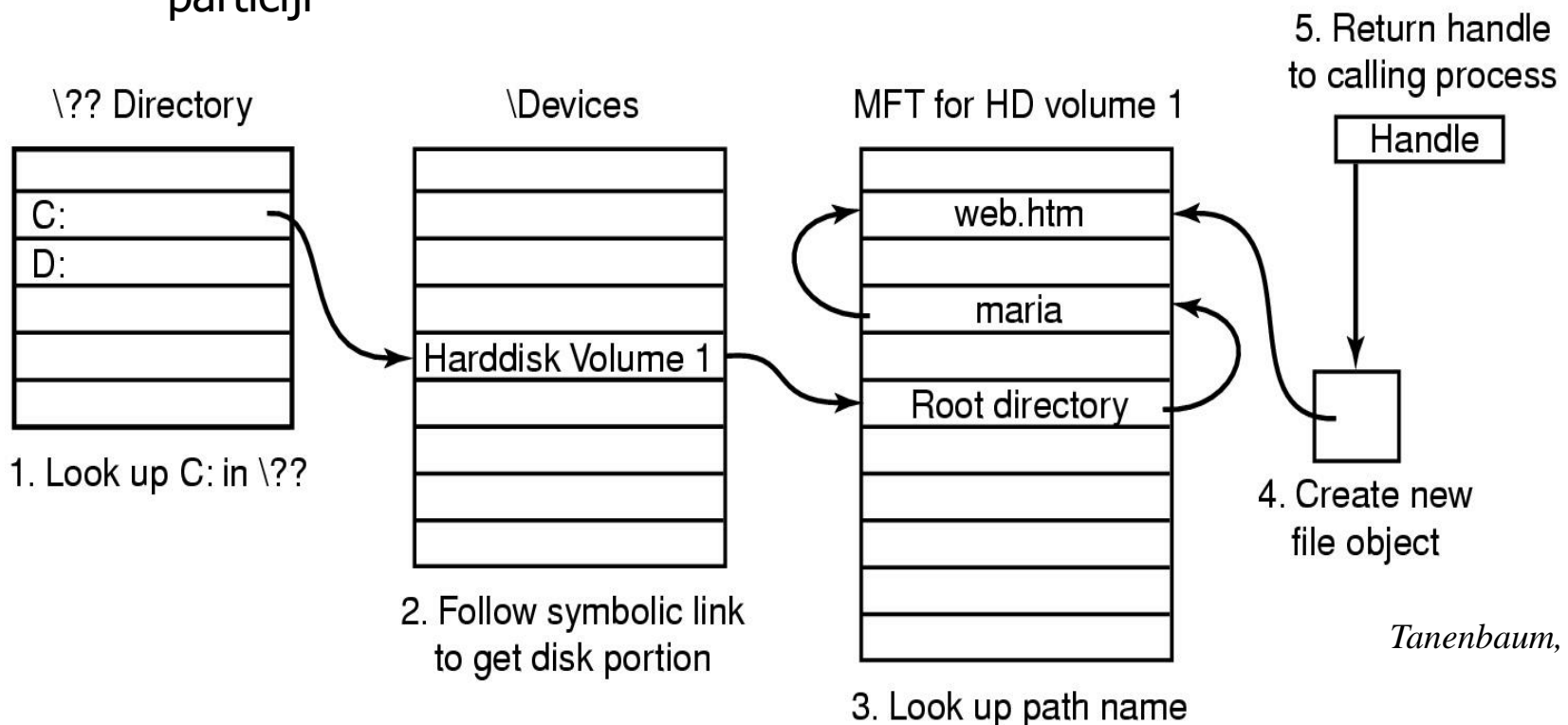


Tanenbaum, 2014

MFT - kreiranje datoteke

✿ Koraci u kreiranju datoteke C:\maria\web.htm

- ❏ Pretražuje se *Object manager* name space i direktorijum \?? da bi se našao C:, zatim na osnovu tok simboličkog linka pretražuje se direktorijum \Device i nalazi objekat \Device\HarddiskVolume1 koji odgovara prvoj partitiji prvog hard diska, a zatim pronalazi MFT za tu partitiju



