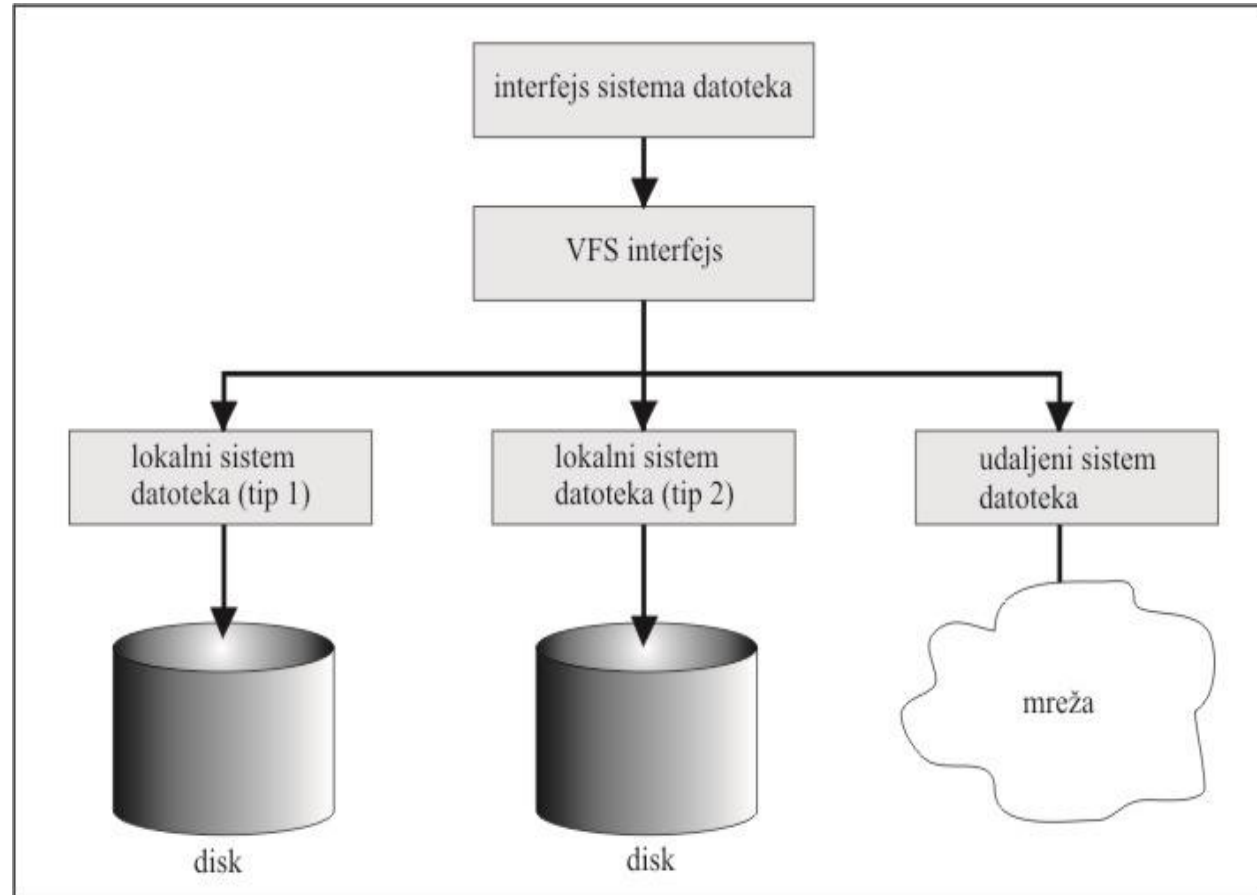


FS

- **Sistem datoteka**-skup metoda i struktura podataka koje operativni sistem koristi za čuvanje podataka
- **Struktura sistema datoteka:**
 - **1. zaglavlje**→neophodni podaci za funkcionisanje sistema datoteka
 - **2. strukture za organizaciju podataka na medijumu**→meta podaci
 - **3. podaci**→datoteke i direktorijumi
- Strukture podataka neophodne za realizaciju sistema datoteka:
 - **PCB**(Partition Control Block)
 - **BCB**(Boot control Block)
 - Kontrolne strukture za alokaciju datoteka(i-node tabela kod Linux-a)
 - Direktorijumske strukture koje sadrže kontrolne blokove datoteka
 - FCB(File Control Block)

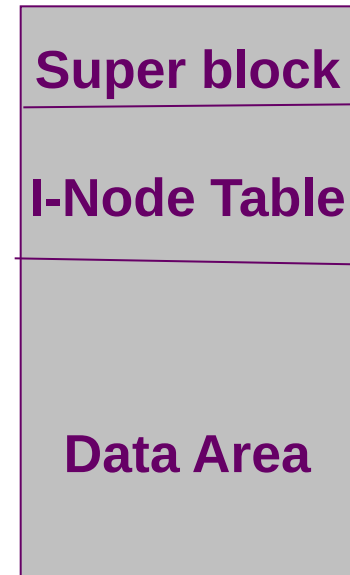
VIRTUELNI SISTEM DATOTEKA(VFS)

- Linux podržava rad sa velikim brojem sistema datoteka(ext2,ext3, XFS,FAT, NTFS...)
- VFS-objektno orjentisani način realizacije sistema datoteka koji omogućava korisniku da na isti način pristupa svim sistemima datoteka
- Način obraćanja korisnika sistemu datoteka
 - korisnik->API
 - VFS->sistem datoteka

