JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

PEDAGOGICKÁ FAKULTA

KATEDRA FYZIKY

Bakalářská práce

Tomáš Náhlík

Ing. Michal Šerý

2006

SOUBOROVÉ SYSTÉMY

Poděkování:

Tímto bych chtěl poděkovat vedoucímu své bakalářské práce a také autorům materiálů, ze kterých při tvorbě této práce čerpal.

Prohlášení

Prohlašuji, že jsem tuto práci vypracoval samostatně a že jsem veškerou použitou literaturu uvedl v Seznamu literatury. Zároveň povoluji Katedře fyziky PF JU v Českých Budějovicích libovolné využití této práce.

Tomáš Náhlík

Obsah

1. Úvod – Obecně o souborových systémech

Co jsou to souborové systémy?

Je to software, který slouží k organizování a používání dat uložených na záznamových médiích (pevný disk, CD apod.). Souborový systém zajišťuje integritu. Tudíž informace, které uložíme, budeme moci později vyvolat v nezměněné podobě. Pro svou činnost file systém ukládá informace o souborech a informace o sobě samotném (vlastnictví, datum, kontrola přístupu, délka souboru a jeho lokace na disku apod.). Bez těchto informací, tzv. *metadata*, by souborový systém nemohl pracovat.

Mezi jeho hlavní funkce patří:

- sledování stavu informací
- strategie určování místa a způsobu uložení souborů a jejich přístupových práv
- přidělování informací procesům
- odnímání informací procesům

Symbolický souborový systém

Na úrovni symbolického systému souborů se používá jména souboru k jednoznačnému určení odpovídajícího zápisu v adresáři souborů. Jinak řečeno: Tak, jak vidíme soubory a adresáře v operačním systému my, uživatelé. Například pomocí nějakého správce souborů.

Logický souborový systém

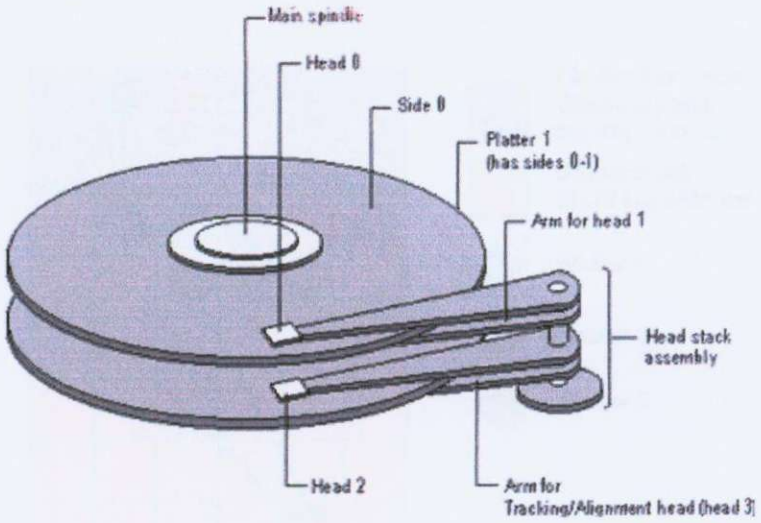
Logický systém souborů provádí transformaci struktury logických vět na lineární řetězec slabik, s nímž se pracuje na úrovni fyzického systému souborů. Používá sekvenční přístup (např. u páskových médií), nebo přímý přístup, kde se specifikuje požadovaný úsek (např. pevný disk).

Fyzický souborový systém

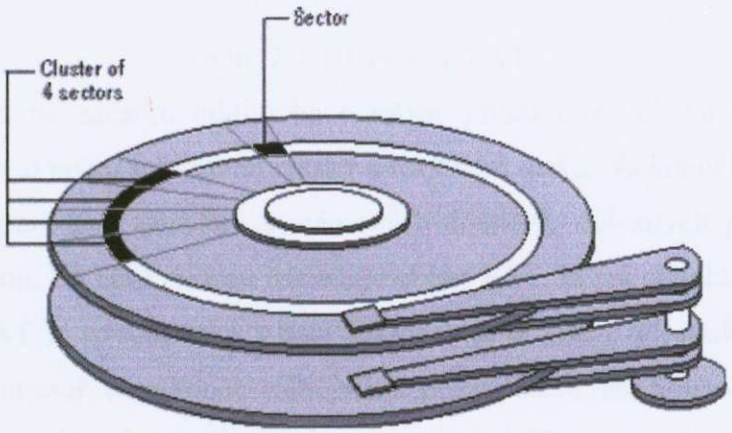
Fyzický systém souborů zajišťuje přidělování diskového prostoru souborům. Transformuje logické adresy na adresy fyzických bloků.

Jak vlastně takový pevný disk uvnitř vypadá a jak funguje?

Uvnitř harddisku se nachází jedna nebo více ploten. Plotna je jakýsi kotouč, na který se pomocí magnetické hlavičky provádí zápis dat. Každá plotna je rozdělena na stopy, což je systém soustředných kružnic, a na sektory, o velikosti 512B. Nejmenší adresovatelnou jednotkou na disku je cluster. Cluster je složen z 1 až 64 sektorů a jeho velikost se tedy pohybuje v rozmezí od 512B do 32kB.



Obr.1.1 Fyzická struktura disku



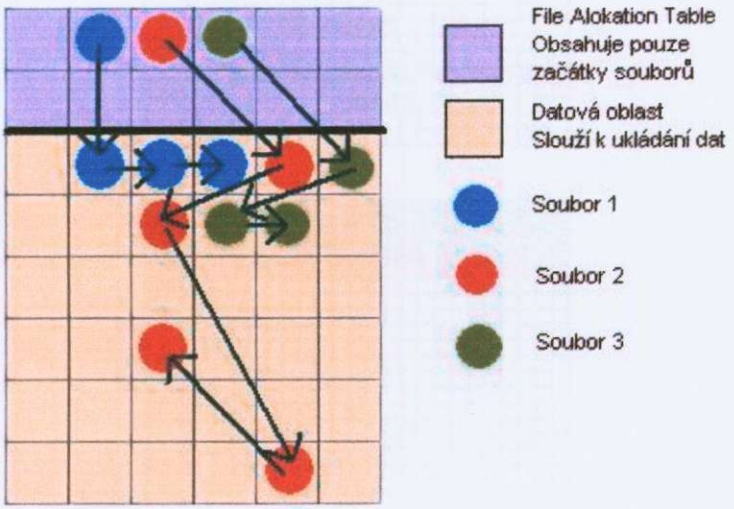
Obr.1.2 Rozdělení disku

2. Diskové souborové systémy

2.1 MSDOS

2.1.1 FAT 16 – File Allocation Table 16

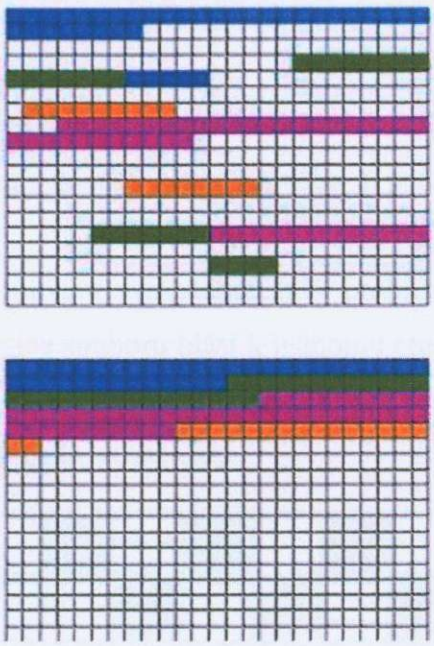
Každý souborový systém typu FAT (File Allocation Table) je založen na principu záznamů v tabulce.



Obr.2.1.1.1 Princip FAT

FAT 16 má na začátku oddílu boot sektor s nultou pozicí, ve kterém jsou uloženy informace o počtu sektorů na cluster a rozložení disku. Na konci boot sektoru je obslužný program pro zavádění systémových souborů, důležitých pro spuštění operačního systému. Po boot sektoru následuje oblast FAT, ta má obvykle dvě kopie. Délka a počet FAT je uveden také v boot sektoru. Za oblastí FAT následuje oblast root – kořenový adresář, a na konec volné místo pro ukládání dat. Souborový systém FAT16 používá pro zápis do tabulky 16bitový záznam. Jelikož největší možný cluster má velikost 32kB, je velikost oddílu FAT16 omezena na 2^{16} * 32kB, což jsou 2GB. Toto omezení lze vyřešit rozdělením větších disků na několik oddílů. Protože nejmenší jednotkou je cluster o velikosti 32kB, dochází na takovémto oddílu k plýtvání místem. Každý soubor menší než 32kB bude zapsán právě do jednoho clusteru - a tudíž jeho skutečná velikost na disku bude navýšena na 32kB. Toto ale není jediný problém souborových systémů založených na principu FAT. Časem zde dochází k takzvané fragmentaci. Fragmentace může být vnitřní, nebo vnější. Vnitřní fragmentace nastává v případě, že zůstávají nevyužity velké části alokačních bloků.

To znamená, že soubory s velikostí menší než 1 cluster zabírají více místa. Vnější fragmentace znamená, že jeden soubor je rozmístěn na mnoha místech pevného disku. Dochází tím ke zpomalení operací čtení/zápis. Proto je nutné po nějaké době užívání provádět defragmentaci (seřazení roztroušených částí jednotlivých souborů za sebe).



Obr.2.1.1.2 Defragmentace

Složka root

Tato složka se nachází za FAT. Její délka není jednoznačná a závisí na konkrétním disku. Kvůli možnosti využití stromové struktury adresářů se systém dívá na všechny složky, kromě složky root, jako na soubory. Každá položka má délku 32bytů. Prvních osm bytů je vyhrazeno pro jméno, přičemž první znak je rozhodující. Následuje 3bytová přípona. Dále je zde 1bytová položka vyjadřující atributy, a to v pořadí od nultého bitu:

0. bit ReadOnly – pouze pro čtení

1. bit Hidden – skrytý

2. bit System – systémový

3. bit ID Volume – položka obsahuje v prvních jedenácti bytech název disku. Kromě tohoto názvu, data a času zařízení neobsahuje tato položka žádné jiné informace. Položka tohoto typu může existovat pouze ve složce rootu a musí být na disku jediná.

4. bit SubDirectory – položka definuje označující element jako složku

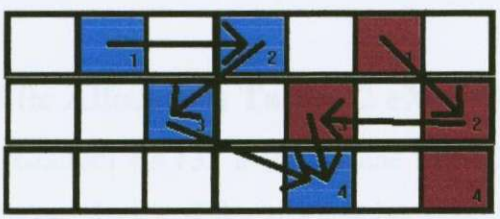
5. bit Archive – archivní bit, nastavuje se při modifikaci souboru

6. a 7. bit nejsou využity

Další dva byty nesou informaci o čase poslední změny souboru, další dva byty nesou informaci o datu poslední modifikace souboru, následují dva byty ukazující na číslo prvního clusteru souboru či adresáře, a poslední je čtyřbytová položka s informací o velikosti souboru, pro složku je zde 0. Z toho vyplývá, že maximální velikost souboru je 4GB.

Každá složka mimo rootu obsahuje na svém začátku dvojici položek „ . “ a „ .. “. Pomocí položky „ . “ ukazuje složka sama na sebe a položka „ .. “ ukazuje na rodičovskou složku.

U oddílů typu FAT se mohou vyskytnout další problémy jako je „překřížení“ souborů. To znamená, že se více souborů hlásí k jednomu clusteru. Tento problém se řeší zkrácením překřížených souborů. Překřížené soubory se zkrátí před společným clusterem, ale tím většinou dojde ke ztrátě dat.



Obr.2.1.1.3 Překřížení souborů

Pro rychlejší práci na diskovém oddílu se systémem FAT16 se používá cachování FAT tabulky. Protože data se čtou po celém povrchu disku a umístění jednotlivých clusterů je nutno číst z FAT tabulky, dochází tak k velmi neefektivním pohybům čtecích hlav. Princip cachování je jednoduchý, jedná se o to, že část nebo celá FAT tabulka se dopředu načte do operační paměti, tzv. RAM (Random Access Memory – operační paměť počítače), a odtud se následně čte pozice jednotlivých clusterů, aniž by bylo nutné neustále přejíždět čtecí hlavou nad FAT tabulku. V operačním systému MS DOS tuto funkci plnil program SMARTDRV.EXE. Ve WINDOWS 9x je již součástí systému.

2.2 Windows

2.2.1 FAT 32 – File Allocation Table 32

Rozdíl mezi FAT16 a FAT32 je především v tom, že FAT32 používá 32bitový záznam pro zápis do tabulky. Což znamená 4 294 967 296 (4G) záznamů, ale nepoužívá se celý rozsah, protože FAT tabulka by zabírala příliš mnoho místa na pevném disku. Díky tomu bylo omezení velikosti jedné partition (diskového oddílu) změněno ze 2GB na 2TB. Další rozdíl je v tom, že umožňuje neomezený počet záznamů v kořenovém adresáři, tzv. rootu. Je zde ještě jeden podstatný rozdíl, který se týká bezpečnosti. Jde o to, že FAT32 používá dvě FAT tabulky, jednu hlavní a jednu záložní. Díky tomuto vylepšení nedochází k takové nekonzistenci souborů jako u FAT16. Navíc FAT32 je robustnější oproti FAT16 a může přemístit složku rootu kamkoli na disk, protože složka rootu je reprezentována jako běžný řetězec clusterů. FAT32 již není přístupná z DOSu a ani z některých verzí Windows 95.

2.2.2 FAT 32x – File Allocation Table 32 eXtension

Existuje ještě jedno rozšíření FAT32 a je nazvané FAT32x (eXtension). Je to soubor extenzí umožňující zacházet s disky o kapacitě větší než 8.4GB. Původní specifikace daná konvencí CHS - Cylindetr/Head/Sector - omezovala kapacitu stanovením limitů na adresovatelné maximum 1024 cylindrů, 255 hlaviček a 63 sektorů. Toto adresovaní dodnes používá jak Win 3x, tak i DOS. Zmíněná extenze FAT32 slouží k tomu, aby byl celý disk plně adresovatelný lineární metodou LBA(Logical Block Addressing). Jedná se o správu velkých disků. Princip tohoto adresování spočívá v tom, že pokud se přesáhne maximální počet sektorů, následující sektory se adresují, jako by byly na jiné plotně, tzn. zvýší se číslo hlavičky.

2.2.3 NTFS – New Technology File System

Souborový systém NTFS je z dílen firmy Microsoft a patří mezi žurnálovací souborové systémy. To znamená, že takovýto systém neukládá pouze holá data, ale také informace o probíhajících operacích. Zvyšuje se tím bezpečnost konzistence dat, protože i při násilném přerušení operací s daty na disku je lze ve většině případů obnovit. Systém NTFS je velmi starý, podporoval jej už operační systém Windows NT 3.1 a vyšší. Dále je podporován operačním systémem Windows 2000 a Windows XP. V ostatních operačních systémech není plně podporován, protože Microsoft

nezveřejnil vnitřní uspořádání tohoto souborového systému. Existují sice různé podpůrné programy, ale jejich spolehlivost není zaručena. Data z tohoto souborového systému je možné číst i pod různými verzemi Linuxu, ale nelze je na disk s tímto souborovým systémem spolehlivě zapisovat. Existuje jisté riziko, že při pokusu o zápis na takovýto oddíl dojde ke ztrátě dat na něm uložených.

