

Operační systémy

Souborové systémy

Operating systems - File systems

Pavel ŽÁČEK

Bakalářská práce
2005



Univerzita Tomáše Bati ve Zlíně
Fakulta Technologická

Poděkování

Chtěl bych vyjádřit poděkování vedoucímu mé bakalářské práce Ing. Martinu Syslovi, Ph.D. za jeho podnětné připomínky, náměty k realizaci a neocenitelnou pomoc během tvorby bakalářské práce.

Prohlašuji, že jsem na celé bakalářské práci pracoval samostatně a použitou literaturu jsem citoval.

Ve Zlíně, 06. 06. 2005

.....

Žáček Pavel

ABSTRAKT

Informační technologie jsou dnes nedílnou součástí života moderního člověka. Studenti se při studiu předmětu "Operační systémy" dozvídají velké množství nových informací, vždyť problematika předmětu, který má za úkol popsat fungování operačního systému a s ním úzce spojeného souborového systému je velmi obsáhlá. Čtenář této bakalářské práce si tak může své vědomosti poměrně hodně rozšířit díky přehlednému vypracování daného tématu. Tato studijní pomůcka předmětu "Operační systémy", která je předmětem této bakalářské práce, poskytuje informace o základních pojmech z oblasti souborových systémů jako jsou adresář, soubor a nebo způsob kódování použitý pro uložení názvu souboru. V druhé části jsou potom popsány vlastnosti jednotlivých souborových systémů.

ABSTRACT

Information technologies are entire part of life modern man. Students with study of subject "Operating systems" get many informations, which describe functions of operating systems and File system. This bachelor work is focused on general elaborating of above mentioned topic and provides informations about basic definitions from file systems as folder, file and coding of file names. There are describing of properties of file systems on the end of this bachelor work.

OBSAH

ÚVOD	6
1 HIERARCHIE DAT	7
2 OPERACE SE SOUBORY, ATRIBUTY	9
3 SOUBOROVÉ SYSTÉMY	11
3.1 PŘEDPOKLADY SOUBOROVÝCH SYSTÉMŮ	11
3.2 ADRESÁŘE	13
3.2.1 Šifrování a dešifrování	13
3.2.2 Jednoúrovňové uspořádání souborového systému	14
3.2.3 Hierarchické uspořádání souborového systému	14
3.2.4 Relativní cesta	15
3.2.5 Odkazy	16
3.3 METADATA	18
3.3.1 Souborové deskriptory	19
3.4 MOUNTOVÁNÍ	20
3.5 USPOŘÁDÁNÍ SOUBORU	22
4 SOUBOROVÉ SYSTÉMY OPERAČNÍHO SYSTÉMU MICROSOFT WINDOWS	23
4.1 SOUBOROVÝ SYSTÉM FAT	23
4.1.1 Hlavní spouštěcí záznam (MBR)	25
4.1.2 Primární a rozšířené oddíly souborového systému FAT	25
4.1.3 Spouštěcí záznam svazku (VBR)	28
4.1.4 Kořenový adresář	29
4.1.5 Alokační tabulky souborů (FAT)	30
4.1.6 Clustery (alokační jednotky)	31
4.1.7 Diagnostické cylindry pro čtení a zápis	32
4.2 SOUBOROVÝ SYSTÉM NTFS	33
4.2.1 Vývojové cíle a charakteristiky NTFS	33
4.2.2 Hlavní přednosti souborového systému NTFS	33
4.2.3 Logická struktura NTFS	36
4.2.4 Fyzická struktura NTFS	38
4.2.5 Shrnutí	45
5 SOUBOROVÉ SYSTÉMY NA BÁZI OPERAČNÍCH SYSTÉMŮ UNIX	46
5.1 SOUBOROVÝ SYSTÉM EXT2	47
5.1.1 Základní specifikace:	47
5.1.2 Symbolický odkaz	49
5.2 SOUBOROVÝ SYSTÉM EXT3	49
5.2.1 Charakteristika	49
5.2.2 Výkon	50
5.2.3 Budoucnost	50
5.3 SOUBOROVÝ SYSTÉM REISERFS	51
5.3.1 Parametry zrychlující práci FS	52
5.3.2 Změna velikosti souborového systému	52
5.3.3 Shrnutí	52
ZÁVĚR	53
SEZNAM POUŽITÝCH ZDROJŮ	54
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	55
SEZNAM OBRÁZKŮ	56
SEZNAM TABULEK	57

ÚVOD

Souborový systém je nedílnou součástí každého počítačového systému, lze tedy předpokládat, že přednášky předmětu "Operační systémy" zaujmou studenty oborů týkajících se informatiky ze všech předmětů zřejmě nejvíce. Tento předmět podává velké množství informací o softwaru operačních systémů a o principech fungování tak složité softwarové architektury jako je operační systém. Výklad látky na téma týkající se souborových systémů je pro studenty zcela jistě jedním z nejzajímavějších a do velké míry také užitečným, protože souborový systém potřebuje ze strany uživatele čas od času údržbu týkající se především kontroly chyb na disku a nebo taky neméně důležitou defragmentaci disku. U operačních systémů, kde není prováděna defragmentace, může docházet ke značnému zpomalení celého systému. Fragmentace souboru, neboli rozmístění daného souboru po částech na disku, je způsobena velkým množstvím kopírování a mazání souborů [1, 2, 8]. Tato studijní pomůcka obsahuje základní informace o souboru, adresáři, hierarchii dat, kódování, atributech souboru, operacích, které mohou být se soubory prováděny a nebo také o mechanismech a metodách, za které jsou souborové systémy zodpovědné. V části, týkající se souborového systému FAT je po rozdělení podle variant pojednáno o společných rysech jako jsou struktury a oblasti na pevném disku, hlavní spouštěcí záznam, primární a rozšířené oddíly s tabulkou rozdělení disku a nebo také co se skrývá pod pojmem cluster. Podrobněji je v této bakalářské práci diskutován dnes zřejmě nejpoužívanější souborový systém NTFS, který je doporučen firmou Microsoft pro použití v operačních systémech s jádrem NT (Windows NT/2000/XP). Souborový systém NTFS je pro své vlastnosti jako jsou možnost kvótování, šifrování a nebo zvýšené odolnosti proti haváriím oproti souborovému systému FAT32, nemalým krokem vpřed. Na závěr jsou popsány základní vlastnosti některých souborových systémů na bázi operačních systémů UNIX.

Pochopení fungování souborového systému může být jednou z důležitých částí výuky studentů informačních technologií, protože může předejít některým chybným krokům, kterých se může neinformovaný uživatel běžně dopustit. Tato studijní pomůcka by měla pomoci nejen studentům, ale i vyučujícím, a to především v době svého vzniku, kdy jsou informace aktuální. Ovšem vzhledem k rychlosti rozvoje souborových systémů lze předpokládat nemalou "trvanlivost" informací obsažených v této bakalářské práci.

1 HIERARCHIE DAT

Informace jsou v počítačových systémech uloženy podle **Hierarchie dat**. Nejnížší úroveň hierarchie dat je složena z bitů. Bity jsou sestaveny ve **vzorech bitu** k reprezentování všech datových položek v počítačových systémech. Může být použito až 2^n možných bitových vzorů pro řetězec s n bity.

Další úroveň v hierarchii dat je pevná délka vzoru bitů nazývaná **byte**, **char** (znak) a **word** (slovo). Při ukládání obsahuje byte 8 bitů. **Word** je množství bitů, které procesor může zpracovat najednou. Takže, word jsou 4 byty na 32 bitovém procesoru a 8 bytů na 64 bitovém procesoru.

Mapa znaků (nebo skupiny bytů) jsou symboly, jako např.: písmena, čísla, interpunkce a nové řádky. Mnoho systémů používá 8-bitových znaků a potom mají 2^8 , neboli 256 možných znaků ve **znakové sadě**. Tři nejvíce populární znakové sady, které se dnes používají jsou:

- **ASCII (American Standard Code for Information Interchange)**
- **EBCDIC (Extended Binary-Coded Decimal Interchange Code)**
- **Unicode®**

ASCII ukládá znaky jako 8-bitový byte a tak může mít 256 možných znaků ve znakové sadě. Protože **ASCII** má malý počet znaků, tak zde není podpora mezinárodní znakové sady.

EBCDIC je často využíván sálovými počítači, zvláště systémy vyvinuté firmou IBM; taky se zde ukládají znaky jako 8-bitový byte.