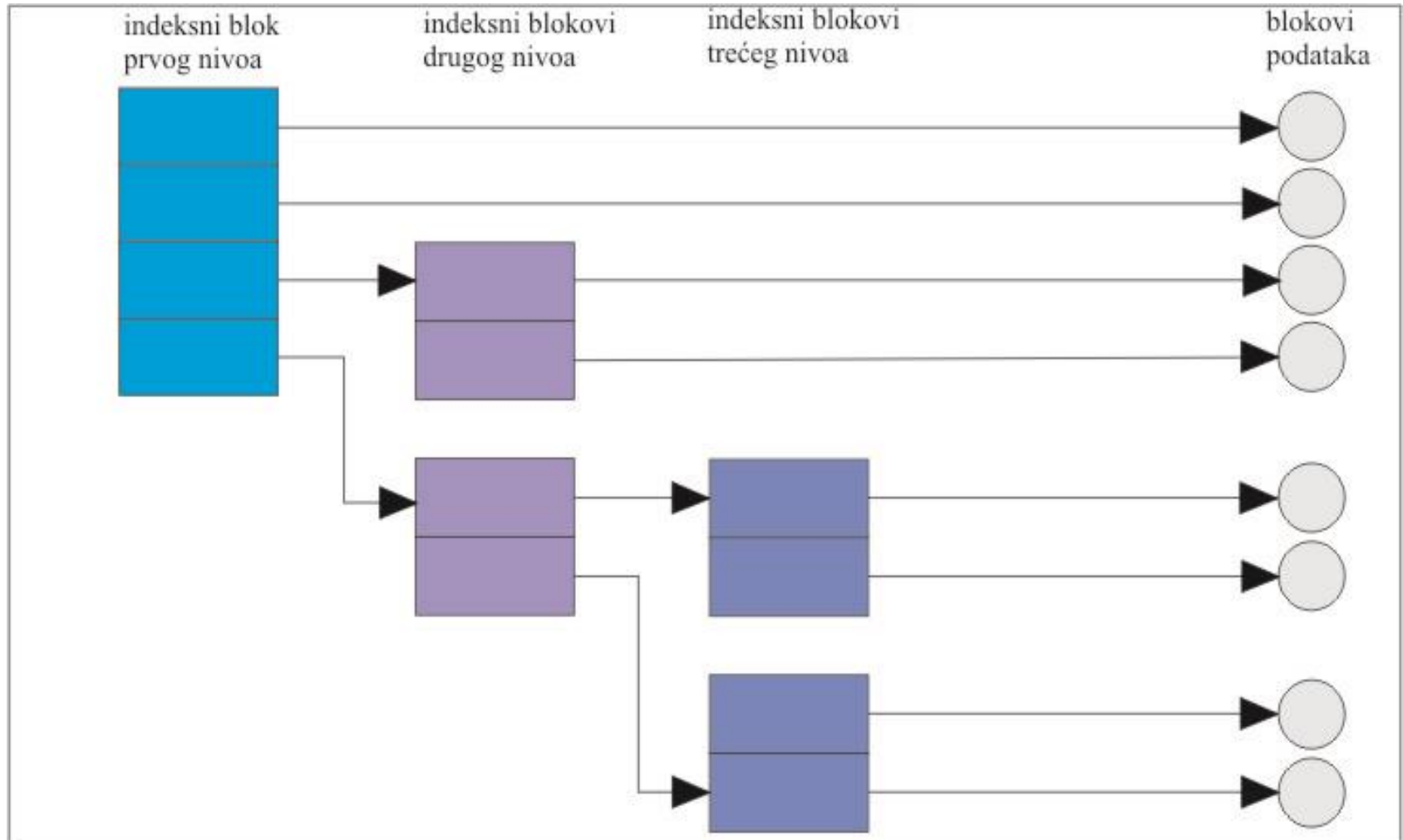


Linux FS

- Linux posmatra svaki sistem datoteka kao nezavisnu hijerarhijsku strukturu objekata(datoteka i direktorijuma) na čijem se vrhu nalazi *root(/)* direktorijum
- Objekti Linux sistema datoteka:
 - zaglavlje(*superblock*)
 - i-node tabela
 - blokovi sa podacima
 - direktorijumski blokovi
 - blokovi indirektnih pokazivača
- i-node-opisuje objekte, oko 128B na disku
- Kompromis između veličine i-node tabele i brzine rada sistema datoteka
 - prvih 10-12 pokazivača na blokove sa podacima
 - za alokaciju većih datoteka koristi se *single indirection block*
 - za još veće datoteke koriste se dvostruki i trostruki indirektni pokazivači



Inode



ext2 base features

- **Unix-like FS**
- **32 bit**
- **write-back cache**
- **long history of fixes and enhancements**
- **robustness**
- **extended attributes**
- **HTrees**
 - ☞ **HTrees make directory operations extremely fast**
 - ☞ **and**
 - ☞ **highly scalable to very large directories**

ext2

- Sistem datoteka **ext2-second extended file system**
- Razvoj sistema datoteka ext2
 - Minix-bez bagova, mnoga ograničenja
 - sistem datoteka ext-koristi VFS API, rešava osnovne probleme Minix-a
 - razvoj sistema datoteka xia-fs i ext2
- Sistem datoteka ext2 je bio baziran na sistemu datoteka ext, ali u potpunosti prerađen
- Uvedeno **dosta ideja** iz BSD FFS-a
- (**Default**) Podrazumevani sistem datoteka mnogih Linux distribucija

KARAKTERISTIKE SISTEMA DATOTEKA EXT2

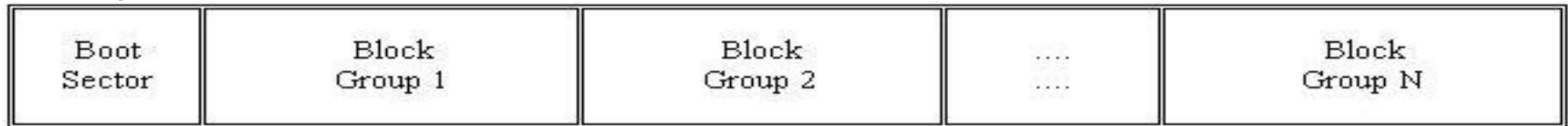
- Sistem datoteka ext2 podržava **maksimalnu veličinu datoteka** od **2TB**
- Maksimalna dužina imena datoteke je 255 karaktera
- Podržava standardne UNIX tipove datoteka
- Atributi datoteka omogućavaju korisnicima da modifikuju ponašanje kernela kada rade sa grupom datoteka
- Mogućnost izbora veličine blokova kod sistema datoteka: 1024, 2048, 4096
- Fast symbolic link
- Maksimalna veličina **volumena** je **16TB**

KARAKTERISTIKE SISTEMA DATOTEKA EXT2

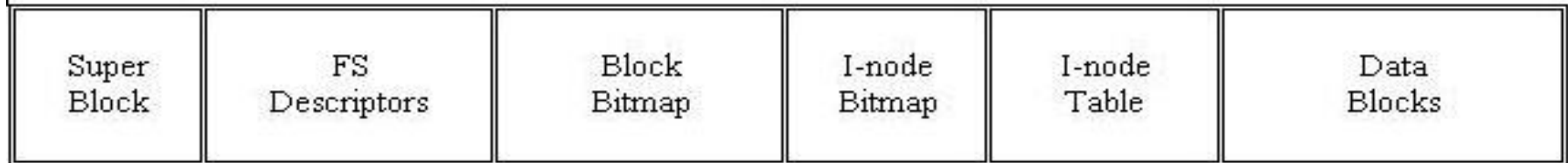
ext2	
Developer	Rény Card
Full name	Second extended file system
Introduced	January 1993 (Linux)
Partition identifier	Apple_UNIX_SVR2 (Apple Partition Map) 0x83 (MBR) EBD0A0A2-B9E5-4433-87C0-68B6B72699C7 (GPT)
Structures	
File allocation	i-nodes
Bad blocks	
Limits	
Max file size	2TB
Max number of files	10^{18}
Max filename size	255 characters
Max volume size	16TB
Allowed characters in filenames	Any byte except NULL and '/'
Features	
Forks	Yes
Attributes	
File system permissions	POSIX
Transparent compression	Yes (Optional)
Transparent encryption	No
Supported operating systems	Linux, BSD, Windows(through an IFS), MacOS X

FIZIČKA STRUKTURA SISTEMA DATOTEKA EXT2

- Uticaj forme BSD FFS-a na fizičku strukturu sistema datoteka ext2
- **Ext2 FS**



- **Block group**



- **FCB in Directory**



ext3 base features

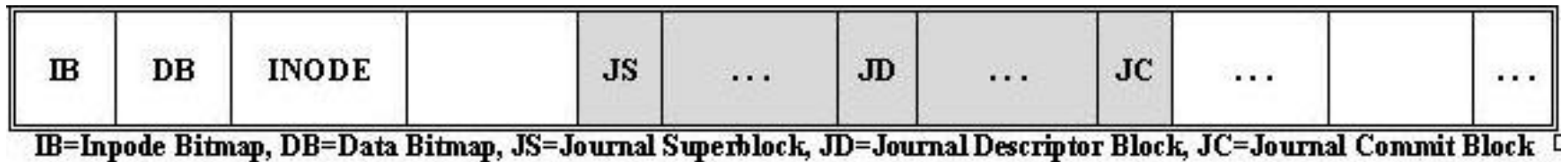
- **Unix-like FS**
- **ext3 = ext2 + journaling**
- **32 bit**
- **Full-compatible with ext2 FS**
- **Journaling**

ext3

- Razvio ga je dr. Stephen Tweedie; implementiran sa kernelom 2.4.15
- The **ext3 or third extended filesystem**
 - ☞ is a journalled file system
 - ☞ that is coming into increasing use among users of the Linux operating system.
- It is the **default file system for the Red Hat Linux, Fedora, Debian and Ubuntu Linux distributions.**
- **Although its performance and scalability is less attractive**
- **than many of its competitors such as ReiserFS and XFS**
- it does **have the significant advantage in that**
 - ☞ it **allows in-place upgrades from** the popular **ext2** file system
 - ☞ without having to backup and restore data.
- The ext3 file system **adds**, over its predecessor - ext2:
 - ☞ **A journal**
 - ☞ **H-tree (hashed tree) directory indexes**
 - ☞ **Online filesystem resizing**