Maximální teoretická velikost dosahuje hodnoty 16 384TB, prakticky ale mohou mít oddíly pouze 4TB. Podporuje velikosti clusteru od 0,5kB do 64kB, standardní velikost je 4kB. Oddíl naformátovaný systémem NTFS je symbolicky rozdělen na dvě části. Prvních 12% je vyhrazeno pro soubor MFT (Master File Table). Zapisovat data do tohoto místa není možné. Oblast MFT je vždy zachována prázdná, aby nejdůležitější servisní soubor (MFT) nebyl při růstu fragmentován. Zbývajících 88 % disku již představuje použitelné místo pro ukládání dat. Protože soubor MFT je nejdůležitějším souborem tohoto file systému, je uprostřed disku umístěna kopie jeho prvních 16 záznamů. Ačkoli část pro soubory obsahuje veškeré fyzicky volné místo, je v něm také začleněna část prostoru MFT. Mechanismus používání MFT je následující: když už soubory není možno zapsat do použitelného prostoru, oblast MFT se jednoduše zredukuje, aby uvolnila místo pro zapisované soubory. Při uvolnění použitelného místa se MFT může zase rozšířit. Tak je možné pro běžné soubory, aby zůstaly v tomto prostoru, a je to normální. Systém zkouší zachovat tuto část volnou, avšak občas selhává. Přesto metasoubor MFT může být fragmentovaný, i když to třeba není nežádoucí.

Struktura MFT

V souborovém systému NTFS je všechno soubor – dokonce i systémové informace. Soubor MFT - Master File Table (hlavní tabulka souborů) představuje centralizovanou složku všech zbývající souborů včetně sebe. MFT je rozdělen na záznamy pevné délky, obvykle 1kB, každý z nich odpovídá nějakému souboru. Prvních 16 souborů je určeno pro vlastní potřeby souborového systému a operační systém do nich nemá přístup. Nazývají se METASOUBORY a prvním z nich je sám MFT soubor. Těchto 16 záznamů má na disku pevné umístění. Zbytek MFT souboru může být umístěn kdekoli na disku jako ostatní soubory a je možné obnovit jeho pozici za pomoci sebe sama s použitím jeho základní polohy prvního MFT elementu.

Prvních 16 metasouborů patří mezi systémové a každý z nich je zodpovědný za určité operace. Výhodou takovéhoto rozdělení je velká flexibilita. Metasoubory

systému NTFS jsou umístěny v rootu a jejich název začíná znakem „$“, proto je obtížné získat z nich nějaké informace za pomoci běžných prostředků.

Název souboru	**Obsah souboru**
$MFT	Sám MFT
$MFTmirr	Kopie prvních 16 záznamů MFT umístěná uprostřed disku
$LogFile	Soubor pro protokolování
$Volume	Interní informace – Název oddílu, verze systému souborů, atd.
$AttrDef	Soupis standardních atributů souborů na oddílu
$.	Složka root
$Bitmap	Bitová mapa volného místa oddílu
$Boot	Boot sektor
$Quota	Práva uživatele na užití disku (od NT5)
$Upcase	Soubor tabulky shody velkých a malých písmen názvů souborů na daném oddílu

Tabulka 2.2.3.1 Přehled funkcí metasouborů systému NTFS

Soubory a toky

První nucenou částí je záznam v MFT. Jak bylo již zmíněno, v MFT jsou uvedeny všechny soubory disku. Všechny informace o souboru - až na samotná data souboru - jsou zde uložena: název souboru, jeho velikost, pozice fragmentů na disku atd. Pokud jeden záznam v MFT na informace nestačí, pak je použito několik záznamů nezávisle jeden na druhém. Nepovinnou součástí jsou toky dat souborů.

1) soubor neobsahuje žádná data - a potom se pro něj nepoužije volné místo.

2) soubor není příliš velký - a potom se učiní velice zdařilé rozhodnutí: data souboru jsou uložena právě v MFT v místě mimo hlavní data s omezením na jeden záznam MFT. Soubor o velikosti stovek bytů zpravidla nemá fyzický obraz v základní souborové oblasti. Všechna taková data souborů jsou uložena právě na jednom místě - v MFT.

Je zde jeden zajímavý případ s daty souboru. Každý soubor na NTFS má dost abstraktní pojetí - nemá žádná data, má jen toky. Jeden z toků má pro nás zcela obvyklý smysl - data soboru. Ale většina atributů souboru jsou také toky. Takže nemáme než základní příznak souboru, kterým je jen číslo v MFT a zbytek je

nepovinný. Tato abstrakce může být velice vhodně využita - např. můžeme uložit k souboru ještě další tok, i když v něm máme nějaká data uložena, třeba informaci o autorovi a obsahu souboru, jak je tomu využito ve Windows 2000. Je zajímavé, že tyto dodatečné toky nejsou viditelné s použitím běžných prostředků: viditelná velikost souboru je jen velikost hlavního toku obsahujícího tradiční data. Tím může nastat třeba situace, že smazáním souboru s nulovou délkou se uvolní 1 GB, protože některý program využívající přídavných toků si do nich uložil informaci o velikosti 1 GB. Momentálně tyto toky nejsou prakticky využity, takže nemusíme mít strach z takové situace, ačkoli je hypoteticky možná. Mějme však na paměti, že soubor na NTFS je mnohem hlouběji a komplexněji pojat, než je možné si představit právě při pohledu na složky na disku. Jméno souboru může sestávat z jakýchkoli znaků včetně celé národní abecedy, protože data jsou zastoupena sadou Unicode - 16bitovým podáním umožňujícím 65535 různých znaků. Maximální délka názvu souboru je 255 znaků.

Složky v NTFS

Složka na NTFS je specifický soubor, který obsahuje odkazy na ostatní vnořené soubory a složky založené na stromové struktuře dat disku. Soubor složky je rozdělen na bloky, z nichž každý obsahuje název souboru, základní atributy a odkaz na element MFT, který podává kompletní informace o elementu složky. Vnitřní struktura složky je binární strom. Binární strom je strom, jehož každý uzel obsahuje nejvýše dva siny a zároveň jeho výška je minimálně rovna $\log_2$(počet externích uzlů) a maximálně rovna počtu interních uzlů. Takováto vnitřní struktura usnadňuje prohledávání dané složky. U systému FAT by bylo nutné projít všechny soubory složky, kdežto u NTFS jde pouze o průchod určité větve binárního stromu.

Např. pro nalezení 1 souboru mezi 1000 je nutné u FAT provést 500 porovnání (podle teorie pravděpodobnosti se hledaný soubor nachází uprostřed), kdežto u systému založeném na struktuře stromu je to něco okolo 10 porovnání.

Proti stromové struktuře naopak svědčí fakt, že údržba takovéto struktury je mnohem náročnější než údržba lineární struktury.

Protokolování NTFS

Protokolování - nebo také žurnálování - je jednou z předností systému NTFS. Jde o to, že tento systém jako většina moderních systémů, je založen na principu transakcí. Každá akce se provede celá a správně, nebo se neprovede vůbec.

Příklad 2.2.3.1 Obnova systému po přerušení zápisu

Provádí se zápis dat na disk. Náhle je uprostřed zápisu zjištěno, že na dané místo nelze zapisovat, např. kvůli poškozenému povrchu. Akce se stornuje, vadné místo se označí a spustí se nová transakce zápisu.

Může nastat ještě komplikovanější případ. V průběhu zápisu dat na disk vypadne napájení. Po opětovném startu systém načte metasoubor $LogFile. Metasoubor $LogFile je takzvaným deníkem transakcí. Je to soubor, do něhož se zapisují žádosti o transakce a jejich vyřízení. Pokud nebyla transakce dokončena, systém tuto transakci stornuje a označí místo použité k zápisu části dat jako volné. Pokud dojde ke zhroucení v průběhu zápisu do deníku transakcí, nic se nestane. Znamená to, že buď se systém pokoušel zapsat žádost o vytvoření transakce, nebo je transakce hotová a systém zapisoval informace o jejím dokončení.

Komprese NTFS

Na oddílech naformátovaných systémem NTFS mají soubory ještě jeden atribut navíc, je jím atribut komprese. To znamená, že NTFS podporuje kompresi souborů na diskovém oddílu - a to přímo v průběhu operací zápisu dat na disk. Tato komprese je velice rychlá - a proto nikterak neomezuje běh aplikací, které přistupují k datům na diskovém oddílu, na kterém je zapnuta komprese. Na jiných oddílech bylo nutné použít nějaký systémový nástroj. U NTFS je tento nástroj implementován přímo do souborového systému. Tato komprese má pouze jednu nepříjemnou vlastnost, a to velmi vysokou virtuální fragmentaci. Komprese se provádí ze 16 clusterů a používá virtuální clustery. To je extrémně flexibilní řešení a při jeho použití dochází k zajímavým důsledkům, např.: lze zkomprimovat pouze jednu polovinu souboru a druhou ponechat bez komprimace. Dosahuje se toho tím, že informace o uložení a stupni komprese daného fragmentu je podobná běžné fragmentaci.

Příklad 2.2.3.2 Příklad komprese souboru

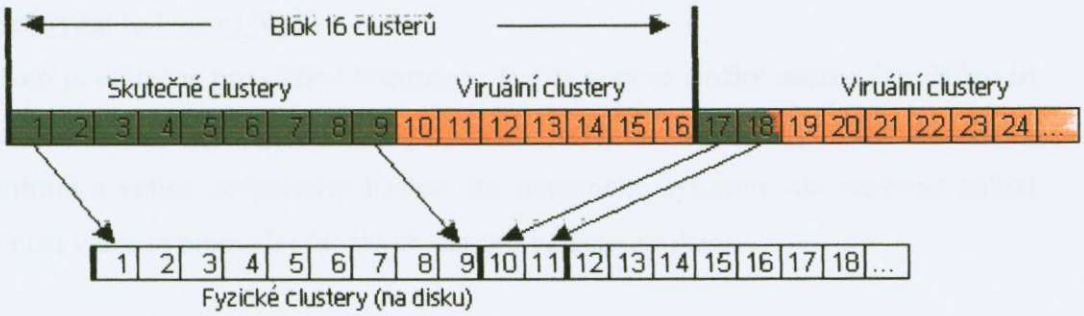
Skutečný záznam fyzického rozložení pro skutečný, nekomprimovaný soubor:

- Clustery souboru 1 až 43 jsou uloženy v clusterech disku od 400.
- Clustery souboru 44 až 52 jsou uloženy v clusterech disku od 8530...

Fyzická rozloha typického komprimovaného souboru:

- Clustery souboru 1 až 9 jsou uloženy v clusterech disku od 400.
- Clustery souboru 10 až 16 nejsou uloženy nikde.

- Clustery souboru 17 až 18 jsou uloženy v clusterech disku od 409.
- Clustery souboru 19 až 36 nejsou uloženy nikde.



Obr.2.2.3.1 Komprese souboru

Je vidět, že komprimovaný soubor má virtuální clustery, které nenesou skutečnou informaci. Jakmile systém spatří takové virtuální clustery, hned si uvědomí, že data v předchozím bloku neobsahují 16 clusterů a dekomprimuje tak data, aby vyplnil virtuální clustery, a to je celý algoritmus.

Zabezpečení NTFS

Systém NTFS má mnoho prostředků pro rozlišování práv objektů. Práva systému NTFS jsou úzce spojena se samotným systémem, to znamená, že nemusí být uváděna v případě fyzického přístupu na disk k datům z jiného operačního systému.

Pevné odkazy (hard links) a Symbolické odkazy (symbolic links) (NT5)

Pevný odkaz vzniká, když stejný soubor má dva názvy (některé záznamy složky ukazují na stejný záznam v MFT). Připusťme, že stejný soubor má názvy 1.txt a 2.txt: pokud uživatel smaže soubor 1, soubor 2 zůstane. Pokud smaže soubor 2, soubor 1 zůstane. To znamená, že oba názvy jsou úplně totožné od okamžiku vytvoření. Soubor je fyzicky smazán pouze, pokud je jeho poslední jméno smazáno.

Zde je mnohem více praktických možností umožňujících vytvořit virtuální složky, mnohem více než virtuální disky - použitím příkazu *subst* v DOSu. Aplikace jsou dostatečně široké: nejprve je to zjednodušení systému složek. Když se vám nelíbí název složky *Documents and settings\Administrator\Documents*, můžete na něj vytvořit odkaz do rootu a systém bude komunikovat se složkou přes původní cestu, ale vy - s mnohem kratším názvem úplně ekvivalentním k ní. Odkaz se dá zrušit

příkazem *rm*. Při smazání odkazu Průzkumníkem či jiným programem se však smaže i obsah toho, na co odkaz ukazuje!

Šifrování (od verze NT5)

Toto je užitečné pro citlivé informace, každý soubor složky může být zašifrován - a pak není možné, aby ho někdo přečetl jinou instalací NT. V kombinaci se standardním a velice bezpečným heslem do samotného systému tato možnost nabízí bezpečnost vámi vybraných důležitých dat pro většinu aplikací.

Kvóty (od verze NT5)

Kvóty jsou další vlastností, kterou přinesla NTFS. Pomocí této funkce může administrátor přidělit každému uživateli určitý diskový prostor. To znamená, kolik může daný uživatel zapsat dat na disk. Toto lze nastavit také pro jednotlivé složky. Je to výhodné pro servery nebo stanice s přístupem více lidí. Tato vlastnost se používá stejně jako serverů Novell Netware.

Typy svazků (volumes) NTFS

Systém NTFS umožňuje vytvářet několik typů svazků

1. Jednoduché svazky (simple volumes)

Jde o svazky vytvořené na jednom fyzickém disku přes jeden nebo libovolný počet oddílů NTFS, které spojíme dohromady.

2. Rozložené svazky (spanned volumes)

Pokud jednoduchý svazek rozšíříme přes několik fyzických disků, stává se jednoduchý svazek rozloženým svazkem. U dynamických disků je možné připojit do již existujícího svazku další disk a rozšířit tak místo kdykoliv. Ve Windows NT 4 a starších byly známy pod názvem volume set.

3. Prokládané svazky (striped volumes)

Data takového svazku jsou ukládána střídavě na dva nebo více fyzických disků. Data těchto svazků jsou alokována střídavě a rovnoměrně na disky prokládaného svazku. Tyto svazky značně urychlují přístup k harddisku. Ve Windows NT 4 a starších znám jako stripe set.

4. Zrcadlené svazky (mirrored volumes)

Jde o svazky vylučující chyby, které ukládají data ve dvou kopiích na dva fyzické disky. Poskytují zálohu dat tím, že používají kopii (zrcadlení) svazku, aby duplikovaly informace uložené na svazcích. Zrcadlo je uloženo na jiný fyzický disk. Pokud data na jednom disku začnou vykazovat chyby a stanou se nepoužitelnými, systém pokračuje v operaci a používá nepoškozená data druhého disku. Zrcadlený svazek je pomalejší než svazek RAID-5 při operacích čtení, ale je rychlejší při operacích zápisu. Ve Windows NT 4 a starších byl označován jako mirror set.

5. Svazky typu RAID-5

Jedná se o svazky, které k vyloučení chyb dat a parity prokládají obsah přes tři nebo více fyzických disků. Když zhavaruje část fyzického disku, je možné znovu vytvořit data, která byla na zhavarované části ze zbývajících dat a parity. Svazky RAID-5 jsou dobrým řešením pro prostředí s počítači, kde je vysoká aktivita čtení dat. Ve Windows NT 4 a starších byl znám pod jménem stripe set with parity

NTFS je vysoce výkonný file systém pro jakýkoliv disk. Minimální doporučená velikosti oddílu je 400 MB. Není problém jej však použít na oddíly menší. Můžeme jím naformátovat také vyměnitelná média, jako je např. ZIP. Výhodou je, že si lze zvolit velikost clusterů. Na NTFS lze také aktivovat účinnou a rychlou kompresi nebo šifrování. Pozor je nutno dát při šifrování, protože je potom médium možné přečíst jen instalací Windows, pod kterou byly soubory zašifrovány, a uživatelem, který je zašifroval. Diskety není možné běžnými prostředky naformátovat jako NTFS.

2.2.4 WinFS

WinFS je systém pro ukládání dat od firmy Microsoft. Není to jenom souborový systém, ale také nástroj pro organizaci, vyhledávání a sdílení velkého množství dat a informací. Jeho základem je souborový sytém NTFS, ke kterému je přidána řada věcí pro zlepšení výkonu. Je to především široká podpora API (Aplication Programming Interface – rozhraní pro programování aplikací) a XML. Data na disku mohou být strukturována do podoby xml souborů buď úplně, částečně, nebo lze uchovávat data i nestrukturovaná. Díky tomu se WinFS nachází na pomezí mezi klasickým a databázovým souborovým systémem.