Unicode® je mezinárodně uznaný standard, který je populární v síti Internet a ve vícejazyčných aplikacích. Jeho cílem je používat jedinečné číslo k reprezentování každého znaku ve všech světových jazycích. **Unicode®** poskytuje 8-, 16- a 32-bitové vyjádření jeho znakové sady. To zjednodušuje konverzi z ASCII k Unicode znakům, Osmý-bit reprezentace bitu Unicode, nazývaný **UTF - 8 (Unicode character set Translation Format - 8 bit)**, odpovídá přímo znakové sadě **ASCII**. HTML soubory používají kódování **UTF-8, UTF-16 a UTF-32**, které se postarají o každou větší znakovou sadu a umožní aplikacím ukládání informací obsahující znaky rozmanitých abeced, jako jsou Řecká, Cyrilice, Čínská a mnoho jiných. Nicméně, aby se vyrovnal UTF-8, je potřeba

větších souborů k ukládání stejného počtu znaků. Například, 12-znakový řetězec "Hello, World" vyžaduje 12 bytů na ukládání použitím 8-bitových znaků, 24 bytů použitím 16-bitových znaků a 48 bytů použitím 32-bitových znaků.

Pole (array) je skupina znaků (např.: jméno osoby, adresa ulice nebo telefonní číslo). **Záznam** je skupina polí. Záznam studenta může obsahovat, například, samostatná pole pro identifikační číslo, jméno, adresu, telefonní číslo, bodový průměr, studijní obor, očekávané datum dokončení studia a tak dále. Soubor je vlastně skupina příslušných záznamů. Například, "studentský" soubor by mohl obsahovat jeden záznam pro každého studenta v univerzitě a nebo např. soubor výplatní listiny by mohl obsahovat jeden záznam pro každého zaměstnance ve společnosti. Nejvyšší úroveň hierarchie dat je souborový systém nebo databáze. Souborové systémy jsou sbírky souborů a databáze je sbírka dat.

Termín **svazek** je jednotka pro ukládání dat, která může obsahovat různé soubory. Fyzický svazek je omezen na jedno zařízení pro ukládání; a logický svazek je takový, jaký může být použit ve virtuálním stroji, tzn. může být rozprostřen na mnoho zařízeních, které jsou fyzicky rozdílná. Jako příklady svazků lze uvést disky CD, DVD, pásky a pevné disky.

2 OPERACE SE SOUBORY, ATRIBUTY

Soubor je vlastně název pro sbírku dat, se kterou mohou být prováděny následující operace:

- **Otevřít (Open)**- otevře existující soubor. Prohledání adresářové struktury a získání atributů souboru
- **Zavřít (Close)** - zavře otevřený soubor
- **Vytvořit (Create)** - vytvoří nový soubor
- **Smazat (Destroy)** - odstraní soubor
- **Kopírovat (Copy)** - kopíruje obsah jednoho souboru do jiného
- **Přejmenovat (Rename)** - změna názvu souboru
- **Zobrazit obsah (List)** - tiskne nebo zobrazí obsah souboru

Položky jednotlivých údajů uvnitř souboru mohou být vykonávány těmito operacemi:

- **Čtení (Read)** - kopíruje data ze souboru k paměti procesu
- **Zápis (Write)** - kopíruje data z paměti procesu k souboru
- **Úprava (Update)** - modifikuje existující datovou část v souboru
- **Vložení (Insert)** - přidá novou datovou část k souboru
- **Mazání (Delete)** - odstraní určitou datovou část od souboru

Soubory jsou charakterizovány těmito atributy:

- **Velikost (Size)** - množství dat v souboru
- **Umístění (Location)** - umístění souboru (cesta na paměťovém médium nebo v logickém systému souborů)
- **Přístup (Accessibility)** - omezení přístupu k datům v souboru.

- **Typ (Type)** - způsob, jak jsou data souboru používána. Například, spustitelný soubor obsahuje strojové instrukce pro proces. Datový soubor může upřesnit aplikaci, jak má přistupovat k jeho datům
- **Proměnnost (Volatility)** - frekvence, s jakou je přidáním nebo vymazáním jednotlivých částí dat tvořen soubor.
- **Aktivita (Activity)** - procento, kdy byly záznamy souboru zpřístupněny během daného časového období.

Soubory se skládají z jednoho nebo více záznamů. **Fyzický záznam** (nebo taky **fyzický blok**) je jednotka informace čtená nebo zapisovaná z paměťového média.

Logický záznam (nebo **logický blok**) je sbírka dat, se kterou se zachází jako softwarovou jednotkou. Když každý fyzický záznam obsahuje přesně jeden logický záznam, můžeme říct, že se soubor skládá z **neblokovaných záznamů (unblocked records)**. Když každý fyzický záznam obsahuje několik logických záznamů, můžeme říct, že se soubor skládá z **blokovaných záznamů (blocked records)**. V souboru s pevnou délkou záznamu, mají všechny záznamy stejnou délku a velikost bloku je základní násobek velikosti záznamu. V souboru s různě dlouhými záznamy, se mohou záznamy měnit na velikost, které dosahují až velikosti bloku.

3 SOUBOROVÉ SYSTÉMY

Z hlediska fyzického je základní technologií pro ukládání dat pevný disk či jiné médium. Z hlediska logického je však tímto základem **souborový systém (FS)**, neboť ten vytváří hierarchickou strukturu svazků a složek, do nichž jsou ukládány jednotlivé soubory. Současně FS uspořádává data tak, aby je byl schopen vyhledat na kterékoliv části disku. FS je obvykle integrovanou součástí operačního systému, přičemž mnohé novější operační systémy podporují několik různých souborových systémů [8]. Souborový systém vlastně organizuje soubory a řídí přístup k datům. Souborové systémy jsou zodpovědné za:

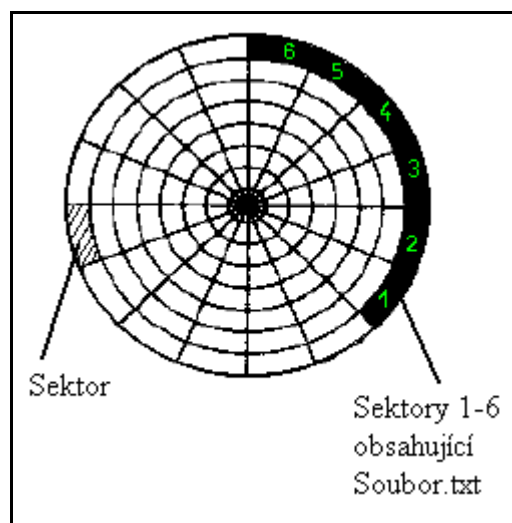
- **Správu souborů (File management)** - poskytuje souborům mechanismy, jako jsou ukládání, odkazy, sdílení a zabezpečení.
- **Správu zálohovacích médií (Auxiliary storage management)** - přiděluje prostor pro soubory na sekundárním nebo terciálním paměťovém médiu.
- **Souborové integritní mechanismy (File integrity mechanisms)** - zajišťují informace, že data uložené v souboru nejsou poškozena. Když je integrita souboru zajištěna, soubory obsahují jen informace, které by měli mít.
- **Přístupové metody (Access methods)** - zpřístupnění ukládaných dat.

Souborový systém je primárně spojen s řídicím sekundárním paměťovým prostorem, zvláště pak s diskovou pamětí. To pak může zpřístupnit data souboru uložená na jiných médiích (např.: v hlavní paměti).

3.1 Předpoklady souborových systémů

Souborové systémy umožní uživatelům vytvářet, modifikovat a mazat soubory. FS by měli přistupovat k souborové struktuře nejvhodnějším způsobem. Tzn., že pro každou aplikaci se zavede přenos dat mezi soubory. Uživatelé by měli být schopní sdílet soubory mezi sebou v pečlivě kontrolovaném a dodržovaném způsobu. Mechanismus pro sdílení souborů by měl poskytovat různé druhy řízeného přístupu jako jsou **přístup pouze pro čtení, přístup pouze pro zápis, přístup pouze pro spuštění** nebo různé kombinace zde uvedených.