KARAKTERISTIKE SISTEMA DATOTEKA EXT3 (1)

- Potpuno je kompatibilan sa sistemom datoteka ext2
 - jednostavno se može dodati journal na sistem datoteka ext2 i montirati kao ext3, ili demontirati(*unmount*) pa montirati ponovo kao ext2
- Struktura sistema datoteka ext3 je vrlo slična sistemu datoteka ext2



■ Atributi sistema datoteka ext3:

- No-atime
- Append-only
- Synchronous write
- No-dump
- H-Tree
- Immutable
- Secure delete
- Allow undelete

KARAKTERISTIKE SISTEMA DATOTEKA EXT3

ext3	
Developer	Open Source Community – Dr. Stephen Tweedie
Full name	Third extended file system
Introduced	November 2001 (Linux 2.4.15)
Partition Identifier	0x83 (MBR) EBD0A0A2-B9E5-4433-87C0-68B6B72699C7 (GPT)
Structures	
Directory contents	Table, HTree
File allocation	bitmap(free space), table (metadata)
Bad Blocks	Table
Limits	
Max file size	16GB-2TB
Max number of files	Variable
Max filename size	255 bytes
Max volume size	2TB-32TB
Allowed characters in filenames	All bytes except NULL and “/”
Features	
Dates recorded	modification (mtime), attribute modification (ctime), access (atime)
Date range	December 14, 1901 – January 18, 2038
Forks	Yes
Attributes	No-atime, append-only, synchronous-write, no-dump, h-tree (directory), immutable, journal, secure-delete, top (directory), allow-undelete
File system permissions	Unix permissions, ACLs and arbitrary security attributes (Linux 2.6 and later)
Transparent compresion	No
Transparent encryption	No (provided at the block device level)
Supported operating systems	Linux, BSD, Windows(trough an IFS)

EXT3

Razvoj	<i>Open Source Community – Društvo Otvorenog Koda</i>
Puno ime	<i>Third extended file system</i>
Prvo pojavljivanje	Novembar 2001, Linux jezgro 2.4.15
Identifikator particije	0x83 (MBR) EBD0A0A2-B9E5-4433-87C0-68B6B72699C7 (GPT)

Struktura

Sadržaj direktorijuma	tabela, stablo
Dodela prostora - alokacija	mapa bitova (<i>bitmap</i>) – <i>slobodan prostor</i> tabela – metapodaci
Ograničenja slobodnog prostora	tabela
Veličina datoteke	
Maksimalan broj datoteka	Ograničen je brojem indeksnih čvorova koji se određuje po kreiranju sistema datoteka i zavisi od veličine volumena, tj. particije i veličina bloka.
Maksimalna veličina imena datoteke	255 bajtova
Maksimalna veličina particije (volumena)	32 TB
Dozvoljeni karakteri u imenu datoteke	Svi osim NULL i '/'

veličina bloka	Maksimalna veličina datoteke	Maksimalna veličina sistema datoteka
1 KB	16 GB	2 TB
2 KB	256 GB	8 TB
4 KB	2 TB	16 TB
8 KB	2 TB	32 TB

EXT3

Datumi koji se beleže Mogućnosti	vreme poslednje izmene objekta (<i>mod time</i>), vreme poslednje izmene indeksnog čvora objekta (<i>i-node time</i>), vreme poslednjeg pristupa objektu (<i>access time</i>)
Raspon datuma	14. dec 1901 – 18. jan 2038
<i>Forks – dodatni podaci povezani sa objektom sistema datoteka</i>	Da
Atributi	No-atime, append-only, synchronous-write, no-dump, h-tree (directory), immutable, journal, secure-delete, top (directory), allow-undelete
Dozvole sistema datoteka	UNIX dozvole, ACL dozvole (Linux jezgro 2.6)
Kompresija na nivou sistema datoteka	Ne
Enkripcija na nivou sistema datoteka	Ne
Podržavani operativni sistemi	Linux, BSD, Windows (sa potrebnim upravljačkim programima)

Full Name: Third extended file system

- **Developer** Stephen Tweedie
- **Introduced** November 2001 (Linux 2.4.15)
- **Partition identifier** 0x83 (MBR)
- **Structures**
 - ☞ **Directory contents** Table, H-tree
 - ☞ **File allocation** bitmap (free space), table (metadata)
 - ☞ **Bad blocks** Table
- **Limits**
 - ☞ **Max file size** 2TiB
 - ☞ **Max number of files** Variable1
 - ☞ **Max filename size** 255 bytes
 - ☞ **Max volume size** 32TiB
- **Features**
 - ☞ **Dates recorded**
 - 📄 modification (mtime),
 - 📄 attribute modification (ctime),
 - 📄 access (atime)
 - ☞ **Date range** December 14, 1901 - January 18, 2038
 - ☞ **Forks** yes

Full Name: Third extended file system

■ Attributes:

- ☞ no-atime
- ☞ append-only
- ☞ synchronous-write
- ☞ no-dump
- ☞ h-tree (directory)
- ☞ immutable
- ☞ journal
- ☞ secure-delete
- ☞ top (directory)
- ☞ allow-undelete

■ File system permissions

- ☞ Unix permissions, rwx
- ☞ ACLs (Linux 2.6 and later)

■ Transparent compression No

■ Transparent encryption No (provided at the block device level)

ext3

- **ext3** is forward and backward compatible with **ext2**,
 - ☞ allowing users to keep existing file systems while very simply adding **journaling** capability.
 - ☞ **e2fsprogs** available
- **ext3**, like ext2, has a multi-vendor team of developers
- **ext3** provides a **generic journaling layer** (jbd)
- **external journaling support (NVRAM)**
- **ext3** has multiple **journaling modes**.

ext3

- **ext3** has broad **cross-platform compatibility**, working on
 - 👉 **32- and 64- bit architectures**
 - 👉 on both **little-endian** and **big-endian** systems.
- **ext3** does not require extensive core kernel changes
- **e2fsck** recovery program
- **ext3** uses this same **e2fsck** code for salvaging FS

Why ext3?

■ Why do you want to **migrate** from **ext2** to **ext3**?

■ **4 main reasons:**

- 👉 availability
- 👉 data integrity
- 👉 speed
- 👉 easy transition

Availability

- After an unclean system shutdown (unexpected power failure, system crash), each ext2 file system cannot be mounted until its consistency has been checked by the **e2fsck** program.
- The amount of time that the e2fsck program takes is determined primarily by the size of the file system, and for today's relatively large (many tens of gigabytes) file systems, this **takes a long time**.
- Also, the more files you have on the file system, the longer the consistency check takes. File systems that are several hundreds of gigabytes in size may take an hour or more to check. This severely limits availability.
- By contrast, **ext3 does not require a file system check**, even after an unclean system shutdown, **except for certain rare hardware failure cases** (e.g. hard drive failures). This is because the data is written to disk in such a way **that the file system is always consistent**.
- The **time to recover an ext3 file system after an unclean system shutdown does not depend on the size of the file system or the number of files**; rather, it depends on the size of the "journal" used to maintain consistency.
- The **default journal size** takes about a **second to recover** (depending on the speed of the hardware).