Na rozdíl od NTFS WinFS nepoužívá k ukládání stromovou strukturu, ale DAG (Direct acyclic graph). DAG je soubor objektů a jejich vztahů fyzicky uložený v relační databázi, která umožňuje uložit jakoukoli hierarchickou strukturu. To umožňuje vyhledávat objekty pomocí jejich vlastností a hodnot těchto vlastností.

Datový model ve WinFS používá pro popis datových struktur typy a podtypy, vlastnosti a položky, relace, omezení a rozšiřitelnost.

Každý objekt uložený ve WinFS musí být nějakého typu. Všechny WinFS typy se skládají z vlastností a položek, jež mohou být buď skalární typy, nebo vnořené typy (Nested Element Type), jako jsou struktury a záznamy (structure, records). Ve WinFS obsahuje mechanismy pro vytvoření nových typů. To lze udělat definováním úplně nového typu, nebo rozšířením typu již existujícího.

WinFS má koncept pro správu relací mezi objekty. Každá relace je identifikována jako zdrojová, nebo cílová. Pokud nějaký objekt závisí pomocí relací na jiném, jeho existence je podmíněna tímto objektem. Jestliže je smazána zdrojová instance, smaže se celá relace. Pokud objekt existuje bez objektu, na kterém závisí, takováto relace se nazývá volná (dangling). Ve Win FS existují dva typy relací, holding a reference. Relace typu holding nemůže být volná. Nejznámější holding relací je relace FolderMember (člen složky), cílem je soubor a zdrojem je složka. Ve WinFS může jeden soubor náležet několika složkám. Soubor je smazán až tehdy, jsou-li smazány všechny relace FolderMember tohoto souboru.

Rozšiřitelnost spočívá v možnosti definovat nové typy.

2.3 UNIX - LINUX

2.3.1 MINIX FS

Minix file systém je první linuxový souborový systém. Implementoval jej tvůrce Linuxu Linus Torvalds. Je to jednoduchý souborový systém založený na Unix V7 FS, používá bitové mapy místo zřetězených seznamů. Velikost oddílu, který lze tímto systémem naformátovat, je omezena na 64MB. Podporuje maximálně 14 znaková jména.

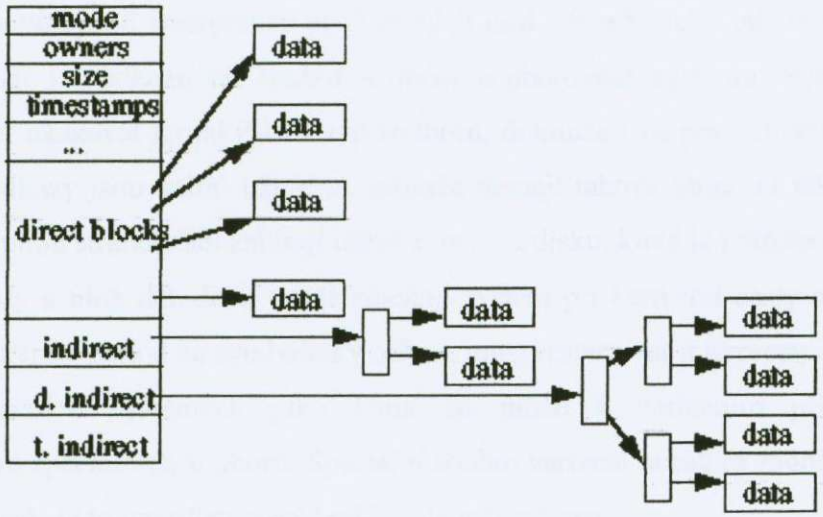
2.3.2 XIA FS

Tento souborový systém navrhl a implementoval Frank Xia. Je to jednoduchý a výkonný souborový systém. Je založen na kódu minix fs a byl uvolněn v lednu 1993. Pro správu volných bloků používá bitmapy místo zřetězených seznamů, z čehož vyplývá, že je výkonnější než extended fs.

2.3.3 EXT FS – Extended File Sytem

Toto je druhý linuxový souborový systém, který je založen na kódu Minix FS. Byl uvolněn v dubnu 1992. Implementoval jej Rémy Card. Oproti minix fs má menší omezení, ale nemá příliš dobře navržený space management. To znamená, že volné bloky jsou spravovány pomocí seznamů. Je pomalejší než minix fs a vyskytuje se zde velká fragmentace.

Základní strukturou souborového systému EXT je inoda. Každý soubor je reprezentován jednou inodou. Každá inoda obsahuje popis souboru a ukazatel na blok dat. V popisu souboru je zaznamenán typ souboru, práva, vlastník, čas a velikost souboru. Ukazatel na blok dat obsahuje adresy bloků dat, kde je daný soubor alokován.



Obr. 2.3.3.1 Inody

Složky jsou uspořádány do hierarchické stromové struktury. Každá složka může obsahovat soubory nebo další složky. Složky jsou zde reprezentovány pomocí speciálních souborů. Složka je vlastně soubor, který obsahuje odkaz na jiné soubory či složky. Takovéto odkazy jsou zde zastoupeny seznamem záznamů. Každý záznam obsahuje číslo inody a název souboru. Jestliže proces použije název cesty, kód kernelu (jádra systému) převede název cesty na číslo inody a začne postupně procházet tabulku inod, aby našel číslo odpovídající požadované cestě. Poté se inoda načte z paměti a použije se jako subsekvenční požadavek.

Souborový systém Unixu používá koncepci odkazů. Několik jmen může ukazovat na stejnou inodu. Inoda obsahuje položku, ve které je uloženo číslo, přiřazenou k souboru. Přidání odkazu na soubor spočívá ve vytvoření záznamu složky, kde číslo inody ukazuje na inodu, a v přidání počtu odkazů do inody. Odkaz se odstraní použitím příkazu *rm*. Pokud je tento příkaz použit, dojde ke snížení počtu odkazů a bude odstraněn soubor s atributem *link*. Pokud počet odkazů dosáhne nuly, dojde k uvolnění inody a tím úplnému vymazání daného souboru. Takovémuto odkazu se říká pevný odkaz (hard link) a nelze jej použít napříč několika systémy souborů, protože obsahuje odkaz na inodu. Pevný odkaz lze použít pouze pro soubory, aby se zamezilo cyklu ve stromu složek.

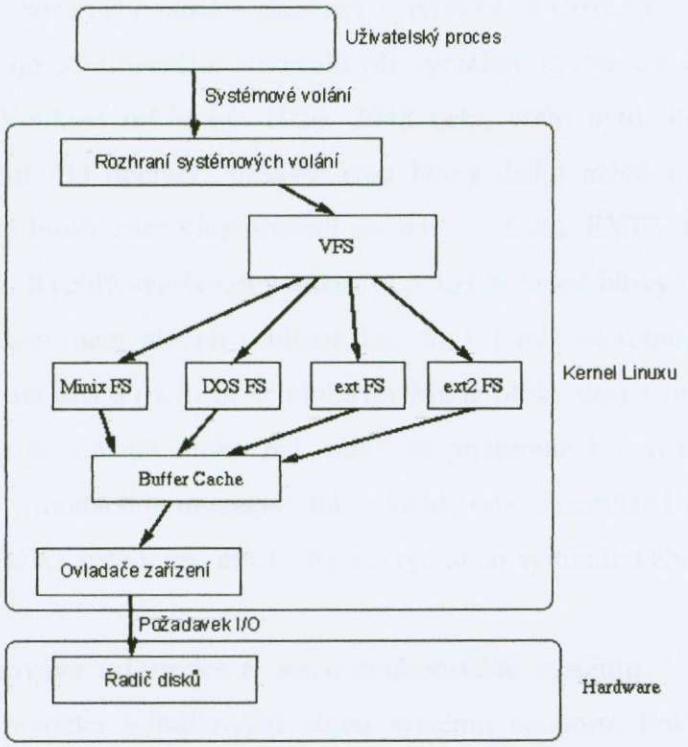
Dále existují ještě symbolické odkazy (symbolic links). Jsou to jednoduché soubory, které obsahují název souboru. Když jádro systému narazí na symbolický odkaz v průběhu konverze cesty na číslo inody, dojde k nahrazení názvu odkazu jeho

obsahem a znovu spustí interpretaci souborových cest. Symbolický odkaz na inody lze totiž použít i k odkazu na soubor v jiném souborovém systému. Symbolické odkazy mohou ukazovat na jakýkoliv typ souboru, dokonce i na neexistující soubor. Symbolické odkazy jsou velmi užitečné, protože nemají taková omezení jako pevné odkazy. Na druhou stranu však zabírají určité místo na disku, které je potřeba alokovat pro jejich inody a blok dat. Ještě navíc zatěžují systém při konverzi cesty na inodu. V případě, že kernel narazí na symbolický odkaz, musí restartovat interpretaci jména.

V operačních systémech jako Unix se může k zařízením přistupovat prostřednictvím speciálních souborů. Speciální soubor zařízení nezabírá žádné místo v systému souborů. Je to jen přístupový bod k ovladači zařízení.

Existují dva druhy speciálních souborů. Jsou to znakové soubory a soubory speciálních bloků. Znakové soubory umožňují vstupně/výstupní (I/O) operace ve znakovém režimu, zatímco soubory speciálních bloků potřebují, aby data byla zapsána v blokovém módu pomocí funkce vyrovnávací mezipaměťi (cache) bufferu. Na speciální soubor se odkazuje hlavním číslem, které identifikuje druh zařízení, a vedlejším číslem, které identifikuje jednotku.

Kernel Linuxu obsahuje vrstvu - virtuální systém souborů (Virtual File System - VFS), který se používá v průběhu operování se soubory. Byl vytvořen proto, aby bylo možné používat Linux na více souborových systémech. Dochází tak k nezávislosti operačního systému na systému souborů. VFS je vrstva, která zpracovává systémová volání, která se orientují na soubory. Volá potřebné funkce kódu fyzického systému souborů, aby provedly I/O. Když proces vydá souborově orientovaný požadavek na systém, kernel zavolá funkci obsaženou ve VFS. Tato funkce ovládá strukturu nezávislých manipulací a přesměruje volání na funkci obsaženou v kódu fyzického systému souborů, který je odpovědný za zpracovávání strukturálně nezávislých operací. Kód systému souborů užívá funkce vyrovnávací mezipaměti bufferu na požadavek I/O na zařízení.



Obr. 2.3.3.2 Struktura vrstev file systémů v linuxu

2.3.4 EXT2 FS – Second Extended File System

Second Extended File System navrhnul Rémy Card spolu s Waynem Davisonem, implementoval jej Rémy Card. Byl uvolněn také v lednu 1993 spolu s xia fs. Jeho implementace je založena na kódu ext fs. Spolu s Rémy Cardem na něm pracovala řada dalších programátorů, kteří jej vylepšili. Stephen Tweedie napsal kód kernelu, Theodore Ts'o napsal řadu utilit.

EXT2 byl vytvořen tak, aby podporoval standardní typy souborů Unixu. Jsou to běžné datové soubory, složky, symbolické odkazy a speciální soubory zařízení. Dokáže pracovat na velkých diskových oddílech až do velikosti 4TB. To je umožněno vrstvou VFS (Virtual File System). EXT2 dále nabízí použití dlouhých názvů souborů, až 255 znaků. Tento limit je možné zvýšit až na 1012 znaků, pokud by bylo potřeba. Tento souborový systém rezervuje některé bloky pro superuživatele. Běžně je rezervováno 5% bloků. Tato vlastnost umožňuje administrátorovi jednoduše zotavit systém ze situací, kdy běžní uživatelé svými procesy zaplnili systém souborů.

Oproti standardnímu Unixu EXT2 podporuje některá rozšíření, která nejsou běžně v Unixových souborových systémech. Souborové atributy dovolují uživateli modifikovat chování jádra systému, když manipulují se sadou souborů. Kdokoli může

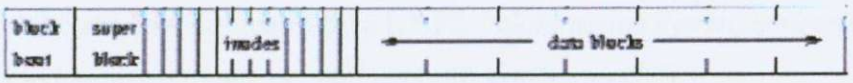
nastavit atributy souboru nebo složky. Soubory vytvořené ve složce zdědí její atributy. EXT2 také umožňuje administrátorovi zvolit při vytváření systému souborů velikost logických bloků. Velikost může být 1024, 2048 nebo 4096 bytů. Použití velkých bloků může urychlit I/O operace, protože jsou hlavy disku méně vytěžovány. Na druhou stranu velké bloky zase více plýtvají místem na disku. EXT2 realizuje rychlé symbolické odkazy. Rychlý symbolický odkaz nepoužívá žádné bloky dat na systému souborů. Cílový název není uložen v bloku dat, ale v inodě samotné. Tato politika může ušetřit místo na disku (nemusí se alokovat žádné bloky dat) a urychlit operace odkazů (není třeba již načítat bloky dat, když se přistupuje k takovému odkazu). Samozřejmě místo v inodách je omezeno, takže každý odkaz nemůže být realizován v inodě. Maximální délka názvu pro cílové jméno rychlého symbolického odkazu je 60 znaků.

EXT2 si uchovává informace o stavu souborového systému. V superbloku je použita speciální položka k indikování stavu systému souborů. Pokud se systém souborů připojí v módu Read/Write, je tato položka nastavena na „Not Clean". Pokud je systém připojen v módu Read-Only, je položka nastavena na „Clean".

Při bootování operační systém použije tuto informaci k tomu, zda se má daný souborový systém ověřovat. Jádro operačního systému do této položky také zaznamenává chyby. Jakmile je nějaké neshoda, je systém označen jako „Erroneous". Při ověřování systému se zjišťuje obsah této položky a podle její hodnoty se provádí ověřování systému souborů, případně se opravují chyby. Každé vynechání kontroly systému souborů je nebezpečné. Systém EXT2 poskytuje dva způsoby spouštění ověřování a kontroly v pravidelných intervalech. Čítač počtu připojení souborového systému se udržuje v superbloku a po každém připojení je jeho hodnota inkrementována. Pokud počet připojení dosáhne určitého počtu, spustí se kontrola i v případě, že je systém označen jako „Clean". V superbloku se udržuje také čas posledního ověřování. Tyto položky superbloku dovolují administrátorovi, aby systém pravidelně spouštěl kontrolu systému souborů. K nastavení intervalů pravidelné kontroly a různých nastavení systému EXT2 slouží program tune2fs.

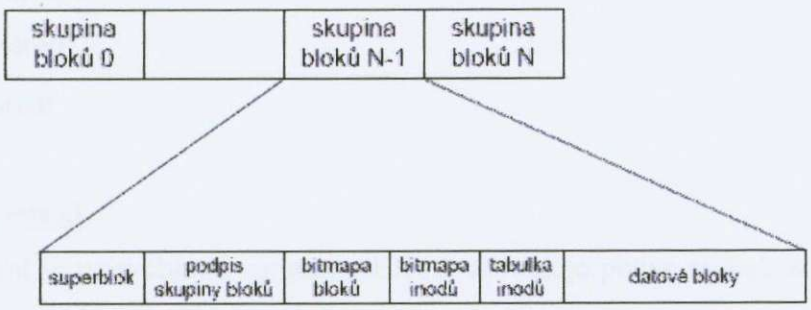
Fyzická struktura EXT2

Fyzická struktura byla silně ovlivněna vrstvou systému souborů BSD FFS (Fast File System). Souborový systém se skládá ze skupin bloků.



Obr.2.3.4.1 Struktura EXT2

Každá skupina bloků obsahuje navíc kopii klíčových záznamů struktury EXT2. Těmito klíčovými informacemi jsou superblok a deskriptory systému souborů, dále také každý blok obsahuje část souborového systému. Je to bitová mapa bloků a inod, tabulka inod a bloky dat.