Souborové systémy by měly zajišťovat **nezávislost zařízení** - uživatelé by měli být schopní odkazovat se na soubory **symbolickými jmény**, než použitím **fyzických názvů zařízení**. Symbolická jména jsou logická, uživatelsky přívětivá jména, jako MůjAdresář/MůjSoubor.txt. Fyzické názvy specifikují, kde se soubor nachází na daném zařízení, např.: disk 2, bloky dat 782-791. Symbolická jména dovolí uživateli **logický pohled** na souborový systém a jejich data tím, že použije jednoznačných jmen souborů a souborových operací. **Fyzický pohled** se týká rozložení dat souboru na konkrétním paměťovém médiu a zařízení specifických operací, které pracuje s daty (Obr.1.).



Obr. 1. Fyzická reprezentace souboru na disku

Navrhnout souborový systém vyžaduje znalost skupiny uživatelů, včetně množství uživatelů, průměrného množství a velikosti souborů na uživatele, vlastnosti aplikací, které v systému běží, a mnoho jiných. Tyto faktory musí být důsledně zváženy - to určuje nejvhodnější uspořádání souborů a adresářové struktury.

Kromě předcházení nahodilé ztrátě nebo zlomyslnému zničení informací, by souborové systémy měly také poskytovat **zálohovací** schopnosti, které usnadní vytvoření náhrady - kopie dat. Dále také **zotavovací** schopnosti, které umožní uživatelům obnovit některé ztracené nebo poškozené data. V citlivém prostředí, ve kterém musí být pro informace dodržena bezpečnost a soukromí, např. v systémech přenosu elektronických financí, systémech trestních rejstříků, systémech lékařských záznamů a jiných, by měl souborový systém poskytovat také **šifrovací** a **dešifrovací** schopnosti. Právě to dělá informace použitelné jen jistou skupinou lidí - tou, která vlastní klíče pro dešifrování.

3.2 Adresáře

Mějme systém s velkým měřítkem sdílení času (timesharing) a podporou rozsáhlého počtu uživatelů. Každý uživatel může mít několik účtů a každý účet může mít mnoho souborů. Některé soubory mohou být malé, např. e-mailové zprávy. Jiné soubory mohou být naopak velké, například databáze. To je běžné pro uživatelské účty obsahující stovky nebo dokonce tisíce souborů.

S komunitou několika tisíc uživatelů, by mohly systémové disky snadno obsahovat milióny souborů. Tyto soubory musí být zpřístupněny rychle bez omezení doby odezvy.

K správě a rychlému nalezení souborů, používají souborové systémy **adresáře**, které obsahují jména souborů a umístění jiných souborů v souborovém systému. Na rozdíl od souborů si adresář neukládá data uživatelů. V tabulce I je popsáno několik obecných polí adresářů a souborů.

Tab. I. Příklady polí u adresářů a souborů

Pole adresáře	Popis
Jméno	Znakový řetězec představující jméno souboru.
Umístění	Fyzický blok nebo logické umístění souboru v souborovém systému (tzn.: název s cestou).
Velikost	Počet bytů, které soubor zabírá.
Typ	Popis účelu souboru (např.: datový soubor nebo soubor adresáře).
Přístupová doba	Čas, kdy byl u souboru naposledy zaznamenán přístup.
Čas úpravy	Čas, kdy byl soubor naposledy modifikován.
Čas vytvoření	Čas, kdy byl soubor vytvořen.

3.2.1 Šifrování a dešifrování

Před lety byly informace zpracovávány počítači velmi omezené, a to na jeden lokální počítačový systém a přístup k němu mohl být kontrolován. Operačními systémy běžícími na dnešních počítačových systémech s velmi vysokým výkonem prochází ohromné množství informací na sítích mezi počítači, obzvláště pak na síti Internet.

Přenosová média jsou nezabezpečená a zranitelná, a proto software operačních systémů často poskytuje schopnosti šifrování a dešifrování. Obě tyto operace mohou být pro procesor tak náročné, že by to bylo nepraktické používat je ve velkém rozsahu, jako se

ukázalo již v minulosti. Ale s rostoucím výkonem procesorů bude mnoho systémů včleňovat šifrování a dešifrování do mnoho aplikací, obzvláště do těch, které zahrnují síťový přenos, jež je jak známo lehce napadnutelný jistou skupinou lidí, tzv. hackerů.

3.2.2 Jednoúrovňové uspořádání souborového systému

Nejjednodušší je uspořádání souborového systému používající **jednoúrovňové (single-level)** (nebo taky **flat**) struktury adresáře. V této realizaci uloží souborový systém všechny soubory použitím pouze jednoho adresáře. V jednoúrovňovém souborovém systému, nemohou mít dva soubory stejné jméno, protože jsou umístěny v jedné lokaci - **kořenu** a nebylo by možné jejich rozlišení. Většina oblastí však používá velké množství souborů a mnoho z nich má stejný název, a proto jsou jednoúrovňové souborové systémy používány velmi zřídka.

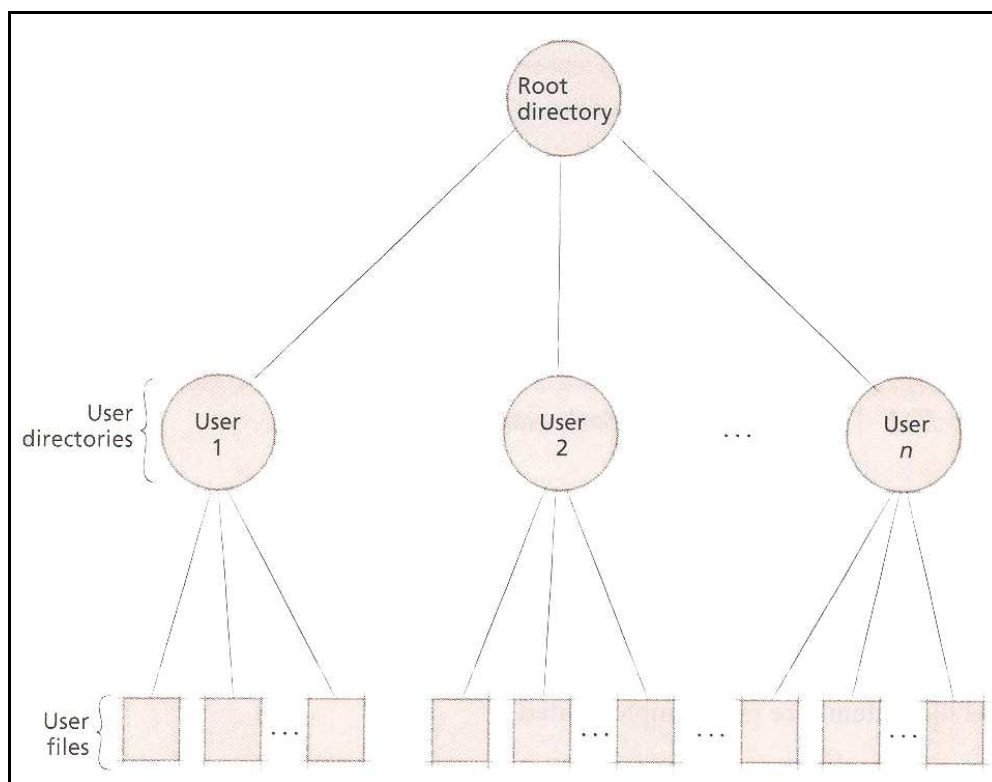
3.2.3 Hierarchické uspořádání souborového systému

Souborový systém může být organizován pomocí **hierarchického uspořádání** (Obr. 2.). **Kořen (root)** indikuje, kde na paměťovém médiu začíná **kořenový adresář (root directory)**. Adresáře jsou vlastně soubory, které ukazují na jiné adresáře a soubory. Na Obr. 2. kořenový adresář ukazuje na různé **uživatelské adresáře (user directories)**. Uživatelský adresář obsahuje vstup pro každý ze souborů konkrétního uživatele a každý záznam ukazuje na umístění odpovídající informace na paměťovém médiu.

Názvy souboru musí být jedinečné jen uvnitř daného uživatelského adresáře. V hierarchickém uspořádání souborového systému, může každý adresář obsahovat několik podadresářů, ale ne více než jeden rodičovský adresář. Název souboru je obvykle tvořen jako **název cesty (pathname)** od kořenového adresáře k souboru. Například, ve **dvouúrovňovém (two-level)** uspořádání souborového systému s uživateli Petr, Pavel a Jan, ve kterém Pavel má soubory VÝPLATY a FAKTURY, pak název s cestou pro soubor VÝPLATY je tvořen jako ROOT:PAVEL\VÝPLATY. V tomto příkladu, ROOT představuje kořenový adresář a použití lomítka (\) se ohraničí jednotlivé položky cesty.