Data Integrity

- Using the **ext3** file system can **provide**
 - 👉 **stronger guarantees about data integrity**
 - 👉 **in case of an unclean system shutdown.**

speed

- Despite writing some data more than once,
 - ☞ **ext3 is often faster than ext2**
 - ☞ **because ext3's journaling optimizes hard drive head motion.**
- You can choose from **three journaling modes** to optimize speed, optionally choosing to trade off some data integrity.
 - ☞ **One mode, `data=writeback`, limits the data integrity guarantees, allowing old data to show up in files after a crash**, for a potential increase in speed under some circumstances. (This mode, which is the default journaling mode for most journaling file systems, essentially provides the more limited data integrity guarantees of the **ext2** file system and merely avoids the long file system check at boot time.)
 - ☞ **Second mode, `data=ordered` (the default mode), guarantees that the data is consistent with the file system**; recently-written files will never show up with garbage contents after a crash.
 - ☞ **Last mode, `data=journal`, requires a larger journal for reasonable speed in most cases and therefore takes longer to recover in case of unclean shutdown, but is sometimes faster for certain database operations.**

Easy Transition

- It is easy to change from **ext2** to **ext3** and
 - 👉 gain the benefits of a robust journaling file system,
 - 👉 without reformatting.
- **ext2 to ext3**
- **#tune2fs -j /dev/hdaX**
- **back to ext2**
- **# tune2fs -O ^has_journal /dev/hdaX**

JOURNALING (1)

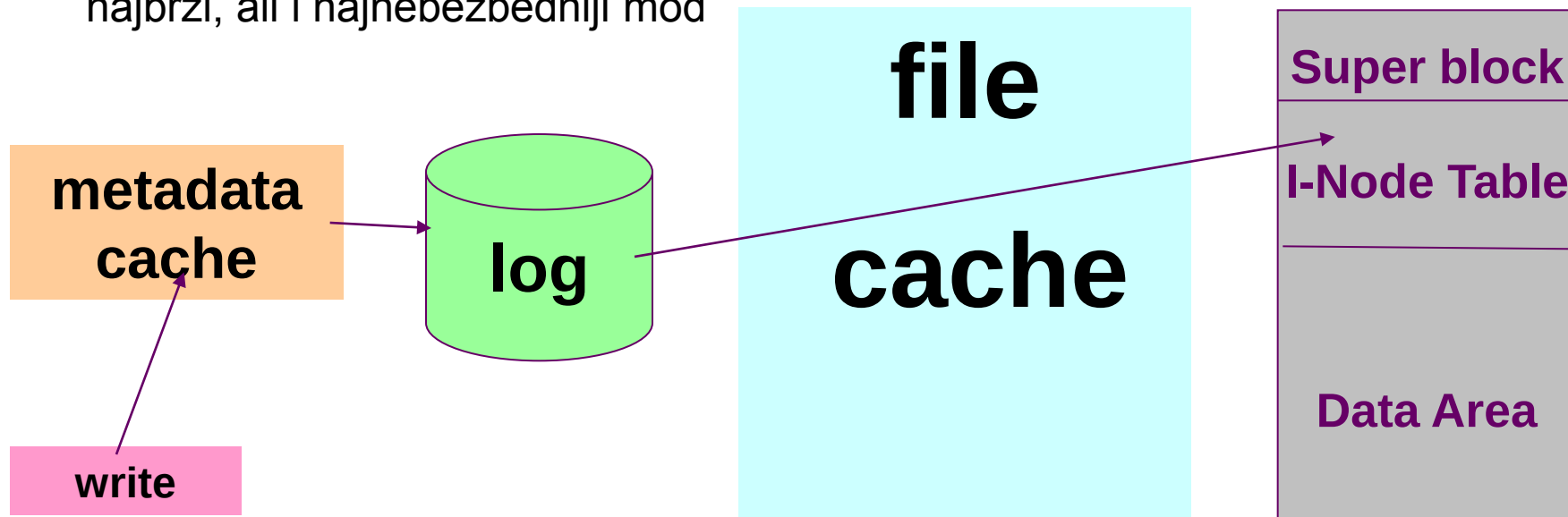
- Najvažnija karakteristika sistema datoteka ext3
- Usled nasilnog prekida rada sistema(nestak struje, kvar hardvera...) može doći do gubitka podataka
- Journal-datoteka koja se čuva na sistemu datoteka
- Dnevnik(*journal*) se jednostavno dodaje na sistem datoteka ext2:
tune2fs -j /dev/hdaX
- Dnevnik se može dodati i na montirani i na nemontirani sistem datoteka
- Posle dodavanja dnevnika sistem datoteka može da se montira kao ext3
mount -t ext3 /dev/hdaX /mnt/negde
- Kod velikih sistema datoteka ext2, **fsck** je mogao, posle pada sistema, da traje i više od sat vremena
- Sistem datoteka ext3 ne zahteva **fsck** posle restart-a
- Sa standardnom veličinom dnevnika na sistemu datoteka ext3, podizanje sistema posle nestanka struje traje oko sekund

JOURNALING (2)

- Journaling je implementiran u sistem datoteka ext3 preko posebnog sloja koji se zove JFS (*JFS layer*)
- JFS sloj vodi računa o kompletnom journaling-u, sistem datoteka nema veze sa journaling-om
- JFS obezbeđuje garancije upisa-ili se cela transakcija izvršava, ili ništa
- Transakcije se ne rade sinhrono, JFS sloj određuje vreme izvršavanja
- Sistem datoteka koristi(kao i ostali) **redo logging**
- Kod journaling-a postoji potvrda o izvršenju (*commit record*), koji označava kraj transakcije
- **Checkpoint** - dnevnik je ograničene veličine, a checkpointing nam omogućava da ponovo koristimo bitove dnevnika
- Transakcije su ograničene veličine, a mogu da postanu jako velike jedino kod sistemskih poziva za upis ili odsecanje(*truncate*)

JOURNALING MODОВИ SISTEMA DATOTEKA EXT3

- Sistem datoteka podržava tri journaling moda
- **Journal mod**-čuvaju se i podaci i meta strukture. Najsporiji od sva tri moda, ali najsigurniji
- **Ordered mod**-u dnevniku se čuvaju samo meta strukture, ali zato prebacije ažurirane promene u datotekama na disk pre nego što napravi promene u meta strukturama. Ovo je podrazumevani journaling mod
- **Writeback mod**-čuva samo promene meta struktura i oslanja se na standardni proces upisa da upiše promene u datotekama na disk. Ovo je najbrži, ali i najnebezbedniji mod



ext3

- There are three levels of journaling available in the of ext3:
- **Journal,**
 - ☞ where both metadata and file contents are written to the journal before being committed to the main filesystem.
 - ☞ This improves reliability at a performance penalty because all data has to be written twice.
- **Writeback,**
 - ☞ where metadata is journaled but file contents are not.
 - ☞ This is faster, but introduces the hazard of out-of-order writes where,
 - ☞ for example,
 - 📄 files being appended to during a crash
 - 📄 may gain a tail of garbage on the next mount.
- **Ordered,**
 - ☞ as with writeback, but forces file contents to be written after its associated metadata is.
 - ☞ This is thought to be an acceptable compromise
 - ☞ between reliability and performance, and hence is the default.
- Versions of ext3 since November 2004 also support **online resizing**.
 - ☞ This means that an ext3 filesystem can be grown while being used, without unmounting.