Obr. 2.3.4.2 Struktura skupiny bloků

Použitím skupin bloků se dosahuje větší stability systému. Protože všechny důležité informace o systému souborů jsou několikrát zkopírované, je jednodušší zrestaurovat poškozenou část superbloku. Dále je tato struktura rychlejší, neboť došlo ke zredukování vzdáleností mezi tabulkou inod a blokem dat. To znamená, že se sníží vzdálenost, kterou musí čtecí hlavy překonat mezi informací, kde je daný soubor uložen a jeho fyzickým výskytem.

V souborovém systému EXT2 jsou složky realizovány jako seznamy odkazů záznamů proměnných délek. Každý takový záznam obsahuje číslo inody, délku záznamu, název souboru a délku názvu daného souboru.

2.3.5 EXT3 FS – Extended File System 3

Třetí verze systému Extended. EXT3 byl napsán Dr. Stephenem Tweediem jako nadstavba standardního linuxového souborového systému EXT2. EXT3 vznikl ze systému EXT2 pouhým přidáním speciálního souboru pro žurnálování. Jelikož EXT3 vychází z EXT2, má většinu jeho vlastností. Jako omezení velikosti souborového systému, velikosti souboru, velikosti bloků dat apod. Přestože některé informace se zapisují na EXT3 vícekrát než jednou, je EXT3 často rychlejší díky optimalizaci externí fragmentace.

EXT3 je obousměrně kompatibilní s EXT2. Pokud máme systém souborů EXT2, lze jej celkem jednoduše a téměř za běhu nahradit systémem EXT3.

Pomocí příkazu *tune2fs -j /dev/hda1* se vytvoří žurnál na svazku hda1. Stejně jednoduše lze žurnálovací soubor zrušit. Postačí k tomu příkaz *tune2fs -O ^has_journal /dev/hda1*. Systém EXT3 dává uživateli možnost vybrat si ze třech úrovní zabezpečení dat pomocí žurnálování. Tato volba se specifikuje v příkazu pro připojení souborového systému.

- Writeback
- Ordered
- Journal

Mód Writeback

Tento mód je nejrychlejší variantou EXT3, žurnáluje pouze metadata. Operace zápisu probíhají v optimálním pořadí s ohledem na výkon. Po nekorektním ukončení systému může dojít k chybám v datech, souborový systém však zůstane v pořádku. Při použití tohoto módu získáme pouze urychlení kontroly konzistence dat po havárii systému, je ekvivalentní systému EXT2.

Mód Ordered

Tento mód je implicitně nastaven. Opět se v něm žurnálují pouze operace s metadaty, ale změny v metadatech jsou uskutečněny až po zápisu dat do filesystemu. Díky tomu sice ztrácí systém lehce na výkonu, ale metadata v každém okamžiku odpovídají datům zapsaným ve filesystemu.

Mód Journal

Toto je nejpomalejší mód systému EXT3. Je to dáno tím, že se do žurnálu zapisují jak operace s metadaty, tak i s vlastními daty. Po nekorektním ukončení je možné úplně zrekonstruovat souborový systém. Tento mód představuje maximální ochranu dat před jejich poškozením v důsledku nekonzistence.

Podpora EXT3

Souborový systém EXT3 je podporován různými distribucemi, např.: RedHat (7.2), Mandrake (8.1), SuSE (7.3), Turbolinux.

Ačkoliv žurnálový souborový systém představuje pro počítač vyšší zátěž, autoři Ext3 se chlubí, že je jejich souborový systém dokonce rychlejší než Ext2. Má toho být docíleno tím, že zápisy jsou prováděny inteligentněji se snahou minimalizovat zbytečné přesuny hlavy disku.

2.3.6 EXT4 – Dual-Disk File System

Souborový systém EXT4 vychází ze systému EXT2 a je někdy také nazýván jako dual-disk file systém, protože používá dva diskové oddíly současně. Aby se plně využily výhody tohoto systému, je nutné mít každý oddíl na jiném fyzickém disku. Oba oddíly obsahují inody, přímé a nepřímé bloky, super blok a bitmapu. Tedy všechny informace o souborovém systému. Rozdíl mezi oddíly je ten, že jeden se používá výhradně pro ukládání adresářů a druhý pouze pro soubory. Symbolické odkazy, pojmenované roury a speciální soubory jsou uloženy na oddílu, kam se ukládají klasické soubory. Toto rozložení umožňuje paralelní práci se soubory a adresáři (proto požadavek, aby oddíly byly umístěny na různých discích). Požadavky na četní a zápis adresářového bloku a souborového bloku mohou být provedeny najednou - a tudíž se snižuje časová náročnost této operace.

Protože systém se nachází na dvou diskových oddílech, mohlo by dojít k problémům při alokaci jednotlivých inod, jelikož oddíly budou obsahovat inody se stejným číslem. Tento problém se řeší pomocí přečíslování, inody určené pro adresáře jsou sudé a inody souborů jsou liché. Toho je dosaženo tím, že do čísla inody se přidá číslo oddílu na místo posledního bitu.

Podobný problém by se mohl vyskytnout při alokaci diskových bloků. Zde se to ale musí řešit jinak, protože na diskový blok se můžeme odkazovat pouze z jednoho místa (dva různé soubory nemohou sdílet totéž místo). Při zápisu nebo čtení může dojít k několika situacím.

- Pokud blok již patří k nějakému souboru, podle čísla inody zjistíme, kterému diskovému oddílu náleží.
- Bloky patřící k superbloku nebo deskriptoru cylindrických skupin (cylinder group) jsou načteny při připojení nebo odpojení souborového systému.
- Volné bloky a bitmapy popisující volné inody jsou přístupné pouze při alokaci či dealokaci. Při dealokaci bloku víme, ze kterého oddílu pochází, díky struktuře dealokované inody. Při vytváření nového objektu určujeme, jaký typ se bude

vytvářet, zda adresář, soubor, či jiná struktura. Z toho vyplývá, že známe i oddíl, na kterém dojde k vytvoření daného objektu. A nakonec pokud je blok alokován pro rozšíření daného objektu, je jasné, ze kterého oddílu.

Výhody tohoto přístupu se projeví při používání vícevláknových či víceuživatelských aplikací, které budou chtít najednou přistupovat jak k souborům, tak k adresářům, protože klasické aplikace generují požadavky na přístup k disku sekvenčně, nikoli paralelně.

2.3.7 ReiserFS

Reiser File System je historicky první žurnálovací systém Linuxu. Jeho tvůrcem je Hans Reiser, po kterém byl tento file systém pojmenován. Ovšem vyvíjelo ho s ním samozřejmě ještě spousta dalších spolupracovníků.

Má mnoho vlastností, které ostatní file systémy neobsahují. Jedná se o žurnálovací souborový systém, a tak jednou z výhod, které poskytuje, je zajištění konzistence dat při výpadku a také zrychlení obnovení systému oproti fsck.

ReiserFS je založen na rychlém balancovaném stromu. Je tak zrychlena práce např. s velkým množstvím položek v adresáři. Není žádným problémem, aby adresář obsahoval třeba 100 000 souborů, ReiserFS s takto obsáhlým adresářem pracuje bez problémů.

Třetí a nejdůležitější vlastností je jeho schopnost pracovat efektivně s místem na diskovém oddílu. ReiserFS neukládá data do bloků pevných velikostí. Pokud chceme uložit několik souborů velkých například 100B, uloží jich Reiser file systém několik do jednoho bloku. Použitím tohoto souborového systému tedy šetříme nezanedbatelnou měrou místo na diskovém oddíle. Stejně tak jsou do jednoho bloku ukládány konce souborů, které už nezabírají celý blok. Na druhou stranu tím však dochází ke snížení výkonnosti souborového systému, kvůli vyšší míře externí fragmentace.

Dalším problémem je přidání dalších dat na konec souboru. V tomto případě je na první pohled zřejmé, že se jedná o složitější problém než např. u Ext2, kde přidání několika bytů je triviální záležitostí. Jednoduše se použije nevyužité místo v rámci bloku, který používá soubor jen z části. Ovšem u ReiserFS musí dojít k přesouvání dat do jiných bloků, aby se tam přidávaná data vešla.

Uchovávání malých souborů se hodí hlavně v případě databází. ReiserFS byl navržen, aby umožňoval ukládat malé záznamy na disku, tedy aby nebylo nutné používat nějaké způsoby shromažďování dat do větších souborů. Tento přístup by měl nejen zvýšit výkon, ale také snížit dobu vyvíjení aplikací tím, že programátoři nemusí vymýšlet způsoby, jak spojit několik malých bloků dat do jednoho velkého. Mohou je jednoduše na disk ukládat každý zvlášť.

ReiserFS má i některé své nedostatky (např.: práce s konci souborů), které výkon snižují. Tyto nedostatky se však dají částečně napravit pomocí správných nastavení při připojování oddílu. Pomocí volby noatime zrušíme záznamy file systému o posledním přístupu ke všem položkám file systému. Další volbou, která zrychlí práci se soubory na takovémto oddílu, je notail. Tím zrušíme ukládání konců souborů do jednoho bloku. Ztratíme tím sice nějaké místo na disku, ale dojde ke zrychlení práce s tímto systémem souborů.

Užitečnou vlastností ReiserFS je možnost změny jeho velikosti. Příkaz *resize_reiserfs* slouží právě ke změně velikosti diskového oddílu. Ke zvýšení velikosti ho dokonce nemusíme ani odpojovat. Pokud souborový systém zvětšujeme, je nutné, aby bylo samozřejmě na disku volné místo (např. zrušením následujícího oddílu pomocí příkazu *cfdisk*). Když chceme zmenšit file systém, musíme nejprve použít *resize_reiserfs-* a poté teprve *cfdisk*.

Zajímavá je podpora špatných bloků a pomocí externího patche i možnost kvót.

2.3.8 JFS

JFS je 64bitový file systém vyvinutý světoznámou firmou IBM s důrazem na vysokou propustnost a spolehlivost hlavně na serverech. Své místo má především na strojích s transakčně orientovanými operacemi.

Mezi jeho nejdůležitější vlastnosti patří žurnálování, to znamená rychlejší obnovu po havárii systému. Dále tento souborový systém používá extendy, díky čemuž dochází ke zrychlení práce se souborovým systémem, který podporuje efektivní a malé struktury pro mapování souborů. Podporuje také různé velikosti bloků. To dovoluje uživatelům zoptimalizovat tento systém podle svých potřeb. Podporuje bloky o velikosti o 512, 1024, 2048, 4096 bytů. Používá dynamické alokování inod, čímž se zamezí rezervování fixního místa na disku pro jednotlivé inody v průběhu vytvoření souborového systému. Dále používá dva způsoby

organizace adresářů. První z nich se užívá pouze pro malé adresáře, které obsahují maximálně 8 položek. Obsah takového adresáře se uloží přímo do příslušné inody daného adresáře. Druhým způsobem je balancovaný strom setříděný podle jména, který poskytuje daleko rychlejší možnost přístupu v porovnání s nesetříděnou organizací. Mimo jiné podporuje také řídké soubory.

JFS je tedy moderní žurnálovací souborový file systém s garantovanou konzistencí dat (díky žurnálu). Je šířen pod licencí GPL. Snaží se být podporovatelný bez zásahu do zdrojových kódů kornelu, a to pod všemi platformami, které podporuje Linux. Jeho podpora je zajištěna pro jádra řady 2.4 a 2.5 pomocí externího patche.

2.3.9 XFS

XFS byl vyvinut firmou SGI jako náhrada file systému EFS a v roce 2001 vyšla podpora tohoto souborového systému i pro operační systém Linux. Tento souborový systém se používá pro high-end servery (obsahuje podporu pro multiprocesorové počítače), zajišťuje rychlé zotavení při pádech a podporuje extrémně velké diskové farmy. Jedná se o žurnálovací file systém s 64-bitovým adresováním. Jeho hlavními výhodami jsou robustnost, důvěryhodnost a možnosti, které ostatní file systémy neposkytují. Má však i svoje nevýhody, jako je slabší výkon při mazání velkého počtu malých souborů nebo příliš veliký kód.

Hlavními výhodami systému XFS je žurnálování, rychlé transakce, vysoké limity a výkon souborového systému. Při žurnálování dochází pouze k zápisům informací o metadatech. XFS se snaží co nejvíce snížit vliv žurnálování při zápisu a čtení dat. Vysoké limity vycházejí z faktu, že XFS je 64bitový systém - a tudíž je schopen obsahovat soubory o velikosti několik milionů terabytů. XFS efektivně používá vstupně-výstupní operace, čímž dosahuje vysoké výkonnosti.

Pro práci se systémem XFS jsou nutné určité uživatelské nástroje. Těmito nástroji jsou *acl, attr, quota, xfsdump* a *xfsprogs*. Příkaz *acl* slouží pro správu ACL (Access contol list), *attr* je určen k manipulaci s rozšířenými atributy (EA - extended attributes), *quota* slouží k monitorování a omezování diskového prostoru používaného uživateli, případně skupinou uživatelů. Balíček *XFSdump* obsahuje příkazy *xfsdump, xfsstore* a další příkazy pro administraci systému XFS. *Xfsdump* prohledává soubory file systému a rozhoduje, které je nutné zálohovat a kopíruje je na určený disk, pásku, případně jiné zálohovací médium. *Xfsdump* ukládá i rozšířené atributy, a to ve

formátu, který je vhodný pro přenos mezi různými souborovými systémy. Příkaz *xfsstore* provádí obnovu dat ze zálohy. Balíček *Xfsprogs* obsahuje příkazy pro práci se systémem XFS, jakými jsou *mkfs, make configure, make, su root, make install.*

XFS je moderní souborový file systém s garantovanou konzistencí dat díky žurnálu. Jeho podpora existuje pro jádra řady 2.4 a 2.5 pomocí externího patche. Obsahuje podporu kvót, EA (extended attributes) a ACL (access control lists). Stejně jako u všech ostatních moderních souborových systémů je zde možnost ukládat žurnál na oddělený diskový oddíl. Je možné za běhu zvyšovat velikost. Podporuje také zálohování pomocí příkazů *xfsdump* a *xfsrestore*. Dále je podporováno swapování do souborů. Pro běh tohoto file systému je doporučováno minimálně 64MB paměti.

2.3.10 UFS – Unix File System

UFS je klasický unixový systém skládající se z několika částí. Na začátku diskového oddílu je boot blok, který obsahuje informace důležité pro start operačního systému. Za boot blokem následuje superblok obsahující informace o souborovém systému - jako je počet data bloků, maximální počet souborů a ukazatel na free list, spojový seznam všech volných bloků souborového systému. Za superblokem je umístěn blok inod a za ním je data blok.

Adresáře a soubory jsou na disku uloženy do stejných objektů, jen s tím rozdílem, že adresář obsahuje seznam odkazů (linků) na další takové objekty, ať už soubory či adresáře. Každému objektu je přiřazena inoda. Inoda je descriptor (popisovač), který uchovává informace o souboru či adresáři. Obsahuje informace o vlastnictví souboru, časové značky pro poslední přístup k souboru a jeho poslední modifikaci a pole ukazatelů na data bloky souboru. Inoda také uchovává ukazatele na prvních 8 data bloků souboru. Pokud velikost souboru přesahuje 8 bloků, obsahuje inoda odkazy a jednoduché nepřímé bloky (simply indirect blocks), které obsahují zbytek souboru. Při velikosti bloku 512 bytů obsahuje inoda odkaz na 128 nepřímých bloků. Jestliže nestačí ani jednoduché nepřímé bloky, lze stejným způsobem alokovat dvojitě či trojitě nepřímé bloky (doubly, triply indirect blocks).

Tento systém je jednoduchý, ale pomalý, protože může číst či zapisovat na fyzické medium datové bloky maximálně o velikosti 512 bytů. A vyhledávání bloků dat na disku, protože data nejsou na disku nijak seřazena, způsobuje další zpomalení.