Dnešní souborové systémy jsou realizovány většinou jako hierarchické souborové systémy, ale jméno kořenového adresáře a druhu oddělovače se může mezi souborovými systémy lišit. V souborových systémech OS Windows je kořenový adresář specifikován písmenem následovaným dvojtečkou (např.: C:) a souborové systémy na bázi UNIX

používají jako kořenový adresář lomítko (/). Systémy Windows používají jako oddělovač opačné lomítko (např.: *C:\Pavel\Výplaty*) a systémy na bázi UNIX používají lomítko (např.: */Pavel/Výplaty*).



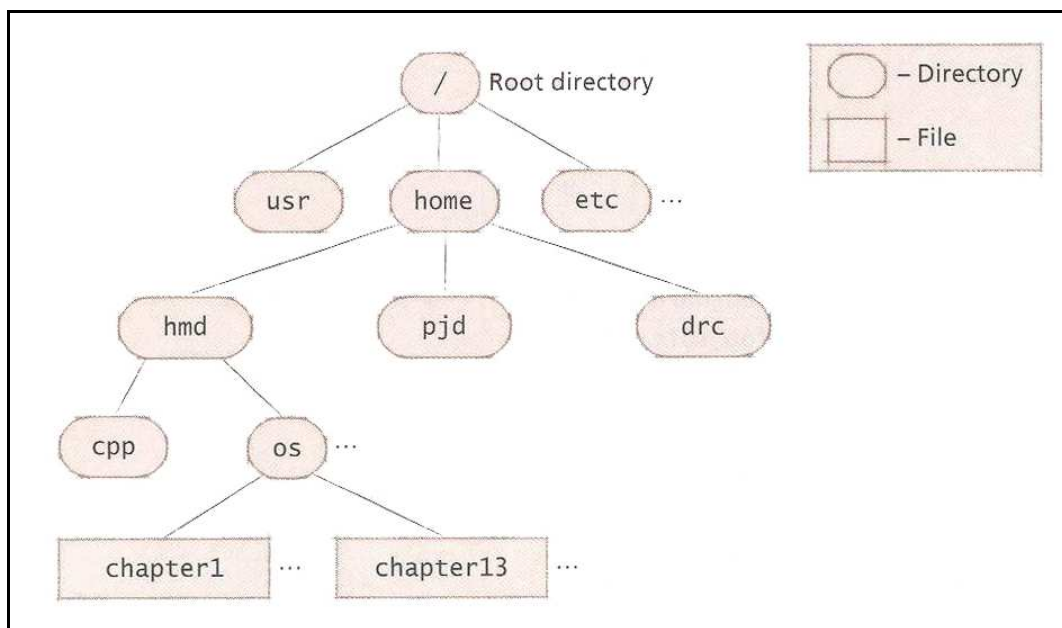
Obr. 2. Dvouúrovňové hierarchické uspořádání souborového systému

3.2.4 Relativní cesta

Mnoho souborových systémů podporuje prvek jako je **pracovní adresář**. Ten slouží především k zjednodušení orientace v názvech cest. Pracovní adresář (reprezentován u souborových systémů Windows a UNIX položkou adresáře specifikovanou pomocí ".") umožní uživatelům specifikovat názvy cest, který nezačínají u kořenového adresáře. Například předpokládejme, že aktuální pracovní adresář je nastaven na */home/hmd/* v souborovém systému na Obr. 3. **Relativní cesta** k */home/hmd/os/chapter13* by byla *os/chapter13*. Tato vlastnost zmenší délku cesty, když se přistupuje k souborům. Když se souborový systém setká s relativní cestou, tak se vytvoří **absolutní cesta** (tzn.: cesta začínající u kořenu). Souborový systém pak překročí strukturu adresáře k uložení požadovaného souboru.

Souborový systém běžně udržuje odkaz na pracovní adresář - **rodičovský adresář (parent directory)** neboli adresář o úroveň výše v hierarchii souborového systému. Na

Obr. 3., je home rodičovský adresář pro hmd, pjd a drc adresáře. V systémech Windows a systémech na bázi UNIX, je odkaz na rodičovský adresář specifikován pomocí "..".



Obr. 3. Příklad hierarchického uspořádání souborového systému

3.2.5 Odkazy

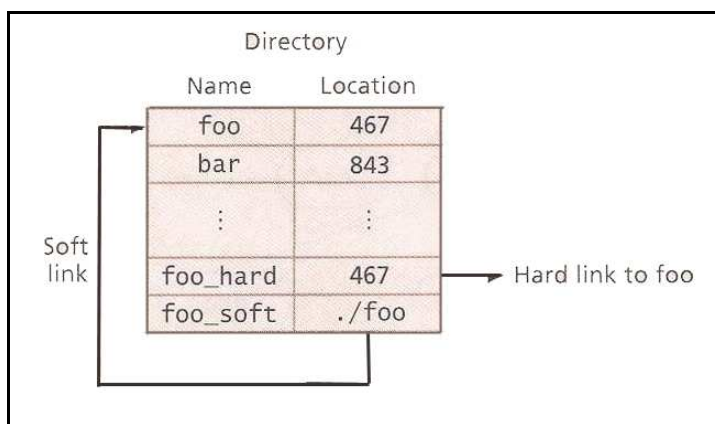
Odkaz (link) je vstup adresáře, který se odkazuje na datový soubor nebo adresář, který je obvykle umístěn v jiném adresáři. Uživatelé často používají odkazy pro zjednodušení orientace v souborovém systému a při sdílení souborů. Předpokládejme, že uživatelský pracovní adresář je `/home/drc/` a uživatel sdílí soubory jinému uživateli umístěné v `/home/hmd/os/`. Tím, že vytvoří odkaz k `/home/hmd/os/` v pracovním uživatelském adresáři, má uživatel přístup k sdíleným souborům jednoduše použitím relativní pathname `os/`.

Symbolický odkaz (soft link) znamená "zkratka" ("shortcut") na systémech Windows, "druhé jméno" ("alias") na systémech Mac-OS a symbolický odkaz v UNIX systémech. Je to vlastně adresářová položka obsahující název s cestou pro další soubor. Souborový systém, lokalizuje cíl symbolického odkazu tím, že překročí adresářovou strukturu použitím specifické cesty.

Tvrдый odkaz (hard link) je vstup adresáře, který specifikuje umístění souboru (běžně číslo bloku) na paměťovém médiu. Souborový systém při zpřístupnění dat pomocí tvrdého odkazu zpřístupní tato data přímo jako fyzický blok.

Obr. 4. objasňuje rozdíl mezi symbolickými a tvrdými odkazy. Soubory *foo* a *bar* jsou umístěny na číslech bloku 467 respektive 843. Vstup adresáře *foo_hard* je tvrdý odkaz, protože specifikuje stejné číslo bloku (467) jako vstup adresáře *foo*. Vstup adresáře *foo_soft* je symbolický odkaz protože specifikuje název s cestou pro *foo* (v tomto případě, */foo*).

Například použitím **reorganizace** disku (přemístění souborů na disku blíže ke středu a naopak) a **defragmentace** disku (spojení souborů, které jsou rozmístěny na disku po částech) může způsobit zlepšení výkonu disku. Během takových operací, se může fyzické umístění souboru měnit, a souborový systém pak vyžaduje aktualizaci umístění souboru v jeho adresářovém záznamu. Protože tvrdý odkaz specifikuje fyzické umístění souboru, tak se může odkazovat na neplatné údaje, v případě změny fyzického umístění odpovídajícího souboru. Když se mění fyzické umístění souboru, může souborový systém používat tyto ukazatele, aby našel tvrdý odkaz, který je potřeba aktualizovat.



Obr. 4. Odkazy v souborovém systému

Protože symbolické odkazy uloží logické umístění souboru v souborovém systému, který se během defragmentace nebo reorganizace nemění, není je potřeba aktualizovat, když jsou data souboru na disku přesunuta. Nicméně, jestliže uživatel přesune soubor do jiného adresáře nebo soubor přejmenuje, symbolické odkazy na tento soubor již nejsou platné. Některé souborové systémy ale vykazují inteligentnější chování. Můžeme dát příklad situace, kdy si uživatel přeje nahradit programový soubor novou verzí programu se stejným jménem. V tomto případě soubor, na který bylo původně odkazováno, může být přesunut, přejmenován nebo zaměněn, ale symbolický odkaz se bude dále odkazovat na platný soubor (novou verzi programu). Většina souborových systémů ale symbolický odkaz neaktualizuje, když je soubor přesunut do jiného adresáře.