Tanenbaum, 2014



UNIX upravljanje datotekama



Tipovi datoteka

- Regularni ili obični (*ordinary*)
- Direktorijum
- Specijalni
- Imenovani datavodi (*Named pipes*)
- Linkovi
- Simbolički linkovi

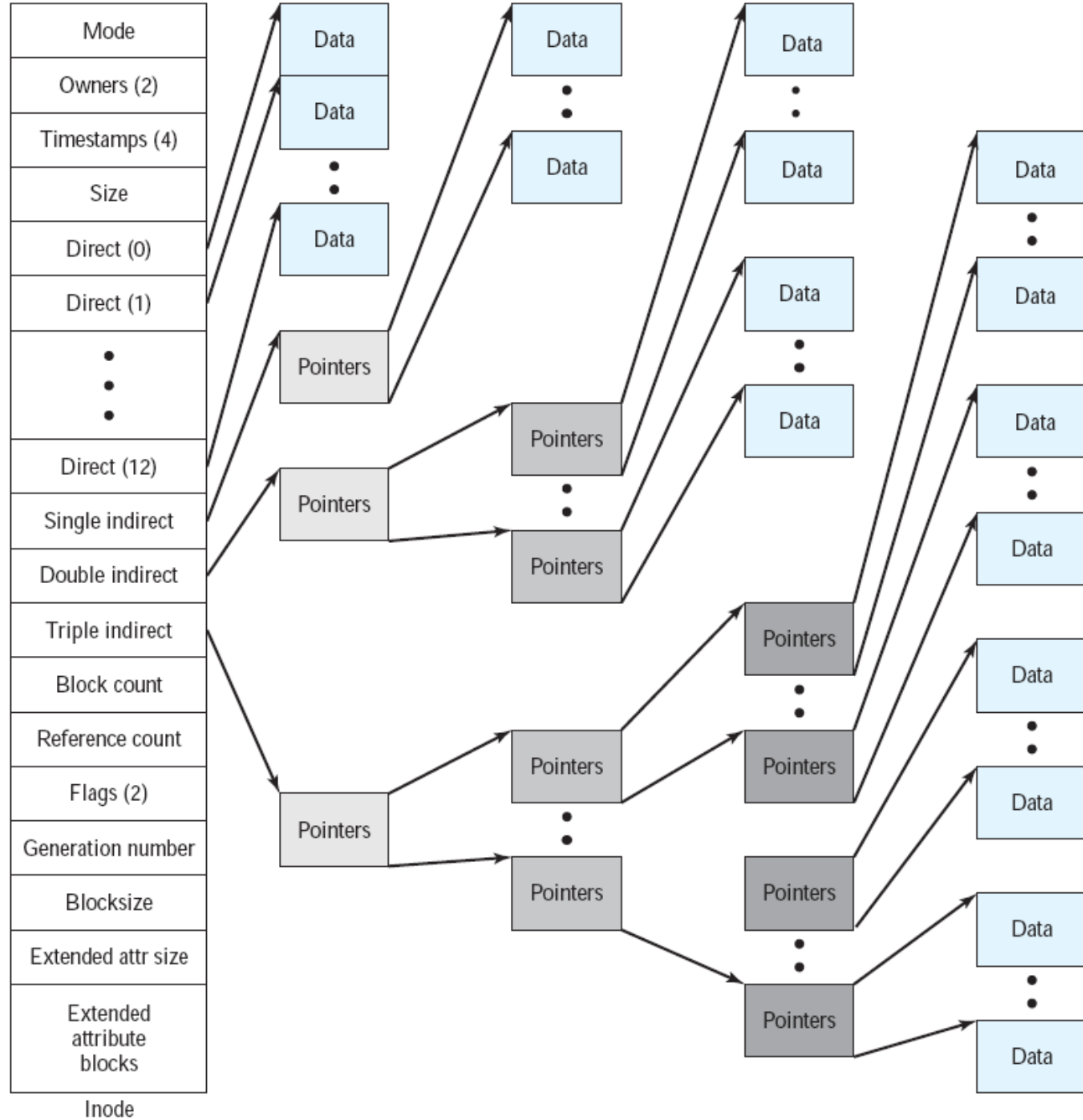
UNIX i-čvor (*i-node*)