2.3.11 FFS – Berkley Fast File System

Fast File System pochází z univerzity v Berkley. Na začátku disku obsahuje boot blok, po kterém následuje super blok. Protože super blok obsahuje důležitá data a při jeho poškození by mohlo dojít ke ztrátě dat, je na disku umístěna ještě jeho kopie. Ta je vytvořena hned při vytváření souborového systému a přistupuje se k ní pouze v případě změny hlavního super bloku. Aby bylo možné ukládat soubory o velikosti 4MB maximálně do dvojitě nepřímých bloků, došlo k navýšení základního bloku na 4KB. Tato velikost může být změněna, ale pouze směrem vzhůru. Velikost těchto bloků je jako základní konstanta souborového systému uložena v super bloku, určuje se při vytváření systému a jen tehdy jde změnit. FFS se vyznačuje také novou architekturou, rozděluje diskový oddíl na části zvané cylinder groups (CG). Cylindrická skupina odpovídá jednomu nebo více skutečných cylindrů na pevném disku. Pokud se CG skládá z více cylindrů pevného disku, je tento blok vždy souvislý. Každá CG obsahuje kopii super bloku, místo pro inody, bitovou mapu volného místa a informace popisující využití bloků v této skupině. Tyto skupiny mají pevně danou délku, ta je určena při vytváření systému počtem inod v ní obsažených. Pokud by všechny kopie super bloku byly umístěné na začátku každé skupiny, došlo by ke ztrátě všech kopií v případě poškození horní plotny disku. Aby se tak nestalo, jsou kopie super bloku umístěny vždy o stopu dále než v předchozí skupině. Ve skutečnosti to potom vypadá, jako by kopie super bloku sestupovaly po spirále od shora dolů po pevném disku. Místo, které se nachází mezi začátkem skupiny a kopií super bloku, se používá pro uložení data bloků. Data jsou uložena po 4KB blocích, což odpovídá objemu dat, se kterým lze provést na jednu diskovou transakci. Aby nedocházelo ke zbytečnému plýtvání místem při ukládání menších souborů, jsou tyto bloky rozděleny na menší části, nazývané fragmenty, přičemž každý fragment je adresovatelný samostatně. Velikost fragmentu lze nastavit při vytváření souborového systému. Nejmenší možnou velikostí fragmentu je 512B, to odpovídá velikosti jednoho diskového clusteru (nejmenší adresovatelná jednotka). Znamená to, že data blok je rozdělen na 2,4, nebo 8 fragmentů. Pokud je zapisovaný soubor menší než data blok, alokuje se pouze odpovídající počet fragmentů a zbylé místo v data bloku zůstane volné. Buď pro případ, že by byl daný soubor dodatečně zvětšen, nebo se na něj uloží jiný soubor, případně jeho část. Naopak, pokud velikost souboru přesahuje velikost data bloku, alokuje se více data bloků, pokud možno jako souvislý blok v cylindrické

skupině. A jestliže zbude část menší než celý blok, systém se pokusí nalézt volné fragmenty pro tuto část. Pokud se informace zapisují na konec souboru, který je uložen do fragmentu a v daném bloku není místo, systém přesune celý blok. Buď najde potřebný počet fragmentů v celku, nebo alokuje celý nový data blok.

FFS je navržen tak, aby se dokázal přizpůsobit fyzickému disku, na kterém je oddíl s tímto systémem vytvořen. Jedná se o tzv. parametrizaci systému. Jde o to, aby byl systém optimalizován pro daný hardware. Počítá se s rychlostí procesoru, počtem otáček disku, přístupovou dobou, počtem sektorů na stopu. Z těchto informací se dá spočítat optimální rozložení dat na disku, aby se při provádění výpočtů procesorem nemuselo čekat na data. Uloží se tak na místo, kam se diskové hlavy dostanou, než proběhne daný výpočet. Tento proces se nazývá otáčková optimalizace (rotationally optimal). Super blok obsahuje vektor seznamů, který se označuje jako tabulky otáčkového rozložení (rotational layout tables). Při vytváření souborového systému jej lze optimalizovat i pro jiný procesor, než na jakém se systém vytváří.

Systém FFS podporuje názvy adresářů a souborů o maximální délce 255 znaků. U FFS byl zaveden mechanismus zamykání souborů. Jedná se o postup, kdy několik běžících procesů potřebuje přistupovat k jednomu souboru. Pokud proces potřebuje použít soubor, nejprve se jej pokusí zamknout a teprve poté s ním může pracovat. Tento princip má u FFS několik nedostatků. Prvním z nich je procesorová náročnost při opakovaných pokusech o zamčení souboru, protože pokud je daný soubor jedním procesem zamčen, druhý proces čeká na odemčení, aby jej zamkl pro vlastní potřebu. Probíhá tak, že se jej pokouší v určitých intervalech zamknout. Dalším nedostatkem je nutnost odemknutí souborů po pádu systému, to musí být uděláno ručně, pomocí skriptu spouštěného po nastartování systému. A posledním nedostatkem je rozdílný přístup procesů při vytváření souborů - ten je dán různými právy procesů.

FFS také podporuje používání kvót. To je důležité obzvlášť pro systémy s velkým počtem uživatelů, kde mezi nimi dochází ke sdílení systémových prostředků. Především se jedná o volné místo na pevném disku, a kde by nebylo vhodné, aby si všechny prostředky alokoval jeden uživatel. Mechanismus kvót omezuje jak maximální počet inod pro uživatele, tak maximální počet diskových bloků. Užívají se dva typy limitů, hard a soft. Při dosažení soft limitu je uživatel upozorněn - a pokud dosáhne hard limitu, nebude mu umožněno alokovat další zdroje a procesy, které je potřebují, budou ukončeny.

2.3.12 ZFS – Zettabyte File System

Zettabyte file sytem je první 128bitový souborový systém. Pochází z dílny firmy Sun Microsystems a je součástí nového operačního systému Solária, avšak je podporován i v systému Solaris 10. Tento souborový systém je postaven na principu virtualizace. Je navržen tak, aby mohl zaujímat několik fyzických disků. Ty jsou začleněny do storage pool. Storage pool by se dal přirovnat k RAIDovému uspořádání disků. Díky 128 bitovému adresování jeho velikost přesahuje dnešní možnosti (Pokud by uživatel vytvářel 1000 souborů za sekundu, trvalo by dosažení maximálního počtu souborů něco okolo 9000 let.). Je určen jak pro servery typu SPARC, tak i pro počítače typu x86. Je tedy možné vzít storage pool ze serveru a připojit ho ke klasickému počítači, případně i opačně. ZFS používá transakčně objektový model COW (copy-on-write). Copy-on-write model pracuje na principu sdílení dat mezi jednotlivými procesy. Pokud je několik procesů, které požadují stejná data, je všem těmto procesům přiřazen odkaz na daná data. Pokud proces změní data a požaduje jejich zápis, je vytvořena nová kopie těchto dat a ta je poté zapsána na jiné místo. To znamená, že ZFS nikdy neprovádí přepis dat na místě. Sytém ZFS také používá princip dynamic striping pro zápis dat do storage pool. Princip dynamic striping se vyznačuje tím, že rovnoměrně využívá všechna media ve storage pool pro zápis. ZFS pro zlepšení vlastností systému lze také zapnout využívání variabilní velikosti bloků, do kterých se zapisují data. To znamená, že systém neplýtvá místem, protože pokud je velikost dat menší než standardní velikost bloku, alokuje se menší blok pro zápis dat.

ZFS je navržen tak, aby co nejvíce zjednodušil svojí administraci. Uvádí se, že je správa systému zjednodušena až o 80%. Zajišťuje integritu dat pomocí 64bitových kontrolních součtů, což umožňuje značnou opravu poškozených dat.

Maximální počet souborů je 2^{48}, maximální velikost souborového systému a maximální velikost souboru je 16 EB (16 Exabyte = 2^{60} bytů, přibližně 10^{18}).

2.3.13 UMSDOS – Unix MSDOS

UMSDOS je linuxový virtuální souborový systém. Uveden byl v lednu 1994 a standardně je podporován od jádra 1.1.36. Byl vytvořen proto, aby se z FAT mohl stát plnohodnotný linuxový file systém. UMSDOS rozšířil systém FAT o podporu dlouhých jmen souborů a adresářů, používání speciálních znaků ve jménech souborů (lze použít tečku, mezeru, různé znaky národních abeced). Podporuje také vlastníka,

uživatele, skupiny a jejich práva k souborům. Tento systém je case sensitive, což znamená, že rozlišuje malá a velká písmena v názvech souborů. Umožňuje uživateli vytvářet odkazy (links) - a to jak symbolické, tak i hardlinky.

Systém UMSDOS funguje na klasickém FAT systému a pro programy MSDOSu je od něj nerozpoznatelný, zatímco pro linuxové se tváří jako plnohodnotný linuxový systém. Protože Fat nepodporuje speciální funkce linuxového filesystému, řeší se to pomocí zvláštního EMD (Extension to Msdos Directory) souboru --linux-.---, do kterého se tyto informace zapisují. Tento soubor se nachází v každém adresáři tohoto souborového systému. Jeho obsah je rozdělen na položky o délce 64 bytů. Každá položka popisuje jeden soubor. Pokud k popisu daného souboru jedna položka nestačí, alokuje se další položka. Pokud je svazek se systémem UMSDOS připojen jako, což je možné, je EMD soubor viditelný. Pokud v tomto stavu dojde k nějaké změně, lze EMD soubor obnovit pomoci příkazu *umssync*.

Výhodou tohoto sytému je, že se nemusí pro linux zřizovat vlastní diskový oddíl s linuxovým souborovým systémem. Jeho nevýhodou je však nižší rychlost.

2.4. MAC OS

2.4.1 MFS – Macintosh File System

MFS je první souborový systém firmy Apple. Byl představen 24.1.1984. MFS je flat file systém, protože nepodporuje stromovou adresářovou strukturu. Obsah adresáře je uložen v tabulce a soubory jsou ukládány jako spojové seznamy.

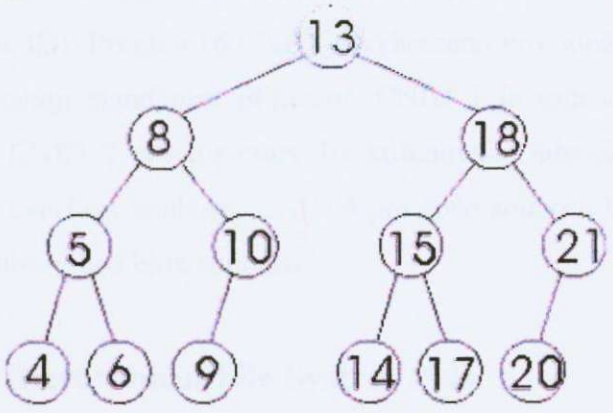
Maximální velikost svazku je 256MB. Soubor může mít maximální velikost shodnou s největší možnou velikostí svazku, 256MB. Počet souborů je omezen na 2^{12} (4096). Délka jména souboru je omezena na 255 znaků.

Podpora tohoto systému byla omezena v září 1985, kdy byla vypuštěna verze systému Mac OS 7.6.1, pouze na čtení z toho oddílu. A podpora byla úplně zrušena ve verzi Mac OS 8.0.

2.4.2 HFS – Hierarchical File System

Jelikož firma Apple požadovala při ukládání souborů ukládat nejen vlastní data, ale i další informace o adresářích a souborech, bylo nutné vyvinout zcela nový systém, který by nahradil dosud používaný MFS. Poprvé byl veřejnosti představen 17. září 1985 v operačním systému MacOS 4.0.

Hierarchický souborový systém, jak již sám název napovídá, umožňuje ukládat soubory a adresáře do stromové struktury. Adresářovou strukturu, obsah souborů i tzv. bad bloky (seznam vadných sektorů disku) uchovává ve formě B*-stromů (binární vyhledávací strom).



Obr.2.4.2.1 Binární vyhledávací strom

HFS rozděluje disk na logické bloky o velikosti 512 bytů. Tyto logické jednotky jsou pak spojovány do alokačních bloků. Velikost alokačních bloků je určena

celkovou velikostí svazku naformátovaného tímto systémem. HFS používá k 16bitové adresaci alokačních bloků, což stanovuje maximální počet alokačních jednotek na 65536. Disk je rozdělen na pět částí, boot blok, master directory blok (MDB, je známý také jako Volume Information Blok - VIB), volume bitmap (bitová mapa volného prostoru), extants overflow file a catalog file (katalog souborů). Pro správu a organizaci souborů a adresářů na disku a pro organizaci informací v paměti je používán správce souborů (File Manager). Boot blok je umístěn na nultém a prvním logickém bloku. Je určen pro uložení informací důležitých pro start operačního systému. Druhý logický blok je určen pro uložení MDB. MDB obsahuje informace o samotném souborovém systému, datum jeho vytvoření a počet souborů, velikost logických bloků a odkaz na bitovou mapu. HFS obsahuje ještě záložní kopii MDB, tzv. alternate master directory block. Za blokem MDB, na třetím logickém bloku, začíná bitová mapa volného prostoru. V bitové mapě je každá alokační jednotka reprezentována jedním bitem, proto je její velikost určena velikostí celého svazku. Pokud je tento bit nastaven na 1, potom odpovídající alokační jednotka je již obsazena. 0 značí volnou alokační jednotku. Za bitovou mapou následuje soubor extents overflow. File manager udržuje v paměti seznam alokačních bloků daného souboru díky tomu, že si zapisuje souvislé bloky disku do paměti. Pokud je paměť vyhrazená file manageru zaplněna, zapíše obsah paměti do extents overflow souboru. Poslední částí HFS je katalog souborů. Tento katalog obsahuje informace o uspořádání adresářů a souborů v adresářové struktuře. Katalog je uložen na disku ve formě B*-stromu. Každý uzel v této struktuře má své jednoznačné identifikační číslo – CNID (Catalog Node ID). Prvních 16 CNID je vyhrazeno pro použití firmou Apple Computers Inc. a obsahují standardní přiřazení. CNID 1 je určeno pro parent ID kořenového adresáře, CNID 2 pro directory ID kořenového adresáře, CNID 3 pro číslo souboru extents overflow souboru, CNID 4 pro číslo souboru katalogu souborů a CNID 5 pro číslo souboru bad blok souboru.

2.4.3 HFS+ – Hierarchical File System Plus

HFS+ je souborový systém firmy Apple, který nahradil dříve užívaný systém HFS. Poprvé byl představen 19.1.1998. Je používán od systému verze Mac OS 8.1. HFS+ je nejen využíván pro pevné disky v počítačích, ale i v zařízeních iPod. Je to žurnálovací systém založený na ukládání dat do struktury B*-stromu. Diskový prostor

zformátovaný tímto systémem je rozdělen na alokační bloky o velikosti 32 bitů. Pro jména souborů a adresářů je použito kódování Unicode, délka jmen je maximálně 255 znaků.

2.4.4 HPFS – High Performance File System

HPFS byl vyvinut pro systém OS/2 v. 1.2 firmami IBM a Microsoft v roce 1989. Tento systém byl podporován Microsoftem v operačním systému Windows NT 3.x. V novějších verzích systému Windows by měl být HPFS podporován za pomoci patche.

Systém HPFS měl nahradit do té doby používaný systém FAT, který byl pro pevné disky se stále vzrůstající kapacitou nedostačující. Z toho souborového systému vychází systém NTFS, ale je s ním nekompatibilní.