Journaling modes in ext3

- **journal**

- **writeback**

- **ordered**

Option= journal

- **journal - logs all**
 - ☞ filesystem data
 - ☞ metadata changes
- **slowest of the three ext3 journaling modes,**
 - ☞ **this journaling mode minimizes**
 - ☞ **the chance of losing the changes** you have made to any file in an ext3.
- Using the "journal" mode requires that an ext3 filesystem write **every change to a filesystem twice** –
 - ☞ **once to the journal**, and then a
 - ☞ **gain to the filesystem itself.**
- **This can reduce the overall performance** of your filesystem,
- **but is the mode most beloved by users,**
- **because it minimizes the chances of losing changes to your files**
 - ☞ **since both metadata and data updates are recorded**
 - ☞ **in the ext3 journal**
 - ☞ **and can be replayed** when a **system reboots.**

Option: ordered

- **ordered** - only logs changes to filesystem metadata,
 - ☞ but flushes file data updates to disk
 - ☞ before making changes to associated filesystem metadata.
 - ☞ This is the **default ext3 journaling mode**.
- Using the **"ordered"** mode,
 - ☞ **only filesystem metadata changes are logged,**
 - ☞ **which reduces redundancy between writing to the filesystem**
 - ☞ **and to the journal and is therefore faster.**
- Though the **changes to file data are not logged,**
 - ☞ **they must be done**
 - ☞ **before associated filesystem metadata changes are made by the ext3 journaling daemon,**
 - ☞ **which can slightly reduce the performance of your system.**
- However, using this journaling mode
 - ☞ **guarantees that files in the filesystem**
 - ☞ **will never be out of sync**
 - ☞ **with any related changes to filesystem metadata**

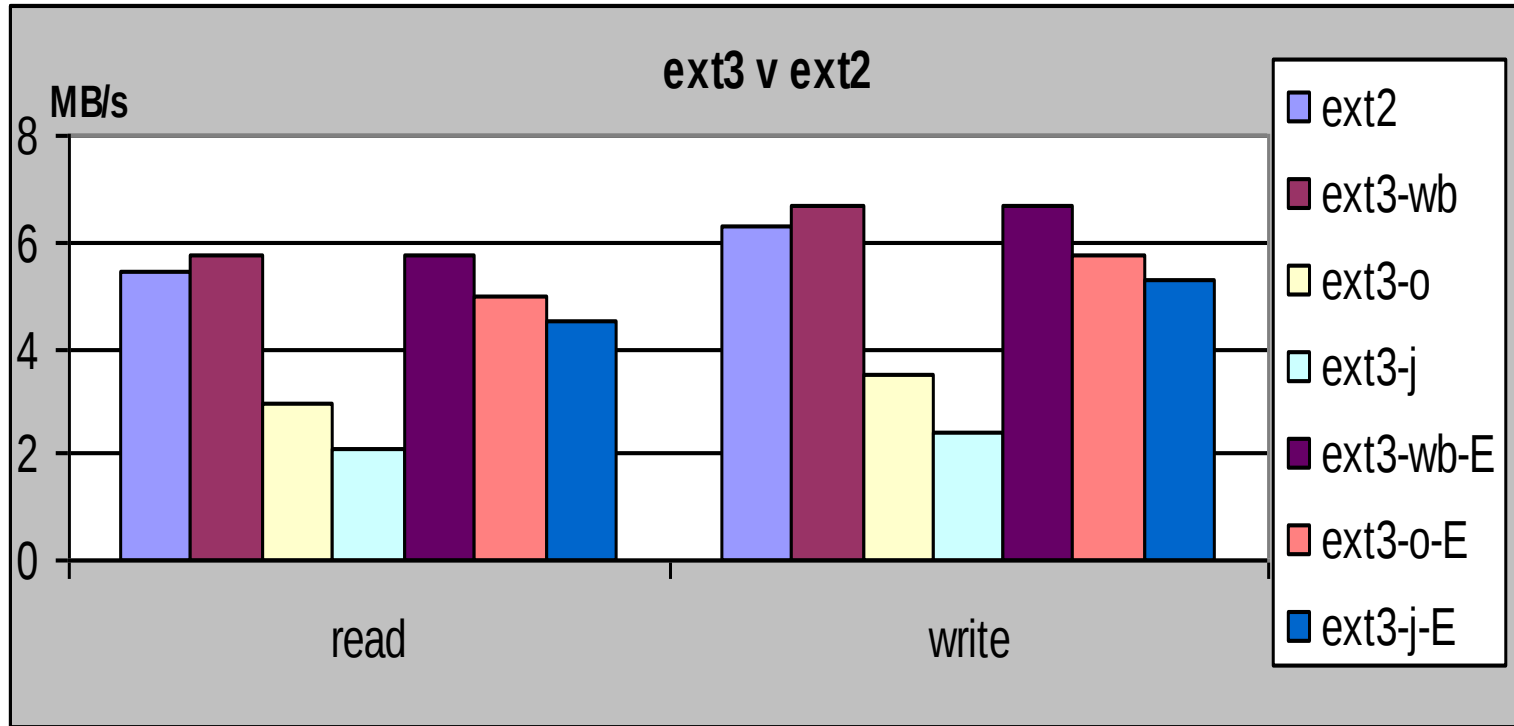
Option: writeback

- **writeback - only logs changes to filesystem metadata**
 - ☞ **but relies on the standard filesystem write process**
 - ☞ **to write file data changes to disk.**
 - ☞ **This is the fastest ext3 journaling mode.**
- **Using the "writeback" mode is faster than the other two ext3 journaling modes because**
 - ☞ **it only logs changes to filesystem metadata and**
 - ☞ **does not wait for associated changes to file data to be written**
 - ☞ **before updating things like file size and directory information.**
- **Because updates to file data are done asynchronously to journaled changes to filesystem metadata,**
 - ☞ **files in the filesystem may exhibit metadata inconsistencies**
 - ☞ **such as owning data blocks to which updated data**
 - ☞ **was not yet written when the system went down.**
- **This isn't fatal, but can be disappointing to users.**