✪ Sadrži osnovne attribute datoteke

File Mode	16-bit flag that stores access and execution permissions associated with the file. 12-14 File type (regular, directory, character or block special, FIFO pipe) 9-11 Execution flags 8 Owner read permission 7 Owner write permission 6 Owner execute permission 5 Group read permission 4 Group write permission 3 Group execute permission 2 Other read permission 1 Other write permission 0 Other execute permission
Link Count	Number of directory references to this inode
Owner ID	Individual owner of file
Group ID	Group owner associated with this file
File Size	Number of bytes in file
File Addresses	39 bytes of address information
Last Accessed	Time of last file access
Last Modified	Time of last file modification
Inode Modified	Time of last inode modification



FreeBSD UNIX i-node



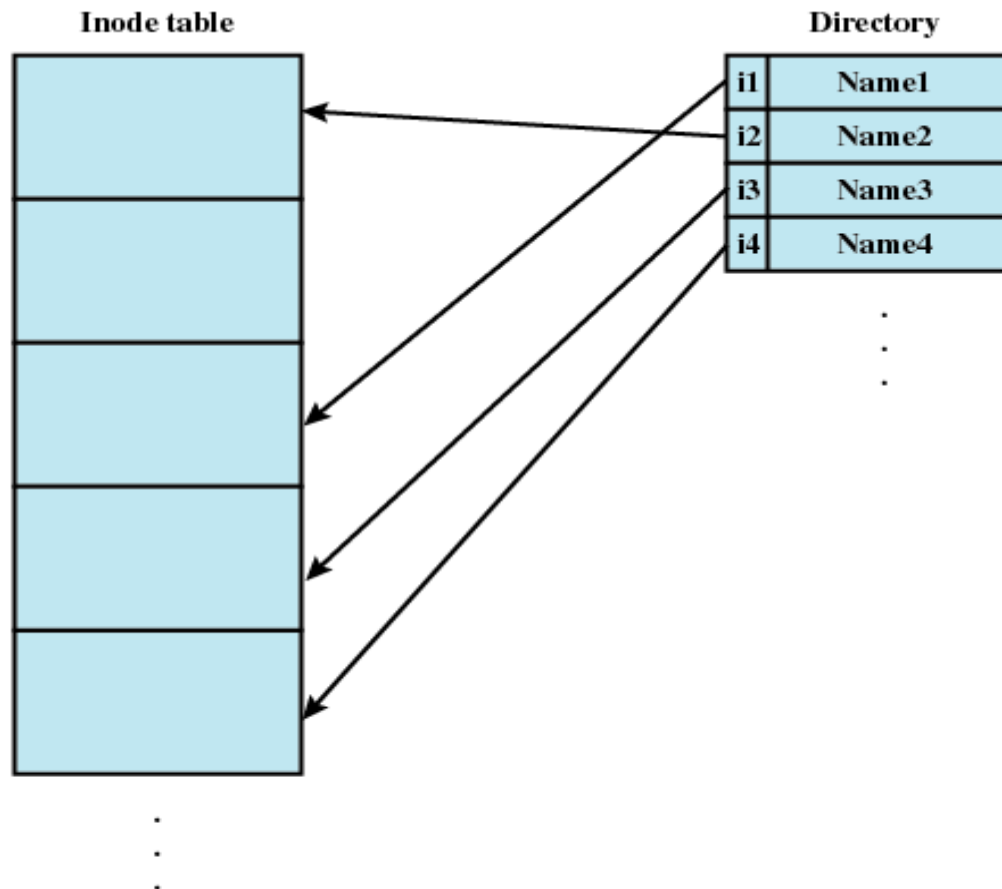
Inode

UNIX direktorijum i i-čvorovi



I-node
number

Ulaz direktorijuma



Direktorijum i i-čvorovi



UNIX file system (3)