Když uživatel odstraní odkaz na soubor musí souborový systém určit zda odstraní také odpovídající soubor. Za tímto účelem udržují souborové systémy jistý počet tvrdých odkazů na soubor. Když počet dosáhne hodnoty nula, souborový systém neobsahuje odkazy na data souboru, tak soubor může být bezpečně odstraněn. Protože symbolické odkazy se neodkazují na fyzické umístění dat souboru, nejsou brány na zřetel při stanovení, zda by soubor měl být odstraněn.

3.3 Metadata

Většina souborových systémů ukládá kromě uživatelských dat a adresářů informace, jako jsou umístění volných bloků paměťového média (slouží k zajištění, že nové data souboru nepřepíší bloky, které jsou již používány) a nebo čas, kdy byl soubor naposledy modifikován (pro účely účetnictví). Tyto informace, se nazývají **metadata**, a slouží k ochraně integrity souborového systému a nelze je modifikovat přímo u uživatelů.

Dříve než je souborový systém používán, je paměťové médium obvykle **formátováno**. Formátování je akce závislá na systému, ale běžně znamená kontrolu paměťového média na nepoužitelné sektory, výmaz některých dat na zařízení a vytvoření kořenového adresáře souborového systému. Mnoho souborových systémů také vytvoří **superblok** k úchově informací a ochraně integrity souborového systému. **Superblok** může obsahovat:

- **Identifikátor souborového systému** - jedinečná identifikace druhu souborového systému
- **Množství bloků** v souborovém systému
- **Umístění volných bloků** paměťového média
- **Umístění kořenového adresáře**
- **Datum a čas** kdy byl souborový systém naposledy modifikován
- **Informace** ukazující zda souborový systém potřebuje být kontrolován (např.: kvůli selhání systému, které předchází zápisu dat z **vyrovnávací paměti (Buffer)** do sekundární paměti).

Jestliže je superblok poškozen nebo odstraněn, nemusí být schopen souborový systém mít přístup k datům souborů. Drobné chyby v datech superbloku (jako např.

umístění volných bloků paměťového média) může způsobit, že souborový systém přepíše existující data souborů. Ke snížení rizika ztráty dat většina souborových systémů distribuuje redundantní kopii superbloku. Souborový systém pak může používat redundantní kopii superbloku k určení, zda je primární superblok poškozen a jestli ano, dojde k jeho nahrazení.

3.3.1 Souborové deskriptory

Když je soubor otevřený, operační systém nejprve najde informaci o souboru tím, že překročí adresářovou strukturu. Vyhne se tak někdy zdlouhavým prodlevám. Systém udržuje tabulku informací v hlavní paměti, která neustále sleduje otevřené soubory. V mnoha systémech operace otevření souboru vrátí **deskriptor souboru** - kladné celé číslo, které registruje do otevřeného souboru. Od tohoto bodu je pak přístup k souboru řízený prostřednictvím deskriptoru souboru.

Tabulka otevřeného souboru obsahuje **kontrolní bloky souboru (file control blocks)**. Tyto obsahují informace, že systém potřebuje spravovat soubor, které se nazývají **atributy souboru**. Atribut souboru je vysoce systémově závislou strukturou. **Kontrolní blok souboru** obsahuje tyto informace:

- **symbolické jméno**
- **umístění** v sekundární paměti
- **organizační strukturu** (např.: sekvenční(sequential), přímý(directly) přístup, atd.)
- **typ zařízení** (např.: pevný disk, CD-ROM)
- **kontrolní data přístupu** (access control data) - Informace, kteří uživatelé mají přístup k souboru a druh přístupu, který je povolen.
- **typ** (datový soubor, objektový program, zdrojový program v jazyce C, atd.)
- **rozestavení** (disposition) - trvalý nebo dočasný
- **datum a čas vytvoření**
- **datum a čas poslední úpravy**
- **počet aktivních přístupů** - např.: počet přečtení

Běžně jsou **kontrolní bloky souboru** udržovány v sekundární paměti a jsou přesunuty do hlavní paměti, když je soubor otevřen. To vede k zrychlení operací se soubory.

3.4 Mountování

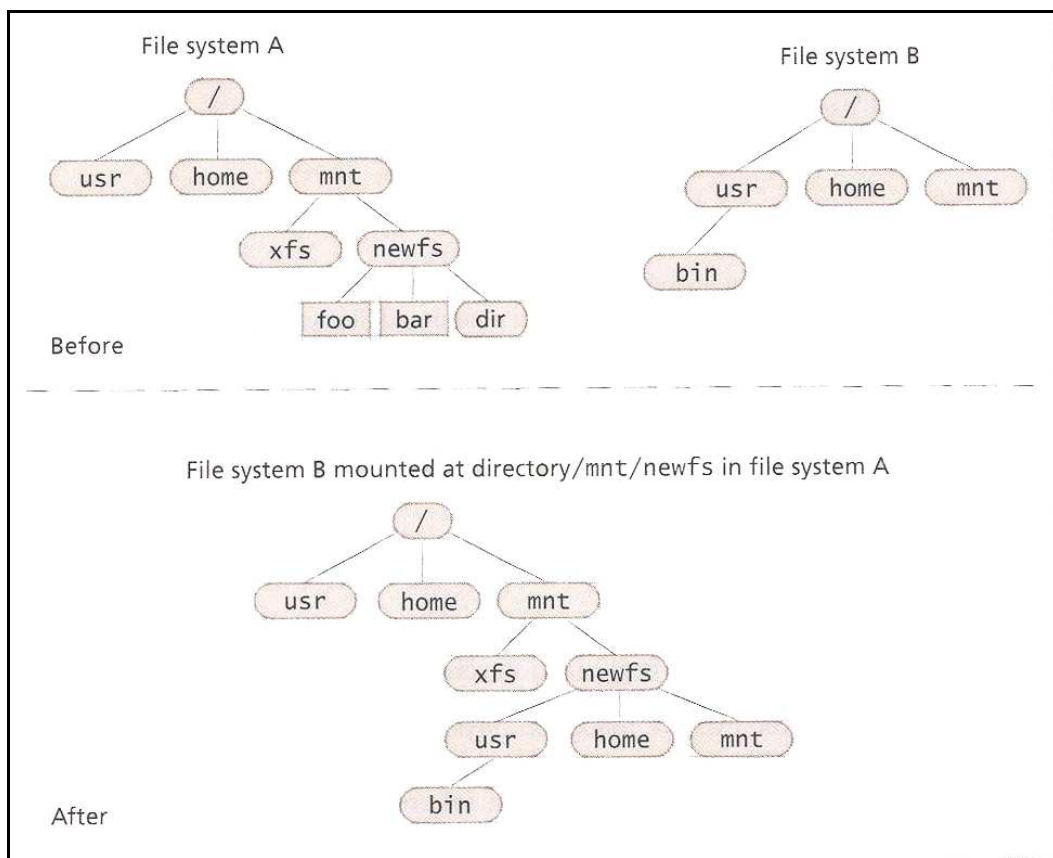
Uživatelé často vyžadují přístup k informaci, která není součástí nativního souborového systému (tzn.: souborový systém, který je trvale připojen na dílčím systému a jehož kořen má vytvořen odkaz na kořenový adresář). Například když mnoho uživatelů ukládá data na druhý pevný disk, DVD nebo jinou pracovní stanici v počítačové síti. Z tohoto důvodu, operační systémy poskytují schopnost **připojení** k různým souborovým systémům. Připojení kombinuje různé souborové systémy do jednoho **namespace** - sada souborů, které mohou být identifikovány jedním souborovým systémem. Sjedení namespace dovolí uživatelům přístup k datům z různých umístění jako by všechny soubory byly umístěny uvnitř nativního souborového systému.

Příkaz mount přiřadí adresář volaný **bodem připojení (mount point)** v nativním souborovém systému ke kořenu připojeného souborového systému. Souborové systémy OS Windows staršího data poskytovaly strukturu **flat mounting**, kde každému připojenému souborovému systému bylo přiřazeno písmeno a byl umístěn do stejné úrovně struktury adresáře. Typickým příkladem je souborový systém obsahující operační systém, který je připojen na C:\ a další souborový systém na D:\.