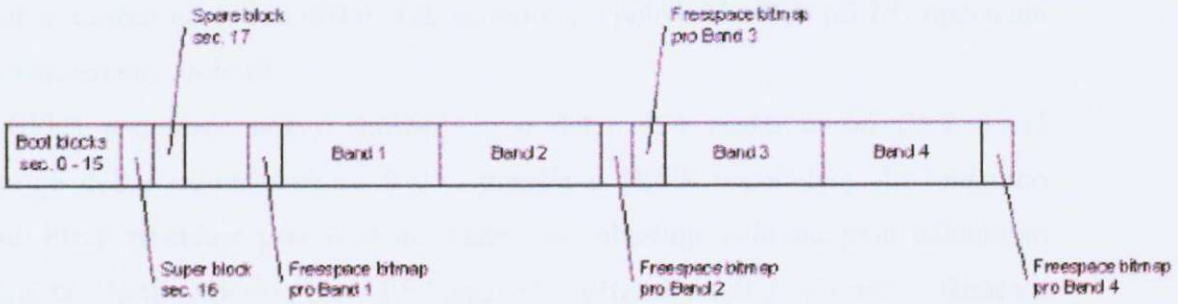
Rozdělení HPFS oddílu:

Každý disk s HPFS je rozdělen na několik částí. Na začátku disku je BootBlock, který je uložen od 0. do 15. sektoru, přičemž nemusí být a ani nejsou využity všechny sektory. V sektoru 0 je uloženo jméno disku a také 32bitový identifikátor disku. Následující sektory BootBlocku jsou rezervovány pro zavaděč operačního systému. Sektor 16 obsahuje tzv. SuperBlock. V SuperBlocku jsou uloženy ukazatele na seznam bitových map volného místa na disku, na seznam BadBlocků (vadných sektorů na disku), na kořenový adresář, na directory block band. Obsahuje také pozici a délku Victory block bandu, stejně jako ukazatel na jeho speciální bitmapu. A dále ještě obsahuje datum posledního CHKDSK /F. Následuje SpareBlock v sektoru 17, který obsahuje další ukazatele a flagy.

Zbytek diskového oddílu je rozdělen na části zvané bands. Bands jsou oblasti disku o velikosti 8MB. Každý band má svou vlastní bitovou mapu volného prostoru (freespace bitmap) o velikosti 2KB, kde 1 bit odpovídá 1 sektoru (0 = používaný, 1 = volný). Tato bitmapa je obvykle uložena pro lichý band na začátku a pro sudý na konci. Z takovéhoto uspořádání vyplývá možnost souvislé alokace prostoru o velikosti téměř 16MB, což jsou 2 bandy o velikosti 8MB zmenšené o velikosti svých bitových map volného prostoru. Pokud je to možné a pokud to systém vyžaduje, alokuje se souvislý prostor. Pokud to není možné, použije se odkaz na další objekt nazývaný Alnode, či Anode (Allocation node), jestliže nebude ani tento objekt postačovat a

alokuje se další. Tímto způsobem zápisu nedochází k tak vysoké fragmentaci. A protože bitmapy jsou v blízkosti uloženého souboru, je přístup k souborům rychlý.

Dále HPFS ještě obsahuje directory block band. Ten je umístěn uprostřed disku a obsahuje kořenový adresář.



Obr.2.4.4.1 Struktura HPFS

Ukládání souborů a adresářů:

Každému souboru a adresáři, včetně kořenového, je přiřazen systémový objekt fnode. Fnode je dlouhý jeden sektor a obsahuje ukazatel LSN (Logical Sector Number) na nadřazený adresář, celkovou délku jména souboru a prvních 15 znaků jména, informace o rozšířených atributech (extended attributes - EA), dále obsahuje přístupová práva k danému objektu a jeho alokační strukturu. Pokud je velikost rozšířených atributů menší než 145B, jsou v objektu fnode celé. Tento fnode lze alokovat kdekoli na disku, ale OS/2 se jej snaží uložit v blízkosti dat daného objektu.

Alokační struktura v objektu fnode může mít několik podob. Podoba se řídí podle velikosti daného objektu a fragmentací jeho dat. Pokud je soubor rozdělen nejvýše na 8 fragmentů, jsou všechny fragmenty, tzv. extents, vypsány v objektu fnode. Pokud je soubor více fragmentovaný, stane se alokační struktura kořenem B+ stromu (Jedná se o typ B-stromu, který má hodnoty uloženy pouze v listech.).

Tak má prostor pro 12 pointrů na podstromy. Vnitřní uzly B+ stromu mohou mít maximálně 60 listů a každý list může obsahovat až 40 ukazatelů na fragmenty souboru. Každý fragment je uložen jako trojice: sektor v souboru, délka a sektor na disku. Každá položka o délce 32 bitů. Toto uložení se nazývá runlength encoding.

Ukazatel na kořenový adresář je, jak již bylo uvedeno, zapsán v SuperBlocku.

V objektu Fnode pro adresář je vždy obsažen ukazatel na kořenový directory block adresáře. Každý direktory block je spojitý a má vždy velikost 2kB. HPFS má snahu alokovat tyto blocky v direktory block bandu. Pokud je tento prostor zaplněn, HPFS alokuje další kdekoli na disku.

Výhody a nevýhody HPFS:

Jeden z hlavních nedostatků systému FAT, maximální velikost oddílu, byl odstraněn použitím stromové struktury pro ukládání dat. Dalším vylepšením HPSF oproti FAT je, že každý band obsahuje jakousi obdobu FAT tabulky a že kořenový adresář je uložen uprostřed disku. Tak se redukuje pohyb hlaviček při I/O operacích s tímto systémem souborů.

HPFS podporuje názvy souborů až o délce 254 znaků oproti FAT, která podporuje délku názvu souborů 8+3. Adresáře u HPFS uspořádány do binárního stromu, který zrychluje prohledávání. Dále ještě obsahuje ochranu proti náhodným chybám, tzv. hotfix sektory, Dirty FS Flag, což je příznak značící nekorektní ukončení systému a nutnost kontroly disku.

Systém HPFS ještě obsahuje rozšířené atributy (Extended Attributes – EA), datovou oblast pro uživatelsky definované atributy o maximální velikosti 64kB na jeden soubor nebo adresář. EA jsou uživatelsky definované atributy. Jsou určeny jako dvojice – klíč, hodnota, kde klíčem je ASCII řetězec a hodnotou může být také ASCII řetězec, nebo libovolná sekvence bytů. Tyto atributy jsou uloženy buď přímo v fnode, jsou-li dostatečně malé, případně jsou uloženy do B+stromů.

Dále také obsahuje podporu pro multitasking (běh několika úloh současně).

Protože položka adresáře obsahuje všechny potřebné informace o souboru či adresáři, jako jsou jeho délka, příznaky, délka EA, čas vytvoření, zápisu a posledního přístupu, délka jména a jméno souboru / adresáře a případný ukazatel na B-strom, ve kterém je uložen. Délka této položky je závislá na jejím obsahu, a proto je její délka zarovnána na 4B.

HPFS podporuje zpožděný zápis na disk. Pokud systém vyšle požadavek o zápis dat na disk, data se uloží do cache paměti a je nastaven příznak (flag) dirty. Pokud je cache paměť zaplněna položkami s příznakem dirty, nebo pokud se právě nepracuje s pevným diskem, spustí proces (démon), který zapíše všechna data na disk.

Kvůli použití B-stromů ve struktuře uložení dat může dojít ke složitějším operacím nad tímto stromem při operacích se soubory. Pro tyto operace si HPFS udržuje několik volných bloků – Emergency Free Block List. Ukazatel na toto volné místo se nachází ve Spareblocku.

Maximální velikost oddílu je omezena 512GB.

3. Síťové souborové systémy

3.1 NFS – Network File System

Tento souborový systém byl vyvinut firmou Sun Microsystem roku 1984 jako distribuovaný FS. Je určen pro sdílení souborů v síti. Přístup k souborům pomocí tohoto systému je stejně snadný, jako by dané soubory byly uloženy na lokálním disku. NFS založen na balíku RPC (Remote Procedure Call – Vzdálené volání procedur). Balík RPC poskytuje systému obecný mechanismus pro aplikace typu klient-server. Jedná se o sbírku funkcí a knihoven.

Ve druhé verzi tohoto systému probíhá komunikace pomocí protokolu UDP. Ve třetí verzi je zavedena podpora protokolu TCP. Je také přidána podpora pro přenos souborů větších než 4GB za pomoci 64bitové atributu udávající velikost souboru. Třetí verze také podporuje asynchronní zápis na disk serveru.

3.2 AFS – Andrew file system

Tento sytém vyvinutý na Carnegie Mellon University původně nesl označení Andrew File System. Nyní jeho vývoj převzala firma IBM, označila jej jako IBM AFS, uvolnila jeho kód a tím umožnila vznik projektu OpenAFS.

AFS je distribuovaný souborový systém, rozdělený na buňky (cells), kde každá buňka může představovat několik administrativně sloučených serverů. Tento systém má dva typy serverů, AFS server a Volume Location Server. Sytém AFS má ve všech uzlech stejný vzhled adresářového stromu. Kořen tohoto stromu je umístěn v adresáři /afs. Do adresáře /afs se poté umisťují adresáře reprezentující jednotlivé buňky. Obsahy jednotlivých buněk jsou uloženy do svazků (volumes), přičemž tyto svazky mohou být umístěny na libovolném AFS serveru. Lze je také v případě nutnosti mezi jednotlivými AFS servery přesouvat, aniž by bylo nutné o tom informovat uživatele, protože struktura adresářových stromů se nemění. Pokud si klient vyžádá přístup k souboru, jeho požadavek se nejprve odešle na Volume Location Server (VLS), který zjistí, kde je daný soubor fyzicky uložen a teprve poté je daný soubor přístupný. Aby nedocházelo ke zbytečnému zdržování při častém používání některých souborů, podporuje AFS lokální cache - buď paměťovou, nebo diskovou. To znamená, že často používané soubory jsou trvale uloženy na lokální stanici. Dalším způsobem, jakým AFS zvyšuje dostupnost dat, jsou replikační servery (replica servers). Ty jsou uzpůsobeny pro uchovávání souborů, které jsou určeny pouze pro čtení (read-only).

Soubory určené pouze pro čtení mohou být uloženy na několika replikačních serverech. K replikačním serverům se přistupuje podle určitých pravidel, klient může k serveru přistoupit, pouze pokud je daný server volný. Pokud je takových serverů se stejným obsahem více, přistupuje se k tomu nejbližšímu.

Přístupová práva AFS uděluje jednotlivým adresářům, nikoli souborům. Soubory přebírají práva od adresářů, ve kterých jsou umístěné. Tři nejdůležitější systémové skupiny jsou – system:anyuser, system:authuser, system:administrators. System:anyuser je jakýkoli uživatel, i ten, který nemá na AFS vytvořený svůj vlastní účet, system:authuser je skupina autorizovaných uživatelů pro daný AFS a system:administrators jsou administrátoři buňky. Každý ACL (Acces Control List – specifikace přístupových práv) adresáře obsahuje pro každou skupinu definovaná práva. Jsou to práva rlidwka (read, lookup, insert, delete, write, lock a administrate).

Právo read – r – definuje právo čtení z adresáře, vztahuje se i na čtení informací z inodu daného adresáře.

Právo lookup – l – definuje právo pro výpis adresáře a přístup do podadresářů.

Právo insert – i – definuje právo pro vytváření nových souborů v adresáři.

Právo delete – d – definuje právo pro mazání již existujících souborů v daném adresáři.

Právo write – w – definuje právo pro zápis do existujících souborů a modifikaci podadresářů.

Právo lock – k – definuje právo pro uzamykání souborů v adresáři.

Právo administrate – a – definuje právo pro správu přístupových práv pro daný adresář.

Pro zjednodušení jsou některá práva v příkazech slučována, např.: read = práva rl (read + lookup), write = práva rlidw (všechna mimo lock a administrate), all = všechna práva, none = odejmutí všech práv.

3.3 CODA FS

Coda FS je pokročilý síťový souborový systém vytvořený na CMU (Carnegie Mellon University) v roce 1987. Je odvozený od souborového systému AFS. Jeho výhodami jsou zaměření na podporu mobilních operací a disconnected operací, volná šiřitelnost, cachovaní na straně klienta, replikace serveru, zabezpečení pro autentifikaci, podpora šifrování a kontrola přístupu. Dále ještě dokončení operací i při

výpadku sítě, adaptace na šířku pásma sítě, možnou rozšiřitelnost a dobře definované sdílení.

Disconnected operace jsou takové operace, které normálně probíhají, když je systém připojen do sítě a dojde k jejímu náhlému výpadku. Dokončení těchto operací je umožněno pomocí cachování na straně klienta. Cachování je jednou z velkých výhod systému CODA. Jedná se o to, že pokud přijde požadavek na přístup k datům od klienta, jsou tato data ze serveru zkopírována a lokálně se uchovávají, dokud se lokální kopie shoduje s kopií uloženou na serveru. To snižuje množství dat, která je nutno přenášet po síti. Toto cachovnání se také někdy nazývá hoarding. Provádí se pomocí příkazu *hoard*, který nakopíruje ze sítě soubory a adresáře vypsané v textovém souboru hoardfile. Další výhodou systému CODA je, že sdílené soubory a adresáře jsou přístupné ze všech počítačů v síti pomocí stejné cesty. To je umožněno tím, že se celý systém CODA na všech pracovních stanicích připojuje do stejného adresáře */coda* (podobně se připojoval systém afs jako /afs).

Před odpojením od sítě by se měl spustit příkaz *cfs disconnect*, který synchronizuje soubory v cache se soubory v síti. Pro znovupřipojení do sítě se používá příkaz *cfsreconnect*. Příkaz *cfsreconnect* taktéž provádí synchronizaci souborů. Pokud byly soubory v síti změněny několika uživateli, používá se program repair, který řeší konflikty v synchronizaci. Pokud soubory změnil pouze jeden uživatel, provede se „neviditelná" synchronizace a systém nepodá žádné hlášení.

3.4 SMB – Server Message Block

SMB původně pochází z dílny IBM, ale do dnešní podoby jej upravila firma Microsoft. Je to síťový protokol používaný ke sdílení souborů, tiskáren, sériových portů a jiných komunikačních zařízení. Byl optimalizován pro použití v lokálních sítích, lze jej ale použít k přístupu mezi jednotlivými podsítěmi v rámci internetu. Pro síťovou komunikaci využívá protokolů TCP/IP. Bezpečnost SMB je zajištěna na dvou úrovních. Na první úrovni – share level – se jedná o zaheslovaný přístup k souborům a na druhé – user level – o přidělení přístupových práv. Pro přístup je nutné se zalogovat na server. Pro potvrzení platného přístupu ke sdíleným prostředkům používá SMB doménovou autentifikaci, tato autentifikace probíhá za použití MSRPC (Microsoft Remote Procedure Call – Vzdálené volání procedur). V roce 1998 došlo k přejmenování SMB protokolu na CIFS (Common Internet File System). Bylo také

přidáno několik vylepšení, především podpora symbolických hardlinků a možnost práce s většími soubory.

SMB pracuje na principu spojení klient-server, kde klient vysílá žádosti na server a ten mu odpovídá. SMB se skládá z několika specializovaných částí. Jedna část je určená pro přístup k souborovému systému, to znamená komunikaci s file serverem (Server, kde jsou uloženy soubory). Ostatní části se používají pro IPC (Interprocess communication – komunikace mezi procesy).

SMB servery sdílejí v síti své souborové systémy a ostatní zdroje. Klientské stanice mohou mít svůj vlastní disk, který nemusí veřejně sdílet na síti, přesto mohou přistupovat ke sdíleným prostředkům serverů v síti.

Pro komunikaci mezi klientem a serverem v síti se používá broadcastingu (broadcast – všesměrové vysílání, vysílá se ke všem stanicím v síti). Broadcasting, který SMB používá, pochází od služby NetBIOS, ve kterém je SMB postaven. Broadcasting může způsobit problémy, ale ty mohou být zmenšeny za použití WINS (Windows Internet Naming Service)

Existuje několik free (volně šiřitelných) implementací systému SMB. Jedním z nich je Samba, která obsahuje jak server, tak klienta. Pro linux existují dvě verze, smbfs a cifsfs, pro NetBSD, FreeBSD a MacOS X je to smbfs. Smbfs a cifsfs jsou přímo implementovány ve virtuálních souborových systémech operačních systémů, ve kterých se používají.

3.4 Google FS

Tento distribuovaný souborový systém od společnosti Google byl navržen pro ukládání velkého množství velmi velkých a zřídka kdy přepisovaných souborů na větší počet nespolehlivých medií. Velikosti souborů se pohybují od stovek megabytů až po několika gigabytové či terabytové soubory. GFS umožňuje ukládat i malé soubory o velikosti několika kilobytů, ale není pro ně optimalizován.

Cluster tohoto sytému se skládá z jediného Master serveru a několika Chunkserverů. Protože cluster obsahuje jediný master server, stává se z něj potencionálně slabé místo, proto master server ukládá pouze informace o umístění souborů a jejich metadata.