Koraci u traženju */usr/ast/mbox*

Root directory

1	.
1	..
4	bin
7	dev
14	lib
9	etc
6	usr
8	tmp

Looking up
usr yields
i-node 6

I-node 6
is for /usr

Mode size times
132

I-node 6
says that
/usr is in
block 132

Block 132
is /usr
directory

6	.
1	..
19	dick
30	erik
51	jim
26	ast
45	bal

/usr/ast
is i-node
26

I-node 26
is for
/usr/ast

Mode size times
406

I-node 26
says that
/usr/ast is in
block 406

Block 406
is /usr/ast
directory

26	.
6	..
64	grants
92	books
60	mbox
81	minix
17	src

/usr/ast/mbox
is i-node
60

Tanenbaum, 2014

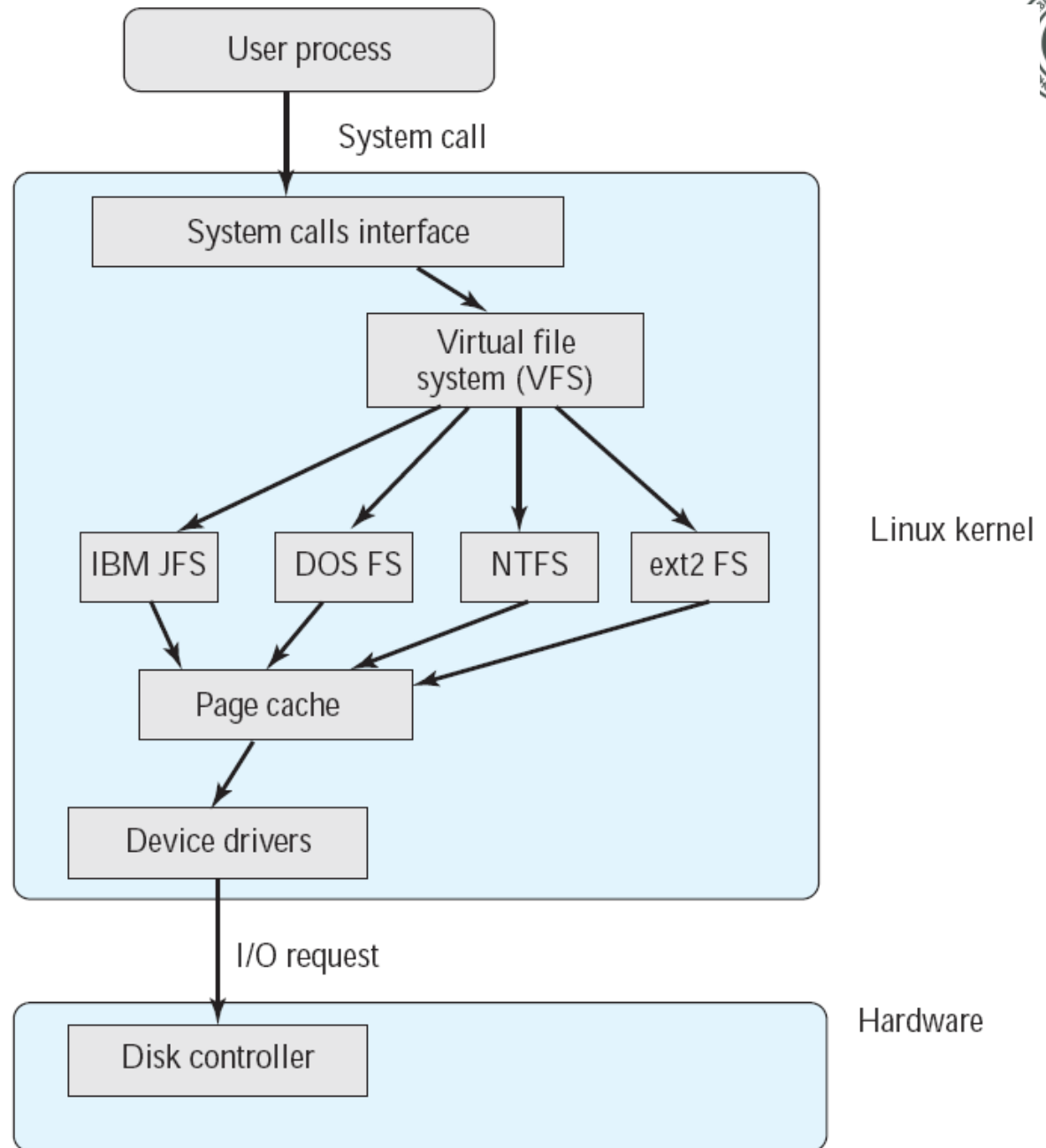


Linux virtualni file system (VFS)