Souborové systémy na bázi UNIX jako jsou UNIX File System, Fast File System a Second Extended File System - ext2, a verze Microsoft's NTFS 5.0 a vyšší, používají **bod připojení**, které mohou být umístěny kdekoli v souborovém systému. Obsahy adresáře u bodu připojení jsou přechodně skryty, zatímco další souborový systém je u tohoto adresáře připojen.

V systémech na bázi UNIX, jsou některé souborové systémy namountovány u jednoho z adresářů umístěného v */mnt/*. Podívejme se na Obr. 5. Příkaz mount umístí kořen souboru systému B přes adresář */mnt/newfs/* souborového systému A. Po operacích mountování, mohou být v souborovém systému A zpřístupněny soubory souborového systému B od */mnt/newfs/*. Například jeden může zpřístupňovat souborovému systému B adresář */usr/bin/* a souborovému systému A adresář */mnt/newfs/usr/bin/*. Když je

souborový systém namountován, obsahy adresáře u bodu mountování (bodu připojení) jsou nepřístupné uživatelům až do odmountování souborového systému.



Obr. 5. Mountování souborového systému

Souborové systémy řídí namountované adresáře **mount tabulkou**. Ta obsahuje informaci o názvech cest konkrétních bodů mountování, a zařízení, které ukládá každý namountovaný souborový systém. Když se nativní souborový systém setká s bodem mountování, tak použije mount tabulku k určení konkrétního zařízení a druhu souborového systému, který je mountován do toho adresáře. Většina operačních systémů podporuje souborové systémy pro vyměnitelné média jako jsou Univerzální Diskový Formát (UDF) pro DVD a ISO 9660 pro disky CD.

Příkaz `umount` dovolí uživateli odpojit namountované souborové systémy. Toto volání aktualizuje mount tabulku a umožní uživatelům přístup k souborům, které byly skryty připojeným systémem.

Uživatelé mohou vytvořit symbolické odkazy na soubory v namountovaných souborových systémech. Když se souborový systém setká se symbolickým odkazem, tak použije specifický název cesty (pathname) k překročení adresářové struktury. Když se

souborový systém setká s bodem mountování, tak překročí adresář od kořenu souborového systému, který je u bodu mountování. Tvrdý odkaz specifikuje číslo bloku, které je asociováno se zařízením, které ukládá odkaz. Uživatelé nemohou vytvořit tvrdý odkaz mezi dvěma souborovými systémy, z důvodu častého střídání fyzicky různých paměťových zařízení.

3.5 Uspořádání souboru

Uspořádání souboru (File organization) znamená způsob, jakým jsou záznamy souboru uspořádány na sekundární paměti. Zde je několik schémat uspořádání souboru:

- **Sekvenční (Sequential)** - Záznamy bývají umístěny ve fyzické části "dalšího" záznamu, kterému fyzicky následuje předchozí záznam. Toto uspořádání je přirozené pro soubory uložené na magnetickém pásku, jakožto sekvenční médium. Soubory na disku mohou být také uspořádány sekvenčně, ale záznamy v souboru, který je větší než největší volná část diskového prostoru, nemusí být uložen souvisle za sebou. To je závislé na vlastnostech konkrétního souborového systému.
- **Přímý (Direct)** - Záznamy jsou přímo (náhodně) zpřístupňovány od jejich fyzické adresy na paměťovém médiu přímým přístupem (Direct Access Storage Device - DASD). Uživatel aplikace umístí záznamy na DASD v některém požadavku pro vhodnou aplikaci.
- **Sekvenčně indexovaný (Indexed sequential)** - Záznamy na disku bývají uspořádány v logické sekvenci podle klíče obsaženého v každém záznamu. Systém udržuje index obsahující fyzické adresy jistých hlavních záznamů. Sekvenčně indexované záznamy mohou být zpřístupňovány sekvenčně v klíči požadavků, nebo mohou být zpřístupněny přímo, vyhledáním v systému vytvořeném indexu.
- **Rozdělený (Partitioned)** - Toto je základní soubor sekvenčních podsouborů (subfiles). Každý následný podsoubor je volán **členem (member)**. Startovací adresa každého členu je uložena v konkrétním adresáři konkrétního souboru.

4 SOUBOROVÉ SYSTÉMY OPERAČNÍHO SYSTÉMU MICROSOFT WINDOWS

4.1 Souborový systém FAT

Souborové systémy, které jsou v současnosti nejčastěji používány, jsou založeny na alokační tabulce souborů (FAT - File Allocation Table), obsahující údaje o datech, uložených v každém sektoru na disku. Existuje několik variant souborového systému FAT, označovaných jako FAT12, FAT16 a FAT32. Hlavním rozdílem mezi nimi je počet číslic, který lze použít k vyjádření čísla clusteru v alokační tabulce. Jinými slovy řečeno, u souborového systému FAT32 jsou používána 32-bitová čísla, zatímco u systému FAT16 jsou používána čísla 16-bitová. Jednotlivé varianty systému FAT jsou využívány takto:

- **FAT12** - Používá se na všech svazcích o kapacitě menší než 16 MB (např. na disketách).
- **FAT16** - Používá se na svazcích o velikosti 16 MB až 2 GB, Operační systémy Windows NT/2000/XP podporují svazky se souborovým systémem FAT16, mající velikost až 4 GB,
- **FAT32** - Lze jej využít na svazcích o kapacitě 512 MB až 2 TB,

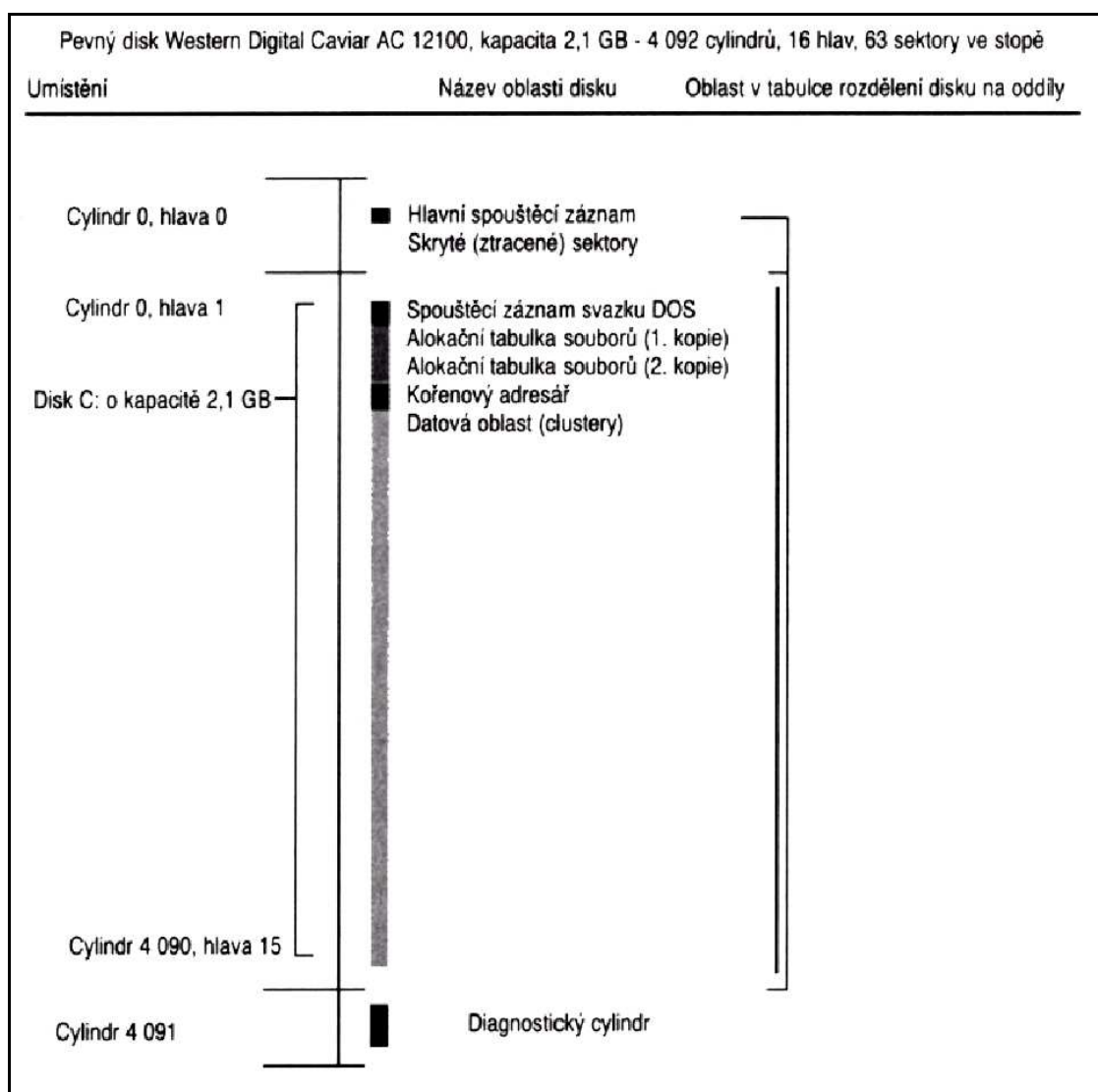
Souborové systémy FAT12 a FAT16 byly součástí původních verzí operačních systémů řady MS-DOS a Windows a z tohoto důvodu je dnes podporuje prakticky každý operační systém, určený pro počítače třídy PC. Naproti tomu souborový systém FAT32 se stal součástí operačních systémů Windows 95, 98, Me a 2000. Operační systém Windows NT/2000/XP, podporuje souborový systém NTFS (NT File System) [8].