Na chunkserveru jsou již fyzicky uložena data. Ta jsou rozdělena na části o délce 64MB, tzv. chunky. Každý chunk je identifikován 64bitovým číslem, které je

mu přiřazeno v okamžiku vytvoření master serverem. Chunky jsou skladovány na chunkserevru jako běžné linuxové soubory, to znamená, že GFS využívá pro fyzické uložení dat linuxové file systémy.

4. Souborové systémy výměnných medií

4.1 Diskety

4.1.1 FAT 12

Systém souborů FAT 12, který je v dnešní době využíván pouze na disketách, má shodnou architekturu se systémem FAT 16, který je používán na pevných discích. Jediným rozdílem je, že k adresaci clusterů je v alokační tabulce použito pouze 12 bitů. Z této hodnoty vyplývají omezení pro tento systém.

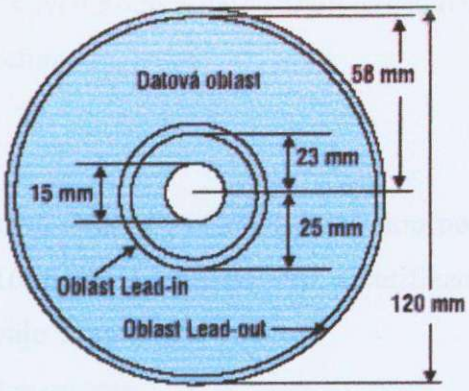
4.2 CD a DVD

4.2.1 Rainbow Books

Co a jak lze zapsat na CD, obsahuje soubor standardů nazvaný Rainbow Books (Barevné knihy). Tento soubor obsahuje také definici, jaké má mít CD rozměry. Je zde popsáno několik druhů CD, které jsou roztříděné podle typu dat, případně podle typu CD (CD-R, CD-RW..).

4.2.1.1 Red book – „Červená kniha"

Je první knihou celé série standardů a obsahuje specifikaci digitálních audio CD (CD-DA Digital Audio), kódování audio dat na CD a fyzikálních rozměrů CD. Byl zaveden firmami Philips a Sony v roce 1980. Audio data jsou uložena ve stopách („track"), které jsou dále rozděleny na sektory. Každý sektor obsahuje 2352 bytů dat. Dále sektory obsahují dvě vrstvy opravných kódů EDC/ECC (Error Detection Code/Error Correction Code) a subcode kanály. Opravné kódy se použijí v případně, že přehrávač není schopen určitý sektor přečíst, tak se jeho hodnota jednoduše dopočte. Subcode kanály P-W obsahují informace, zda jsou následující data audio nebo digitální data (P), časové informace relativní i absolutní (Q) a grafické informace (R-W). Disk je podle tohoto standardu rozdělen na 3 části. Lead-in, datová část a Lead-out. Disk může obsahovat maximálně 99 stop v celkové délce až 76 minut.



Obr. 4.2.1.1.1 Popis CD

Lead-In obsahuje digitální „ticho" v hlavním kanálu a v subcode kanálu Q tzv. TOC (Table Of Contents – tabulku záznamů). Tato oblast je také určena pro kalibraci a nastavení čtecího laseru. Datová část obsahuje záznamy a oblast Lead-Out definuje konec datové části a obsahuje digitální ticho nebo nulová data.

Red book dále obsahuje definici CD + G a CD Text. CD + G představuje rozšíření CD-DA a dovoluje na CD umístit grafická a textová data, která jsou zobrazována současně s přehráváním hudby. Tato data jsou umístěná v subcode kanálech R-W, která poskytují pouze malý prostor (asi 3%). Formát CD + G dále ještě obsahuje dva podformáty CD – MIDI s instrukcemi pro MIDI zařízení a CD – User, které dovoluje tvůrci CD vložit instrukce specifické pro danou aplikaci. Formát CD Text je dalším rozšířením CD-DA, který umožňuje přidat textové informace o celém CD nebo o jednotlivých stopách do subcode kanálů R-W. Oblast Lead-In potom obsahuje informace týkající se celého disku a jednotlivých stop. V záznamovém poli jsou informace o aktuální stopě. Data jsou kódována jako znaky, aby se dosáhlo maximální efektivity. CD Text je kompatibilní se standardem ITTS (Interactive Text Transmission System), což znamená, že textové informace mohou být zobrazeny na přehrávačích, které tento formát podporují.

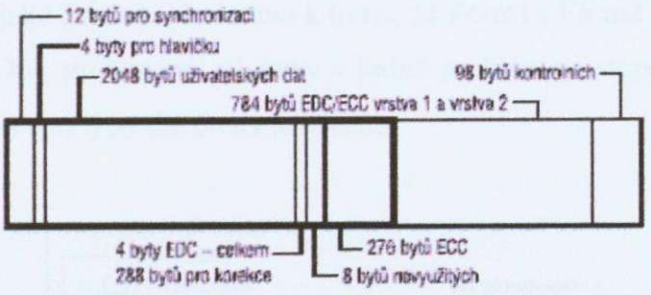
4.2.1.2 Yellow book – „Žlutá kniha"

Specifikace pro CD-ROM byla vyvinuta firmami Philips a Sony a vydána v roce 1983 jako rozšíření CD pro ukládání počítačových dat. Základem specifikace byla červená kniha, ale žlutá rozšířila původní definici o možnost ukládání digitálních

dat včetně korekčních a opravných kódů a dále rozšíření standardu CD-ROM – CD-ROM XA (eXtended Architecture)

CD-ROM mode 1

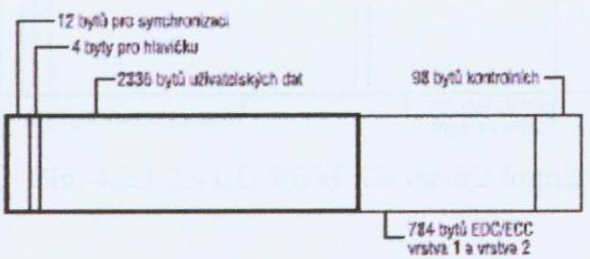
Používá se pro počítačová data se zvýšenou kontrolou pomocí vložení dalších opravných kódů. Prvních 16 bytů je použito pro identifikaci čteného sektoru a posledních 288 bytů představuje 3. opravnou vrstvu.



Obr. 4.2.1.2.1 CD-ROM mode 1

CD-ROM mode 2

Používá se pro komprimovaná audio a video data, později předefinována jako XA, kde není tak vysoký požadavek na korekci chyb. Využívá pouze CIRC chybové korekce. Tento formát zápisu není prakticky používán, neboť vyžaduje další hardware k přečtení dat. Formát neobsahuje 3. opravnou vrstvu, proto není vhodné jej používat pro počítačová data.



Obr. 4.2.1.2.2 CD-ROM mode 2

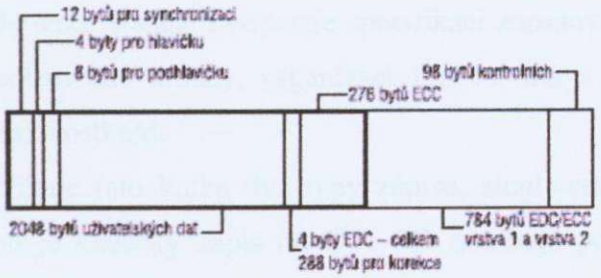
CD-ROM mixed mode

CD-ROM mixed mode obsahuje jak data, tak audio stopy. První stopa je datová, za ní následuje jedna nebo více audio stop. Většina přehrávačů CD rozpozná datové a audio stopy, ale na uživateli je, aby přehrál pouze audio. Při přehrávání datové stopy ji přehrávač interpretuje jako digitální šum, nebo jako ticho. V prvním případně může dojít k poškození reproduktorů. Tento formát není již používán k produkci audio CD,

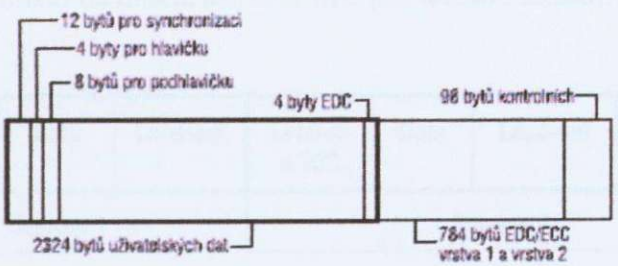
neboť byl nahrazen formátem CD-Extra (CD - Plus). Nevýhodou tohoto formátu je, že není možné přehrávat digitální a audio data současně, protože jsou fyzicky uložena na různých místech.

CD-ROM XA

Byl standardizován v roce 1989 firmami Philips, Sony a Microsoft jako rozšíření formátu CD-ROM. Formát umožňuje uložit počítačová a audio data ADPCM (ADaptive Pulse Code Modulation) do jedné stopy. Počítačová data jsou uložena jako Form1 a audio data jako Form2. Vzhledem k tomu, že Form1 i Form2 jsou ve formátu CD-ROM mode 2, XA mohou být uloženy v jedné prokládané stopě. Na rozdíl od CD-ROM mixed jsou oba typy dat čteny současně.



Obr. 4.2.1.2.3 CD-ROM XA mode2 form1



Obr. 4.2.1.2.4 CD-ROM XA mode2 form2

Aby bylo možno přečíst disky zapsané v tomto formátu, je nutný další hardware k oddělení počítačových a audio dat. Tento formát je používán u CD-I Bridge, Photo CD, VideoCD a disků pro Sony Playstation.

4.2.1.3 Green book – „Zelená kniha"

V této knize jsou popsána CD-I (CD-Interactive). Dále obsahuje specifikaci dat dle ISO 9660, audio dat dle ADPCM úrovní A, B, C, video dat v reálném čase a operačního systému CD-RTOS (Compact Disc Real Time Operating System). Tato

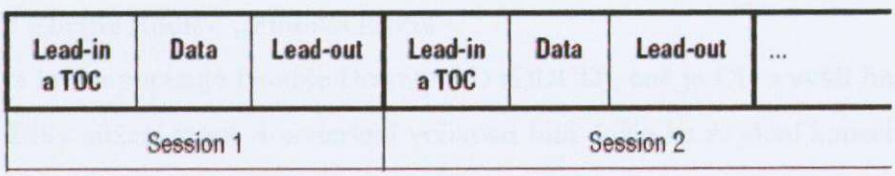
CD byla standardizována v roce 1987 firmou Philips. Jde o obdobu disků CD-ROM XA s prokládanými počítačovými a audio daty. Tato CD je možné přehrát pouze ve speciálních CD-I přehrávačích z důvodu nutnosti synchronizace hudby, videa i textu. Na tato CD se vejde až 19 hodin audio dat, 7500 statických obrázků nebo 72 minut videa v komprimovaném formátu MPEG.

Další formáty CD-i jsou CD-Bridge, CD-I MPEG 1 a CD-I Ready. Tyto formáty mají shodnou definici uložení sektorů.

4.2.1.4 Orange book – „Oranžová kniha"

Tato kniha je rozdělena do tří částí. První část popisuje zápis na magneto-optické disky CD-MO (Magneto-Optical), druhá část popisuje zápis na CD-R a třetí zápis na CD-RW. Dále tento standard popisuje specifikaci zapsaných a nezapsaných disků, specifikaci předlisované drážky, organizaci uložení dat a jejich spojování, doporučení měření odrazivosti atd.

Pro CD-R specifikuje tato kniha dva typy zápisu, single-session nebo multi-session. Single session je klasický zápis na CD. CD obsahuje pouze jednu oblast Lead-In, datovou oblast a jednu oblast Lead-Out. Při zápisu CD jako multi-session má každé session svou vlastní Lead-In, datovou i Lead-Out oblast. Lead-In a Lead-Out oblasti zabírají určité místo na disku, asi 13,5 MB pro každou session.



Obr. 4.2.1.4.1 Multi-session CD

4.2.1.4 White book – „Bílá kniha"

Bílá kniha byla vydána v roce 1987 firmami Sony, Philips, Matsushita a JVC jako standard pro ukládání videa CD. Tento nový formát se nazývá Video Compact Disc (VCD). Data jsou zapsána na disk v souborovém systému ISO 9660 ve formě CD-ROM mode 2 XA (viz. Žlutá kniha) a maximální délka činí 74 minut. Audio data jsou ve formátu MPEG-1 layer 2 s datovým tokem 224kbps (Kilobits per second – kilobitů za sekundu), vzorkovací frekvencí 44,1 kHz a dvěma kanály (stereo). Video data jsou uložena ve formátu MPEG-1. Pro NTSC video má rozlišení 352x240 bodů a

29,97 snímků za sekundu (frame per second – fps), pro PAL video má rozlišení 352x288 a 25 fps. Datový tok se pohybuje okolo 1,13 Mbps.

4.2.1.5 Blue Book – „Modrá kniha"

Modrá kniha specifikuje Enhanced CD (známý také jako CD-Extra, či CD Plus) a CD Plus Graphics (CD+G). CD-E je multi-session disk, který obsahuje v první části audio a ve druhé data ve formátu CD-ROM XA. CD+G obsahuje audio data a grafická data. Takovýto disk lze přehrát v klasickém CD přehrávači, ale pokud je přehrán v přehrávači CD+G připojeném k televizi či počítačovému monitoru, zobrazují se grafická data zároveň při přehrávání audio stop. Grafická data jsou obvykle texty písní, informace o autorovi, skladbě, CD a podobně. Tato CD se používají jako Karaoke disky.

4.2.1.6 Beige Book – „Béžová kniha"

Béžová kniha byla vydána v roce 1992 a popisuje Photo CD od firmy Kodak. Na Photo CD se vejde téměř 100 fotek ve vysokém rozlišení. Data jsou zapsána ve formátu CD-ROM XA. Fotky z tohoto CD lze přímo vytisknout pomocí speciálního zařízení od Kodaku. Tento formát byl nahrazen formátem Picture CD také od firmy Kodak.

4.2.1.7 Purple Book – „Fialová kniha"

Fialová kniha popisuje Double Desinty CD (DDCD), což je CD s vyšší hustotou zápisu dat. Díky zúžení stopy a zmenšení velikosti bitů došlo ke zvýšení kapacity CD téměř na dvojnásobek, na 1,3 GB. Pro tato CD jsou zapotřebí zvláštní mechaniky, jak pro čtení, tak i pro zápis. V dnešní době jsou DDCD vcelku zbytečná, když lze zapisovat na DVD o trojnásobné velikosti, než mají DDCD.

4.2.1.8 Scarlet Book – „Šarlatová kniha"

Tento standard vytvořily firmy Philips a Sony v roce 1999. Šarlatová kniha definuje formát Super Audio CD (SACD). Obsahuje tři různé typy CD, jednovrstvé, dvouvrstvé, nebo dvouvrstvé hybridní CD. Je to dvouvrstvé CD zpětně kompatibilní s CD-DA. Zvuk se nahrává pomocí metody Direct Strem Digital (DSD). DSD je metoda, která umožňuje převod analogového signálu na 1 bitový signál pro zápis.

Jedná se o bezztrátovou kompresi. Vzorkovací frekvence DSD je 2,8 MHz a umožňuje frekvenční rozsah až 100 kHz a dynamický až 120 dB.

4.2.2 High Sierra

Dříve každý operační systém používal svůj přirozený souborový systém pro zápis na CD, což sice nevadilo pro bootovací CD se systémem, ale způsobovalo velké potíže při přenosu dat z jednoho systému na jiný. Proto vznikl systém High Sierra, který definoval uložení dat na CD v husté sekvenci. Byl to hierarchický souborový systém s osmi úrovněmi vnoření. Podporoval zápis pouze nejdůležitějších atributů, jako čas zápisu, rozlišení souboru a adresáře, jméno, příponu a verzi. A pro každý soubor byla definována systémová oblast, která byla určena pro další rozšíření atributů.