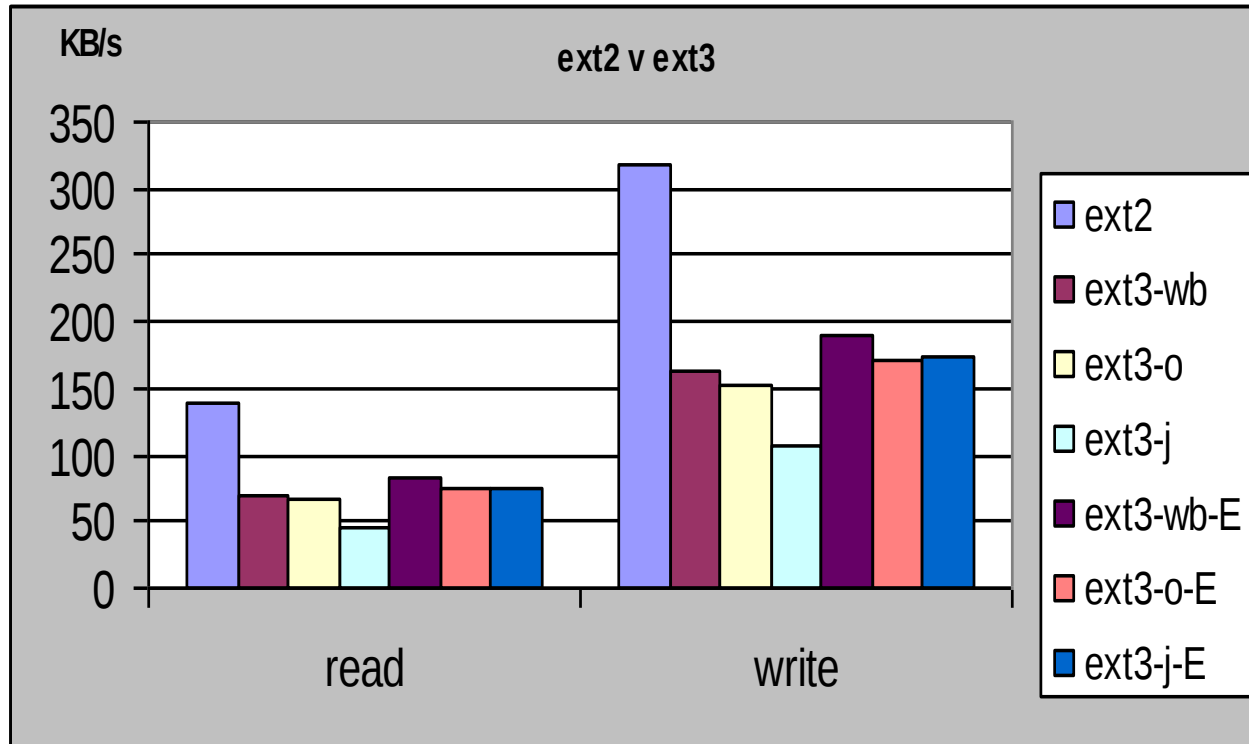
External journaling

- Journal outside of FS
- Journal position
 - ☞ Separate disk partition
 - ☞ NVRAM
- Simulation of NVRAM using RAM disk
 - ☞ To install trd:
 - ☞ **#mknod /dev/trd b 240 0**
 - ☞ **#insmod trd.o trd_size=50000** (For a 50 megabyte device)
 - ☞ To create an external journal:
 - ☞ **#mke2fs -O journal_dev /dev/trd**
 - ☞ To create an ext3 filesystem on /dev/hda5 which uses the external journal device:
 - ☞ **#mke2fs -J device=/dev/trd /dev/hda5**
 - ☞ **#mount /dev/hda5 /mnt/some/place -t ext3**

Journaling: Small file performance



Journaling: Ultra-small file performances



PREDNOSTI I NEDOSTACI SISTEMA DATOTEKA EXT3

■ Prednosti:

- **Dostupnost**-posle nasilnog gašenja, ext3 ne zahteva proveru sistema datoteka(sistem datoteka je uvek konzistentan)
- **Integritet podataka**-moguć izbor tipa i nivoa zaštite koji se primenjuje na datoteke
- **Brzina**-journaling sistema datoteka ext3 optimizuje kretanje glave diska, a moguć je izbor tri journaling moda
- **Jednostavno prebacivanje**-veoma je jednostavno prebaciti se sa sistema datoteka ext2 na ext3

■ Nedostaci:

- Kompatibilnost sa sistemom datoteka ext2, onemogućava sistemu datoteka ext3 uvođenje nekih naprednih funkcija
- Relativno mala veličina i za datoteke i za ceo sistem datoteka
- Postavljanje nula na pokazivače blokova u i-node-u za obrisane datoteke

POBOLJŠANJA I NOVE VERZIJE SISTEMA DATOTEKA EXT3

- **Poboljšavanje** sistema datoteka ext3 bez promene formata
 - zamena tradicionalne šeme indirektnog mapiranja blokova ima više prednosti, ali implementacija ove zamene nije jednostavna zbog nekompatibilnosti sa starijim verzijama
- Smanjenje korišćenja baferskih zaglavlja(buffer_head) u sistemu datoteka ext3
- Odložena alokacija bez extenata
- Efikasna alokacija višestrukih blokova
- Asinhrono brisanje linka/odsecanje datoteke
- Proširena podrška za nlink-ove

POBOLJŠANJA I NOVE VERZIJE SISTEMA DATOTEKA EXT3

■ Indeksiranje direktorijuma(*directory indexing*)

- sistem datoteka ext3 je koristio povezane liste(BSD FFS)
- veliki problemi za performanse kod pojedinih aplikacija
- ubacivanje H-stabala(*Hash Trees*) u direktorijumske strukture
- **H-stabla koriste 32-bitne heševe** kao ključeve gde svaki ključ odgovara određenom opsegu vrenosti u granskom bloku
- **H-stabla imaju veoma veliki faktor raspona**
- povećanje performansi za faktor od 50-100

■ Poboljšavanje skalabilnosti

- problem skalabilnosti u ext3 kodu i JBD-u(*generic journaling layer*)
- Alex Tomas i Andrew Morton otklanjaju zaključavanja u sistemu datoteka
- ubačena je **fina podela zaključavanja** (umesto **lock_super()**)
- JBD rutine su koristile BKL(Big Kernel Lock)
- ubacuje se **nova podela zaključavanja**

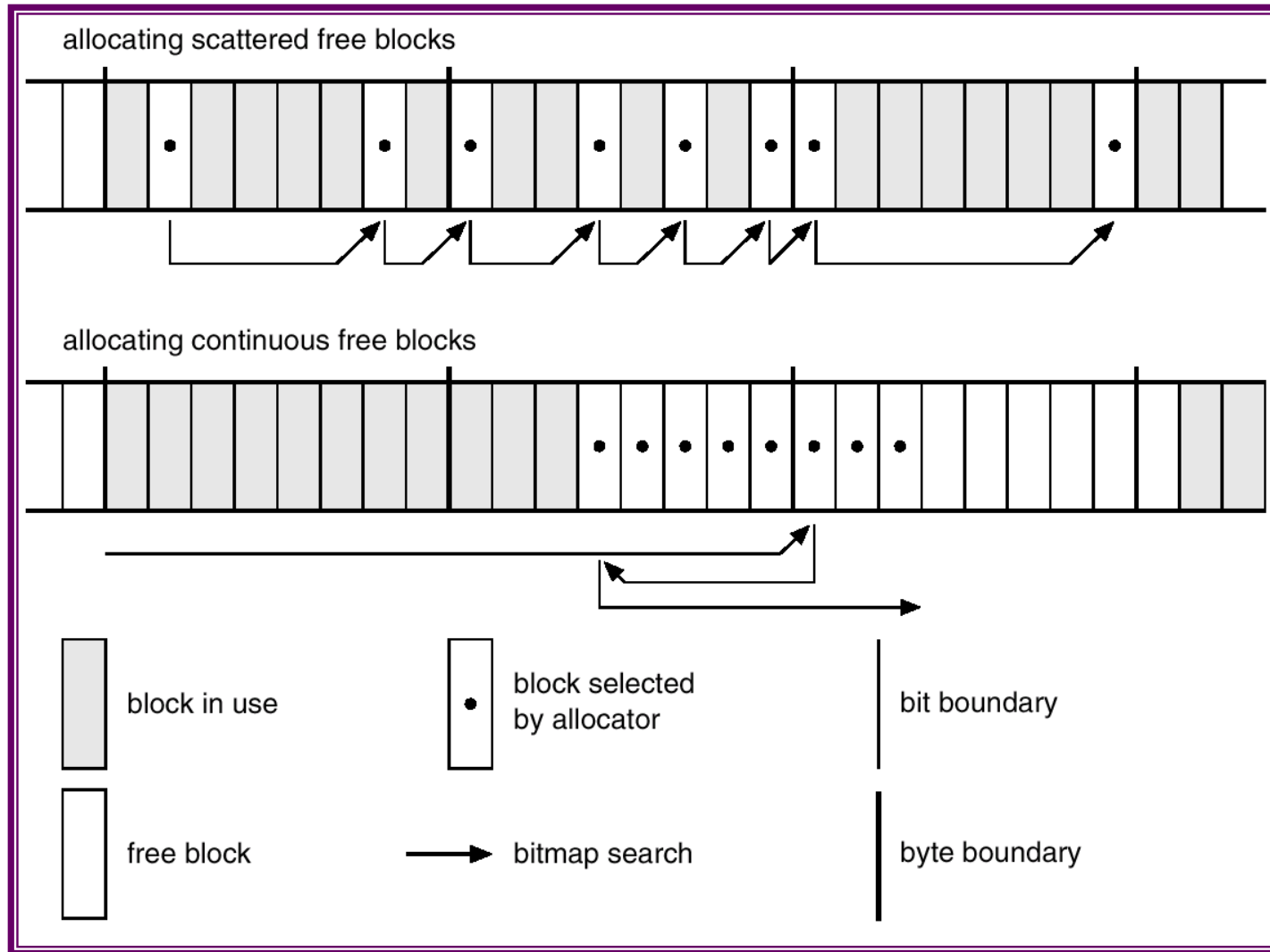
POBOLJŠANJA I NOVE VERZIJE SISTEMA DATOTEKA EXT3

■ **Alokator blokova baziran na rezervaciji**

■ **(Reservation based block allocator)**

- sistemi datoteka uvek pokušavaju da smeste datoteke kontinualno u cilju smanjenja pomeraja glave diska što je više moguće
- ako sistem datoteka alokira blokove *on-demand*, onda kada se dve datoteke upisuju simultano, alokacija blokova može da se izmeša
- neki sistemi datoteka kao rešenje koriste **metodu prealokacije**
- kod rezervacije, kada i-node-u zatrebaju blokovi, alokator **rezerviše opseg blokova (rezervacioni prozor)**
- kod dodele novog bloka, regija mora da bude blizu ciljnog bloka, ne sme da se ukrsta sa drugim rezervacionim prozorom i regija mora da ima bar jedan slobodni blok
- **rezervacioni prozori** su **indeksirani** kao **red-black tree** kako bi alokator mogao da odredi da li je konkretna regija već rezervisana od strane drugog i-node-a