Aby bylo možné spravovat jednotlivé soubory na pevném disku a aby všechny aplikace měly stejné rozhraní pro práci se souborovým systémem bez ohledu na použité médium, vytváří operační systém na pevném disku či jiném médiu několik základních struktur, shodných pro všechny operační systémy, podporující daný souborový systém. V případě souborového systému FAT se jedná o tyto struktury a oblasti (uvedené struktury jsou řazeny zhruba v tom pořadí, v jakém se na disku vyskytují):

- Hlavní spouštěcí záznam a spouštěcí záznam rozšířeného oddílu
- Spouštěcí záznam svazku

- Kořenový adresář
- Alokační tabulka souborů
- Clustery (jedná se o jednotlivé alokační jednotky v datové oblasti)
- Diagnostické cylindry pro čtení a zápis

Na pevném disku naleznete všechny uvedené struktury, zatímco na disketě chybí hlavní spouštěcí záznam a diagnostické cylindry. Hlavní spouštěcí záznam je pouze jeden, zatímco každý oddíl či svazek disku má svůj spouštěcí záznam svazku, svůj kořenový adresář, svou alokační tabulku souborů a své clustery. Všechny tyto struktury jsou na disku vytvářeny programem FDISK. Schéma výše popsaných struktur vidíte na Obr. 6.



Obr. 6. Základní struktury souborového systému řady FAT na pevném disku

Každá z těchto oblastí má svůj význam a svou funkci. To také znamená, že poškození kterékoliv z nich může mít poměrně vážné následky pro vaše data. Přitom efekt je obvykle dominový - např. poškodí-li se pouze hlavní spouštěcí záznam, nebude operační systém schopen vůbec přistupovat k datům na celém disku [8].

4.1.1 Hlavní spouštěcí záznam (MBR)

Prvním operačním systémem, podporujícím pevné disky a umožňujícím jejich rozdělení na oddíly, byl DOS 2.0, uvedený na trh roku 1983. Přitom rozdělováním disku na oddíly se rozumí rozdělování disku na několik oblastí, nazývaných svazky. Zde je nutné hned zdůraznit jeden ze základních principů: všechna média, která **mohou** být rozdělena na oddíly, na tyto oddíly **musí** být rozdělena. Jinými slovy řečeno, i když chcete používat na pevném disku jediný oddíl, musíte spustit program pro rozdělování disku na oddíly (FDISK) a tento jediný oddíl musíte vytvořit. Oddíl je mnohdy nazýván logickým svazkem, neboť každému oddílu je obvykle operačním systémem přiřazeno odlišné písmeno jednotky. Informace o každém oddíle je uložena ve spouštěcím záznamu oddílu (či svazku), nacházejícím se na počátku každého svazku. V hlavním spouštěcím záznamu disku se také nachází hlavní tabulka, v níž jsou uvedeny všechny vytvořené oddíly [8].

Hlavní spouštěcí záznam je vždy uložen v prvním sektoru celého disku (cylindr 0, hlava 0, sektor 1) a je tvořen těmito daty:

- **Hlavní tabulkou rozdělení disku** (Master partition table) - Obsahuje seznam všech oddílů, vytvořených na daném disku, a informace o umístění spouštěcích záznamů jednotlivých oddílů. Tato tabulka je velmi malá a může obsahovat maximálně čtyři záznamy. Aby bylo možné vytvořit na disku více než čtyři oddíly, lze každý oddíl rozdělit ještě na několik logických svazků.
- **Hlavní spouštěcí kód** - Malý program, prováděný BIOSem systému. Jeho hlavním účelem je předání řízení systému tomu oddílu, který je v tabulce označen jako aktivní (tj. ten oddíl, ze kterého má být spouštěn operační systém).

4.1.2 Primární a rozšířené oddíly souborového systému FAT

Většina operačních systémů podporuje až 24 oddílů na jediném disku, přičemž každému oddílu je přiřazeno jedinečné písmeno jednotky v rozmezí C: až Z:. Hlavní tabulka rozdělení disku však může obsahovat pouze čtyři záznamy. Řešení tohoto rozporu přinesla

až možnost vytváření oddílů dvou základních typů: **primárních** a **rozšířených**. Primární oddíl by měl být prvním oddílem pevného disku a je mu přiřazeno jedno jediné písmeno disku. Naproti tomu každý rozšířený oddíl může obsahovat několik logických svazků, tj. jedinému rozšířenému oddílu může být přiřazeno několik různých písmen jednotek. Jak primární, tak rozšířené oddíly jsou zaznamenány v hlavní tabulce rozdělení disku. Přitom na pevném disku se souborovým systémem řady FAT můžete pomocí programu FDISK vytvořit pouze jeden rozšířený oddíl, z čehož vyplývá, že na běžném počítači s operačním systémem řady Windows obvykle najdete nejvýše dva oddíly: jeden primární a případně jeden rozšířený.

Logické svazky, vytvořené v rozšířeném oddílu, nejsou zaznamenány v hlavní tabulce rozdělení disku. Díky tomu nemohou tyto svazky být spouštěcími. V jednom jediném rozšířeném oddílu lze vytvořit až 23 logických svazků, pokud předpokládáme, že na témže disku byl již vytvořen primární oddíl. Každý logický svazek pak obsahuje svou vlastní tabulku rozdělení, obsahující informace o konkrétním svazku. Hlavní tabulka rozdělení celého disku obsahuje odkaz pouze na tabulku rozdělení prvního rozšířeného oddílu a ta pak obsahuje informace o umístění tabulky rozdělení druhého oddílu apod.

Protože hlavní spouštěcí záznam obsahuje první program, který je spouštěn z pevného disku po zapnutí počítače, stává se také častým cílem tvůrců počítačových virů. Vir, který poškodí či upraví MBR, znemožní BIOSu vyhledání aktivního oddílu a tedy i zavedení operačního systému. Kromě toho, že MBR je prvním spouštěným programem, lze případný vir po nákaze počítače zachytit jen obtížně - žádný antivirový program totiž nelze spustit. Jedinou možností je pak spuštění počítače z diskety (chráněné proti zápisu) a spuštění antivirového programu z ní.

V prvním sektoru každého vytvořeného oddílu je uložen spouštěcí záznam svazku. Informace o umístění tohoto záznamu jsou uvedeny i v hlavní tabulce rozdělení disku. Pomocí programu FDISK můžete jeden z těchto oddílů označit jako aktivní, tj. jako oddíl, ze kterého má být spouštěn operační systém. Jinými slovy řečeno, je-li nějaký oddíl označen jako aktivní, MBR předá v průběhu spouštění řízení systému spouštěcímu záznamu tohoto oddílu. V hlavní tabulce rozdělení disku mohou být také uvedeny další diskové oddíly, používající jiný souborový systém (např.: NTFS systému Windows NT/2000/XP, HPFS systému OS/2, ext3 systému Linux apod.). Platí však, že z operačních systémů řady Windows 9x/Me nebudete mít k těmto datům přístup. Pokud pak chcete na

takovém počítači spouštět několik různých operačních systémů, musíte obvykle použít program, který vám umožní výběr spouštěného systému (tzv. boot manager). Podrobnější popis formátu hlavního spouštěcího záznamu a hlavní tabulky rozdělení naleznete v tabulce II. V tabulce naleznete údaje pro všechny čtyři záznamy tabulky rozdělení, umístění na disku (offset adresy), kde se dané pole nachází, a jeho délku. Některá pole jsou uložena jako dword, což jsou čtyři byty, které je nutné číst v obráceném pořadí - to je dáno vnitřní strukturou procesorů Intel a kompatibilních [8].