4.2.3 ISO 9660

ISO 9660 je postaven na systému High Sierra a jeho vylepšeních. Na začátku každého disku s formátem ISO 9660 se nachází systémová oblast určená jako boot sektor u bootovacích disků. Za touto systémovou oblastí se nachází volume descriptor (popisovač svazku). Skládá se ze dvou částí, typu descriptoru a charakteristiky decriptoru.

Hodnota descriptoru	Typ descriptoru
0	Boot Record
1	Primary volume descriptor
2	Supplemtary volume descriptor
3	Volume partition descriptor
255	Volume descriptor set terminator

Tab. 4.2.3.1 Označení a typ descriptoru

Každý descriptor má na disku ještě svojí záložní kopii. Díky tomuto uspořádání může program tento popisovač přeskočit, pokud mu nerozumí nebo pokud je daný popisovač poškozen, a pokračovat dále ve čtení disku, dokud nenarazí na pro něj srozumitelný popisovač. To také umožňuje používat několik různých typů descriptorů současně a disk se může skládat z několika druhů obsahu.

Nejmenší možná velikost svazku je omezena velikostí primary descriptoru, který popisuje souborový systém, a ending descriptoru, který má podobu tabulky proměnné velikosti obsahující informace o dalších použitých popisovačích.

Primary volume descriptor je podobný svým obsahem a funkcí superbloku v souborových systémech Unixu. Obsahuje informace o rozdělení disku systémem ISO 9660. Jeho součástí je také odkaz na souvislý blok obsahující kořenový adresář. Tento souvislý blok obsahuje všechny adresářové položky nacházející se v rootu. Zbylé adresářové položky jsou uloženy také v primary volume desriptoru. Je zde ještě také uložena informace o velikosti logických bloků, na které je CD rozděleno, a položka Volume Space Size, která udává velikost místa na CD.

Souborové atributy v ISO 9660 jsou celkem jednoduché. Nejdůležitějším atributem je ten, který rozlišuje, zda je daný objekt soubor nebo adresář. Atributy souborů jsou uloženy v adresářových položkách adresářů, ke kterým dané soubory náleží, případně v rozšířených atributových záznamech.

Vyhledávání souborů v ISO 9660 může probíhat dvěma způsoby. Buď se projde celá adresářová struktura, nebo se použije tabulka cest. V této tabulce jsou vypsány všechny položky a jim odpovídající soubory. Použitím této tabulky se značně zkrátí doba vyhledávání.

Úrovně standardu ISO 9660

Standard ISO 9660 má několik úrovní omezení. ISO 9660 Level 1 má omezenu maximální délku jmen souborů na 8+3, osm znaků ve jméně, tři na příponu. Je také omezena maximální hloubka adresáře na osm úrovní. U ISO 9660 Level 2 byl zvýšen maximální počet znaků ve jméně na 31 a ISO 9660 Level 3 přidává podporu pro fragmentovaný zápis souborů, to se vyskytuje hlavně při paketovém zápisu. Všechny 3 úrovně omezují znaky použitelné ve jméně na velká písmena, číslice a podtržítko. Také je omezena maximální úroveň vnořených adresářů na osm.

Tato omezení lze obejít pomocí rozšíření tohoto standardu. Rozšíření se nazývají Romeo, Joliet, Rock Ridge a El Torito.

4.2.3.1 Joliet

Rozšíření Joliet pochází od firmy Microsoft a podporuje názvy dlouhé až 64 znaků. V DOSu bude viditelný zkrácený název souboru (6 znaků, vlnovka a číslo). Ve jménech se mohou vyskytovat znaky Unicode mezinárodní znakové sady a také mezery.

4.2.3.2 Romeo

Rozšíření Romeo podporuje názvy o délce 128. Ve jménech se mohou vyskytovat i mezery. Ale disk s tímto systémem je čitelný pouze ve Windows, protože neexistuje předpis pro zkrácení jména souborů.

4.3.2.3 Rock Ridge

Rozšíření Rock Ridge (RRIP – Rock Ridge Interchange Protocol) přidává podporu pro normu POSIX (Soubor standardů pro API v Unixových systémech). Umožňuje ukládat na CD i informace specifické pro Unixové souborové systémy, jako jsou vlastník souboru, skupiny a další. Také zvětšuje maximální délku názvu souborů na 255 znaků. Podporuje symbolické linky a větší adresářovou hloubku.

4.3.2.4 El Torito

Rozšíření El Torito umožňuje uživateli nabootovat počítač z CD. Poprvé bylo představeno v lednu 1995.

Poslední verzí tohoto standardu je ISO 9660:1999, který ruší omezení maximální hloubky vnořených adresářů. Místo toho zavádí maximální délku cesty na 207 znaků. A odebírá tečce její speciální význam ve jménech souborů (oddělovač jména souboru a jeho přípony).

Podpora pro multisession (vícesekční) CD je zavedena ve standardu ISO 13490, který je postavený na ISO 9660.

4.3.3 UDF – Universal Disk Format

UDF je standardem pro zápis dat na optická média a vychází z normy ISO/IEC 13346, která je známá také jako ECMA-167. Je náhradou za standard ISO 9660. Původně byl UDF definován pro ROM (Read Only Memory) média a pro média,

která jsou omezená počtem přepisů, jako magneto-optické disky, DVD disky a CD-R a CD-RW. Později byla přidána podpora pro flash disky, Iomega REV a CD-MRW.

UDF se vyskytuje v několika verzích, přičemž každá verze přidává podporu pro jiné typy disků.

- UDF 1.00 – první verze UDF, vydaná 24.10. 1995
- UDF 1.01 – vydán 3.11. 1995, základní podpora DVD
- UDF 1.02 – 30.8. 1996, někdy se označuje jako UDF Bridge format (DVD jsou definována pomocí UDF i ISO 9660), oprava podpory DVD
- UDF 1.50 – 7.2. 1997, packetový zápis na CD-R/RW, podpora DVD-RAM, nikoli však jejich plná kompatibilita (nejdou přehrát ve stolních DVD přehrávačích), limitovaná podpora Windows 95/NT.
- UDF 2.00 – 3.4. 1998, přidány streamy, ACL (Access Control Lists) a podpora pro Power Calibration (Nastavení optimální energie pro vypálení CD-R/RW, provádí se v oblasti v blízkosti středu média, ještě před samotným zápisem dat). DVD jdou přehrát jak v počítači, tak ve stolním přehrávači. Objevuje se jako standard ve Win 95 (OSR2), Win 98, Win ME, Win 2000 i Win XP.
- UDF 2.01 – 14.3. 2000, upřesnění ohledně nečitelných sektorů, podpora pro real time soubory.
- UDF 2.50 – 30.4. 2003, podpora pro zápis metadat, doporučení ohledně DVD+R/RW a DVD-R/RW/RAM
- UDF 2.60 – 1.3. 2005, doporučení pro Blu-Ray a HD-DVD.

4.3.3.1 Mount Rainier

Mount Rainier je rozšířením pro UDF. Přidává podporu paketového zápisu na disk. Díky tomu lze používat CD-RW jako diskety, ale je k tomu zapotřebí speciální program.. Pro Mount Rainier se nemusí používat žádná zvláštní CD či DVD, funguje se všemi přepisovatelnými médii, pouze jej musí podporovat mechanika. Mount Rainier je nezávislý na operačním systému.

Tento standard používá několik posledních sektorů na správu chyb. Chybné sektory jsou zaznamenány v tabulce v lead-in sektoru, v administrativní oblasti, a její kopie je umístěna v lead-out sektoru.

Mount Rainer je podporován posledními verzemi Linuxu a měl by být podporován i Windows Vista. Pokud operační systém nepodporuje formát Mount Rainier, je zapotřebí jiného software pro čtení i zápis takového média.

5. Vytvoření vlastního file systému

Nejprve je nutné vytvořit návrh souborového systému, přitom je vhodné vycházet z klasického linuxového systému. Při návrhu bychom se měli vyhnout fragmentaci souborů, nebo se jí alespoň snažit udržet co nejmenší, co nejvíce snížit plýtvání nevyužitým místem a optimalizovat výkon FS. Vhodné je také se vyhnout velkému počtu omezení a počítat s možným rozšířením v budoucnu.

V první řadě je nutné vytvořit a nadefinovat struktury uložené na disku. Jedná se o popis super bloku *struct xxx_super_block*, popis inod *struct xxx_inode*, popis deskritptoru cylindrické skupiny *struct xxx_group_descriptor* a popis adresářové struktury *struct xxx_dir_entry*. Tyto struktury musí být uloženy v souboru *include/linux/xxx_fs.h*, kde xxx je jméno nového souborového systému. Dále je potřeba vědět, které údaje z námi vytvořených struktur lze načíst do struktur generických. Generické struktury se nalézají v *include/linux/fs.h*. Na tato data lze nahlížet jako na data nezávislá na souborovém systému. Pro struktury závislé na souborovém sytému je nutné vytvořit zvlášťní struktury. Struktura *struct xxx_sb_info* (informace o super bloku) bude uložená v *include/linux/xxx_fs_sb.h* a struktura *struct xxx_inode_info* (informace o inodách) bude v *include/linux/xxx_fs_i.h*. Dále je ještě nutné přidat do souboru *include/linux/fs.h* makra *#include <linux/xxx_fs_sb.h>* a *#include <linux/xxx_fs_sb.h>*. Tato makra zpřístupní náš souborový systém. Struktury *struct xxx_sb_info* a *xxx_inode_info* přidáme do unionů na konci příslušných generických struktur. Nyní si vytvoříme adresář *fs/xxx*. Nejjednodušším způsobem, jak začít programovat svůj vlastní systém, je vzít již vytvořený linuxový systém a provádět na něm modifikace. Naše práce bude jednodušší, pokud má být náš systém pouze pro čtení (read-only), ale je dobré do budoucna počítat i s možností zápisu. V adresáři *fs/xxx* vytvoříme struktury popisující chování našeho file systému. Tyto struktury obsahují ukazatele na funkce závislé na souborovém systému, jsou to *file_system_type, file_operations, inode_operations, super_operations*. Není nutné

vždy implementovat všechny tyto operace, protože v některých případech lze využít již existující generické funkce z jádra linuxu. Pro umožnění zápisu je nutné implementovat alokaci a uvolňování bloků a inod. Nyní už zbývá pouze odladit systém, začlenit jej do jádra a přidat podporu pro uživatele, příkazy *mkfs, fsck.*
Tento návod jsem převzal z http://www.manualy.sk/seminar/Papers97/linuxfs/.

Závěr

Nelze jednoznačně určit nejlepší systém souborů, lze jen určit, že daný systém je optimální pro danou úlohu. Při výběru souborového systému pro svůj počítač by měl každý uživatel zvážit, k čemu bude počítač používat a jak cenná data bude uchovávat na svém disku. Samozřejmě, že běžný uživatel, který užívá počítač k prohlížení internetových stránek a občas k psaní nějakých dopisů, nebude vyžadovat používaní nejnovějšího šifrovaného systému souborů, ani nebude vyžadovat zajištění 100% konzistence dat za všech okolností. Kdežto počítače používané v bankách, nemocnicích, letištích a velkých firmách budou potřebovat to nejlepší zabezpečení a nulovou ztrátu dat, stejně jako rychlý přístup k informacím a jejich sdílení mezi jednotlivými odděleními a pobočkami. To nelze zajistit jenom výběrem správného souborového systému, ale lze tomu v určité míře napomoct. Výběr souborového systému závisí také na operačním systému, se kterým daný počítač pracuje. Pokud počítač používá operační systém Windows, je výběr souborového systému značně omezen (FAT32x, NTFS, v budoucnu WinFS). Naopak pokud je operačním systémem Unix (Linux), má uživatel k dispozici nepřeberné množství file systému, v němž ale není tak jednoduché se zorientovat.

Jsem si vědom toho, že jsem zdaleka nepopsal všechny souborové systémy a všechny možnosti jejich nastavení a použití. Cílem této práce bylo alespoň postihnout vývoj file systémů od jejich prvopočátků až po žhavé novinky současného vývoje, jako WinFS a ZFS, a pomoci ostatním uživatelům při výběru souborového systému. Doufám, že se mi to podařilo a že tato práce bude přínosem i pro ostatní.

Seznam použité literatury

[1] http://www.sun.com

[2] http://www.wikipedia.org

[3] http://meta.cesnet.cz/cs/docs/software/system/afs.html

[4] http://www.abclinuxu.cz

[5] http://www.zive.cz

[6] http://www.c-sharpcorner.com

[7] http://www.linktionary.com

[8] http://www.slider.com

[9] http://www.cse.unsw.edu.au

[10] http://www.psc.edu

[11] http://www.reference.com

[12] http://www.osta.org

[13] http://www.kernelthread.com

[14] http://www.linuxplanet.com

[15] http://meta.cesnet.cz

[16] http://www.cs.cmu.edu

[17] http://www.root.cz

[18] http://www.umel.feec.vutbr.cz

[19] http://mnemec.kuceraipc.de

[20] http://www.osxzone.com

[21] http://www.os2.cz

[22] http://fdmech.fd.cvut.cz/nastenka/Linux-seminar/reports/NFS/

[23] http://jt.sf.cz

[24] http://www.manualy.sk

[25] http://www.cdr.cz

[26] Ondřej Čada, Operační systémy, Grada 1994

Přílohy

Název souborového systému	Autor	Rok vydání	Operační systém
FAT16	Microsoft	1983	MS-DOS 2.0
FFS	Kirk McKusick	1983	4.2BSD
MFS	Apple Computer	1984	Mac OS
HFS	Apple Computer	1985	Mac OS
HPFS	IBM & Microsoft	1988	OS/2
JFS	IBM	1990	AIX
NTFS	Microsoft, Gary Kimura, Tom Miller	1993	Windows NT
ext2	Rémy Card	1993	Linux
UFS1	Kirk McKusick	1994	4.4BSD
XFS	SGI	1994	IRIX
UDF	ISO/ECMA/OSTA	1995	-
FAT32	Microsoft	1996	Windows 95b
HFS Plus	Apple	1998	Mac OS 8.1
ext3	Stephen Tweedie	1999	Linux
ReiserFS	Namesys	2001	Linux
ZFS	Sun Microsystems	2004	Solaris

Tab.: Přehled file systémů s rokem vydání, autorem a původním (nativním) OS

Operační systém	Maximální délka názvu souboru	Přípustné znaky	Maximální délka cesty	Maximální velikost souboru	Maximální velikost svazku
FAT12	255 bytes	Any Unicode except NUL	No limit defined	32MB	1MB to 32MB
FAT16	255 bytes	Any Unicode except NUL	No limit defined	2GB	16MB to 2GB
MFS	255 bytes	Any byte except :	No path (flat filesystem)	256MB	256MB
HFS	31 bytes	Any byte except :	Unlimited	4GB	2TB
FAT32	255 bytes	Any Unicode except NUL	No limit defined	4GB	512MB to 2TB
HPFS	255 bytes	Any byte except NUL	No limit defined	4GB	2TB
NTFS	255 characters	Any Unicode except NUL	32,767 Unicode characters with each path component (directory or filename) up to 255 characters long	16EB	16EB
HFS Plus	255 UTF-16 characters	Any valid Unicode	Unlimited	8EB	8EB
FFS	255 bytes	Any byte except NUL	No limit defined	4GB	256TB
UFS1	255 bytes	Any byte except NUL	No limit defined	4GB to 256TB	256TB
UFS2	255 bytes	Any byte except NUL	No limit defined	512GB to 32PB	1YB
ext2	255 bytes	Any byte except NUL	No limit defined	16GB to 2TB4	2TB to 32TB
ext3	255 bytes	Any byte except NUL	No limit defined	16GB to 2TB4	2TB to 32TB
ReiserFS	4032 bytes/255 characters	Any byte except NUL	No limit defined	8TB	16TB
XFS	255 bytes	Any byte except NUL	No limit defined	9EB	9EB
JFS	255 bytes	Any byte except NUL	No limit defined	8EB	512TB to 4PB
UDF	255 bytes	Any Unicode except NUL	1023 bytes	16EB	?
ZFS	255 bytes	Any Unicode except NUL	No limit defined	16EB	16EB

Tab.: Přehled limitů jednotlivých souborových systémů