Ext2fs Block-Allocation Policies



POBOLJŠANJA I NOVE VERZIJE SISTEMA DATOTEKA EXT3 (3)

- **Online promena veličine sistema datoteka** (Online file system resizing)
- ovo svojstvo podrazumeva promenu veličine sistema datoteka dok je montiran
- kod za promenu veličine zavisi od blok uređaja
- **tri faze** po kojima se sistem datoteka povećava:
 - ☞ 1. proširivanje poslednje parcijalne grupe blokova kako bi postala kompletna
 - ☞ 2. dodavanje nove grupe blokova već postojećem bloku u tabeli za opis grupe blokova
 - ☞ 3. dodavanje novog bloka u tabelu za opis grupe blokova i dodavanje nove grupe tom bloku

POBOLJŠANJA I NOVE VERZIJE SISTEMA DATOTEKA EXT3 (4)

■ Prošireni atributi

- mnoga nova svojstva operativnih sistema zahtevaju od sistema datoteka da budu sposobni da povežu male količine meta podataka sa datotekama
- Andreas Gruenbacher je ubacio proširene attribute u sistem datoteka ext3
- prošireni atributi se čuvaju u jednom bloku koji mogu da dele i-node-ovi sa istim proširenim atributima
- Alex Tomas i Andreas Dilger su implementirali podršku za čuvanje proširenih atributa u proširenom i-node-u sistema datoteka ext3
- i-node mora biti veći od 128B

■ Extenti, odložena alokacija i alokacija extenata

- Za ova poboljšanja postoje zavrpe, ali još nisu integrisane u kernel

■ Mape extenata

- extent-alokaciona jedinica sistema datoteka promenljive veličine
- mapiranje između logičkih i fizičkih blokova za velike datoteke
- problem oko implementacije jer zahteva promenu formata sistema datoteka
- implementacija motivisana dugotrajnim operacijama odsecanja

POBOLJŠANJA I NOVE VERZIJE SISTEMA DATOTEKA EXT3

■ **Odložena alokacija**

- posle upisa, samo se mapiraju blokovi i označavaju strane koje treba da se upišu, stvarni upis se dešava kasnije
- odložena alokacija odvodi ove operacije korak dalje tako što odlaže alokaciju novih blokova do **vremena za write-back**
- smanjuje se fragmentacija
- spaja alokacije više blokova u jednu efektivno smanjujući i broj ciklusa potrebnih za ponavljanje **get_block()** poziva
- izbegava se potreba za ažuriranjem diska kod kreiranja meta struktura za datoteke koje ne ostaju dugo na sistemu

■ Implementacija odložene alokacije **bazirane na extentima**

- Inicijalna procena ove tri **zacrpe(extenata, odložene alokacije i alokacije extenata)** pokazuje značajno povećanje propusne moći

SISTEM DATOTEKA EXT4

- Sistem datoteka ext4-*fourth extended file system*
- Andrew Morton objavio 2006. godine
- Priključen je kernelu 2.6.19
- Podržava extente i volumene do 1024PB
- Kompatibilan sa sistemom datoteka ext3

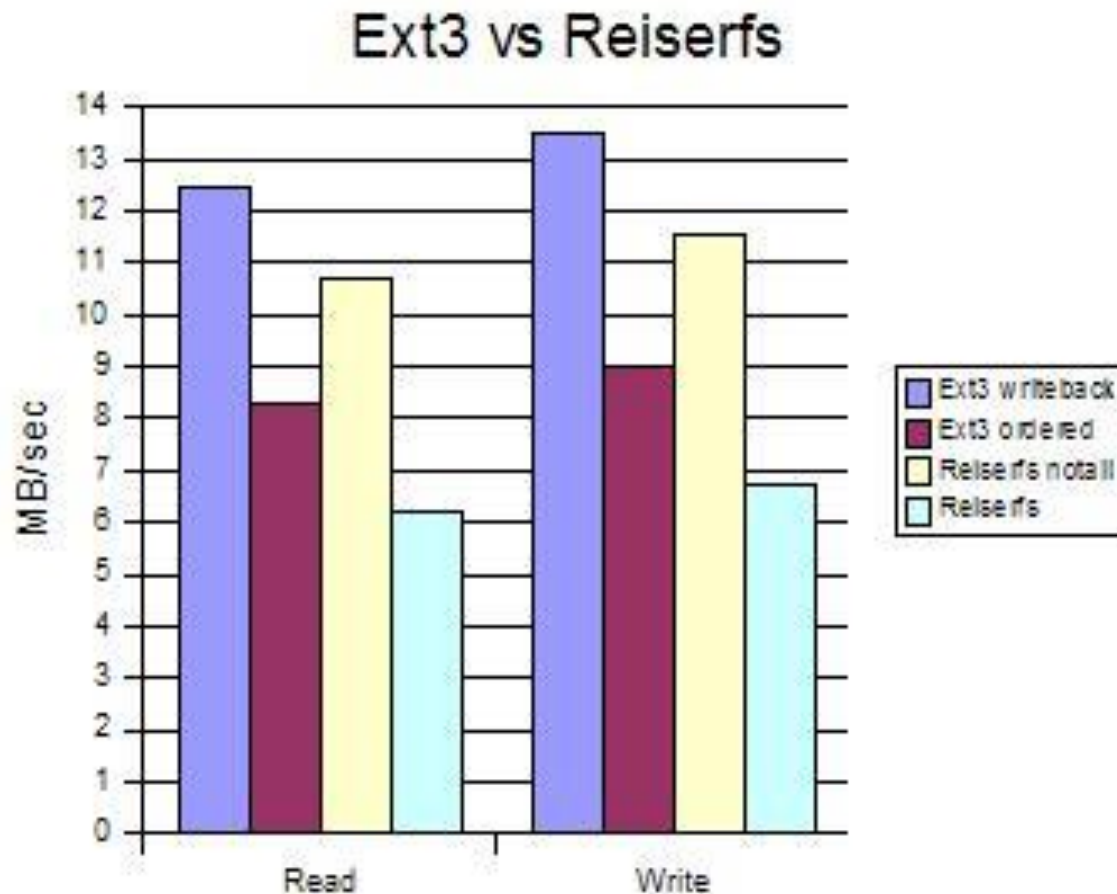
POREĐENJE SISTEMA DATOTEKA EXT3 SA DRUGIM SISTEMIMA DATOTEKA

- Performanse sistema datoteka ext3 i ReiserFS u radu sa malim datotekama



POREĐENJE SISTEMA DATOTEKA EXT3 SA DRUGIM SISTEMIMA DATOTEKA

- Performanse sistema datoteka ext3 i ReiserFS u radu sa malim datotekama



POREĐENJE SISTEMA DATOTEKA EXT3 SA DRUGIM SISTEMIMA DATOTEKA

- Svojstva sistema datoteka ext3 i drugih sistema datoteka

	Hard links	Soft links	Block journaling	Metadata only changes	Case sensitive	Case preserv.	File change log	Incremental snapshots	XIP	FS encryption
HPFS	No	No	No	No	No	Yes	No	?	No	No
FAT32	No	No	No	No	No	Partial	No	No	No	No
NTFS	Yes	Partial	No	Yes	Yes	Yes	Yes	Yes	?	Yes
ext2	Yes	Yes	No	No	Yes	Yes	No	No	Yes	No
ext3	Yes	Yes	Yes	Yes	Yes	Yes	No	No	?	No
Reiser4	Yes	Yes	Yes	No	Yes	Yes	No	?	?	Yes
XFS	Yes	Yes	No	Yes	Yes	Yes	Yes	No	?	No
JFS	Yes	Yes	No	Yes	Yes	Yes	No	?	?	No