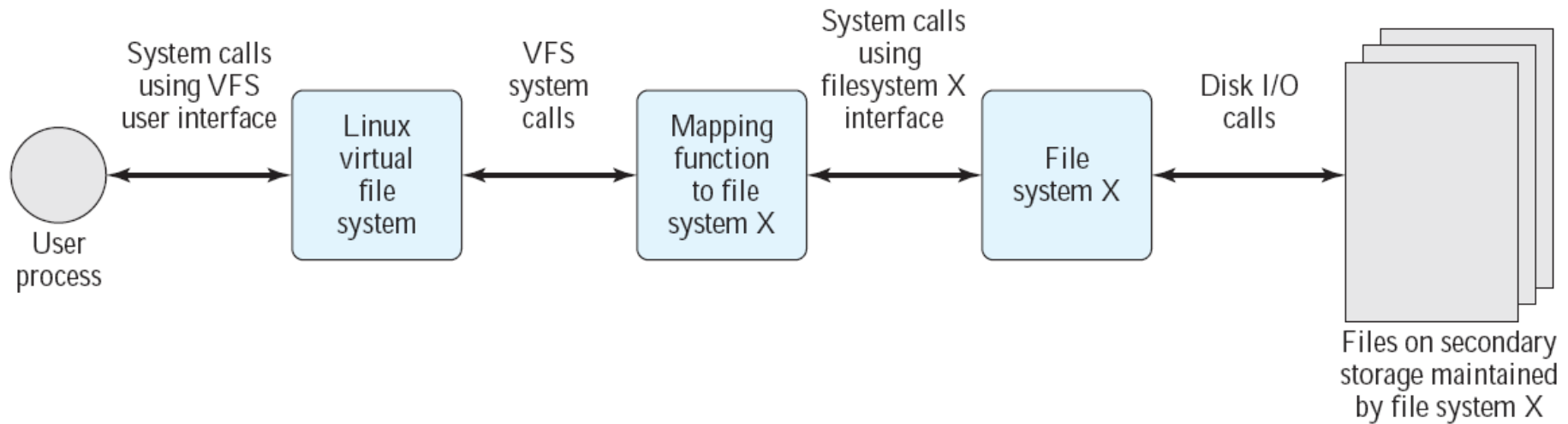
- ✿ Uniformni interfejs file system-a za korisničke procese
- ✿ Predstavlja opšte karakteristike i ponašanje svih podrazumevanih file system-a
- ✿ Polazi od pretpostavke da su datoteke objekti koji dele osnovna svojstva bez obzira na ciljni file system



Kontekst Linux VFS-a



Koncept Linux VFS-a





Primarni objekti u VFS-u (implementirani kao C strukture)

- ✿ *Superblock object*
 - ▣ Predstavlja specifičan mount-ovan file system
- ✿ *Inode object*
 - ▣ Predstavlja specifičnu datoteku
- ✿ *Dentry object*
 - ▣ Predstavlja specifičan ulaz direktorijuma (directory entry)
- ✿ *File object*
 - ▣ Predstavlja otvorenu datoteku pridruženu procesu



Domaći zadatak

❖ Poglavlje **12 Upravljanje datotekama**

- ❖ 12.14 Ključni pojmovi, kontrolna pitanja i problemi

❖ U/I animations

- ❖ <https://apps.uttyler.edu/Rainwater/COSC3355/Animations>
- ❖ Interrupt-Driven I/O Cycle
- ❖ The Life Cycle of an I/O Request
- ❖ Disk Scheduling Algorithms: FCFS, SSTF, SCAN