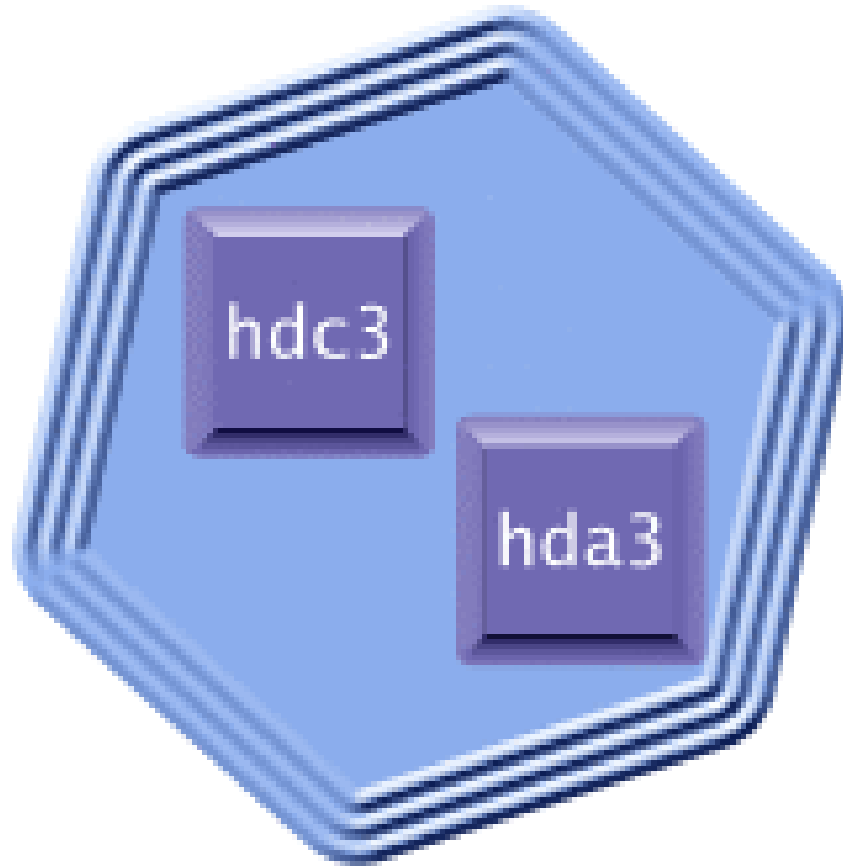
POREĐENJE SISTEMA DATOTEKA EXT3 SA DRUGIM SISTEMIMA DATOTEKA

- Razlike u meta strukturama sistema datoteka ext3 i drugih sistema datoteka

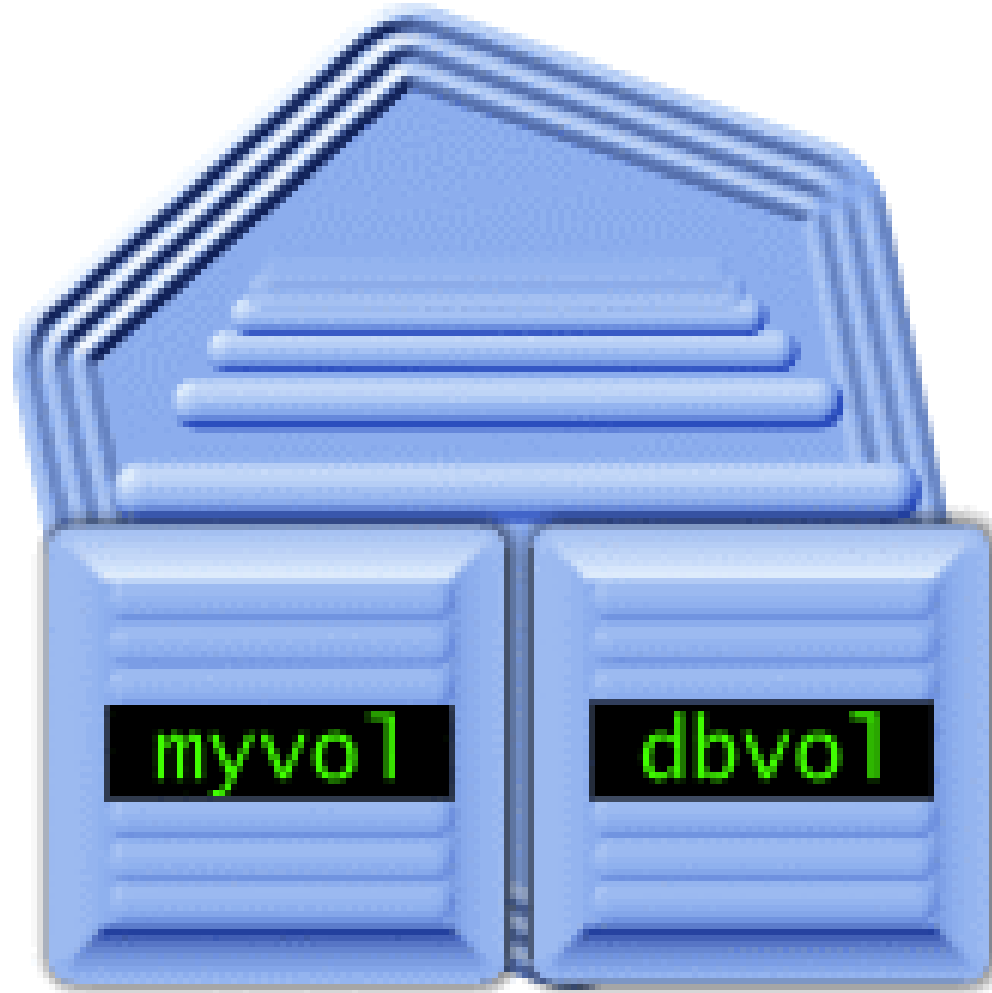
	Stores file owner	POSIX file permissions	Creation TS	Last access/ read TS	Last metadata change TS	Last archive TS	Access Control Lists	Extended attributes/ Alternate data streams/ forks
HPFS	Yes	No	Yes	Yes	No	No	No	Yes
FAT32	No	No	Yes	Yes	No	No	No	No
NTFS	Yes	No	Yes	Yes	Yes	No	Yes	Yes
ext2	Yes	Yes	No	Yes	Yes	No	Yes	Yes
ext3	Yes	Yes	No	Yes	Yes	No	Yes	Yes
Reiser4	Yes	Yes	No	Yes	Yes	No	No	No
XFS	Yes	Yes	No	Yes	Yes	No	Yes	Yes
JFS	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes

LVM – Volume Group

- Figure 1.1: A volume group is created out of physical volumes



Logical Group in Volume Group



Adding additional extents from our volume group, expanding the size of our logical volume