Tab. II. Hlavní spouštěcí záznam a hlavní tabulka rozdělení disku

Hlavní tabulka rozdělení disku, záznam č. 1		
Offset adresy (dec.)	Délka	Popis
1BEh (446)	1 byte	Byte, indikující možnost spouštění (80h = aktivní, jinak 00h)
1BFh (447)	1 byte	Počáteční hlava (či strana) oddílu
1C0h (448)	16 bitů	Počáteční cylindr (10 bitů) a sektor (6 bitů)
1C2h (450)	1 byte	Byte, indikující typ souborového systému
1C3h (451)	1 byte	Konečná hlava (či strana) oddílu
1C4h (452)	16 bitů	Konečný cylindr (10 bitů) a sektor (6 bitů)
1C6h (454)	1 dword	Relativní offset sektoru oddílu
1CAh (458)	1 dword	Celkový počet sektorů v oddílu
Hlavní tabulka rozdělení disku, záznam č. 2		
Offset adresy (dec.)	Délka	Popis
1CEh (462)	1 byte	Byte, indikující možnost spouštění (80h = aktivní, jinak 00h)
1CFh (463)	1 byte	Počáteční hlava (či strana) oddílu
1D0h (464)	16 bitů	Počáteční cylindr (10 bitů) a sektor (6 bitů)
1D2h (466)	1 byte	Byte, indikující typ souborového systému
1D3h (467)	1 byte	Konečná hlava (či strana) oddílu
1D4h (468)	16 bitů	Konečný cylindr (10 bitů) a sektor (6 bitů)
1D6h (470)	1 dword	Relativní offset sektoru oddílu
1DAh (474)	1 dword	Celkový počet sektorů v oddílu
Hlavní tabulka rozdělení disku, záznam č. 3		
Offset adresy (dec.)	Délka	Popis
1DEh (478)	1 byte	Byte, indikující možnost spouštění (80h = aktivní, jinak 00h)
1DFh (479)	1 byte	Počáteční hlava (či strana) oddílu
1E0h (480)	16 bitů	Počáteční cylindr (10 bitů) a sektor (6 bitů)
1E2h (482)	1 byte	Byte, indikující typ souborového systému
1E3h (483)	1 byte	Konečná hlava (či strana) oddílu
1E4h (484)	16 bitů	Konečný cylindr (10 bitů) a sektor (6 bitů)
1E6h (486)	1 dword	Relativní offset sektoru oddílu
1EAh (490)	1 dword	Celkový počet sektorů v oddílu

Hlavní tabulka rozdělení disku, záznam č. 4		
Offset adresy (dec.)	Délka	Popis
1EEh (494)	1 byte	Byte, indikující možnost spouštění (80h = aktivní, jinak 00h)
1EFh (495)	1 byte	Počáteční hlava (či strana) oddílu
1F0h (496)	16 bitů	Počáteční cylindr (10 bitů) a sektor (6 bitů)
1F2h (498)	1 byte	Byte, indikující typ souborového systému
1F3h (499)	1 byte	Konečná hlava (či strana) oddílu
1F4h (500)	16 bitů	Konečný cylindr (10 bitů) a sektor (6 bitů)
1F6h (502)	1 dword	Relativní offset sektoru oddílu
1FAh (506)	1 dword	Celkový počet sektorů v oddílu
Byty, indikující spouštěcí sektor		
Offset adresy (dec.)	Délka	Popis
1FEh (510)	2 byty	Byly, indikující spouštěcí sektor (55AAh)

Data uložená v hlavní tabulce rozdělení disku určují počátek a konec každého oddílu, jeho velikost, zda lze z něho spouštět operační systém a jaký typ souborového systému je na oddílu použit [8].

4.1.3 Spouštěcí záznam svazku (VBR)

Spouštěcí záznam svazku se nachází v prvním sektoru kterékoliv části disku, označené jako oddíl či svazek. V případě disket či médií Zip se jedná o první sektor celého média, neboť tato média jsou operačním systémem rozpoznávána jako svazky bez nutnosti jejich dělení na oddíly. Spouštěcí záznam svazku se podobně jako hlavní spouštěcí záznam skládá ze dvou částí:

- **Bloku parametrů disku** - Někdy se tato část označuje též jako blok parametrů média. Obsahuje specifické parametry, týkající se daného svazku: např. velikost, počet sektorů, velikost clusteru a název svazku.
- **Spouštěcího kódu svazku** - Program, který zahájí proces spouštění operačního systému. V případě systémů řady Windows 9x/Me se jedná o soubor IO.SYS.

Spouštěcí záznam svazku aktivního oddílu pevného disku je načítán hlavním spouštěcím záznamem disku. V případě disket je spouštěcí záznam svazku načítán ROM BIOSem. Tím okamžikem spouštěcí záznam svazku přebírá řízení systému, provede několik testů a následně se pokusí o načtení prvního souboru operačního systému (tedy IO.SYS v případě Windows 9x/Me). Platí, že podobně jako hlavní spouštěcí záznam i spouštěcí záznam svazku se nachází mimo datovou oblast svazku [8].

4.1.4 Kořenový adresář

Adresářem je poměrně jednoduchá databáze, obsahující informace o souborech, uložených v oddíle se souborovým systémem FAT. Každý záznam této databáze má délku 32 bytů, přičemž jednotlivé záznamy ani pole nejsou oddělovány žádnými speciálními znaky. V adresáři jsou uloženy prakticky všechny informace, které jsou operačnímu systému o daném souboru známy:

- **Název souboru a jeho extenze** - Název souboru, tvořený 8 znaky, a 3 znaky určující extenzi názvu souboru. Název a extenze se oddělují tečkou, avšak v adresáři tato tečka uložena není.
- **Byte s atributy souboru** - Tento byte obsahuje návěští, signalizující standardní atributy souboru, používané systémem DOS.
- **Datum a čas poslední změny** - Datum a čas vytvoření či poslední změny souboru,
- **Velikost souboru** - Velikost souboru v bytech.
- **Odkaz na počáteční cluster** - Číslo clusteru oddílu, ve kterém daný soubor začíná.

Podrobněji je formát adresáře popsán v tabulce III. Existují ovšem i informace, týkající se konkrétního souboru, které v adresáři nenajdete. K těmto informacím patří zejména odkazy na další clustery, využívané daným souborem, a informace o spojitosti souboru. Tyto údaje jsou uloženy v alokační tabulce souborů FAT [8].

Tab. III. Struktura adresáře souborového systému FAT

Offset adresy (dec.)	Délka	Popis
00h (0)	8 bytů	Název souboru
08h (8)	3 byty	Extenze názvu souboru
0Bh (11)	1 byte	Atributy souboru
0Ch (12)	10 bytů	Rezervováno (00h)
16h (22)	1 word	Čas vytvoření či poslední úpravy
18h (24)	1 word	Datum vytvoření či poslední úpravy
1Ah (26)	1 word	Počáteční cluster
1Ch (28)	1 dword	Velikost v bytech

V systémech DOS a Windows 9x existují dva základní typy adresářů - **kořenový adresář** a **podadresář**. Každý svazek může mít pouze jeden kořenový adresář. Přitom kořenový adresář je na každém svazku uložen na pevně daném místě, následujícím hned za oběma kopiemi alokační tabulky souborů. Velikost kořenového adresáře závisí mimo jiné

na typu a kapacitě disku, avšak pro daný disk je velikost kořenového adresáře pevná. Kořenový adresář se vytváří při chodu programu FORMAT a po jeho vytvoření nelze jeho velikost změnit. Omezení počtu záznamů pro kořenové adresáře jsou uvedena v tabulce IV. Podadresáře jsou uloženy jako soubory v datové oblasti disku, což znamená, že jejich velikost se může měnit dynamicky. Jednou z výhod souborového systému FAT32 je to, že kořenový adresář může být umístěn kdekoliv na disku a může mít neomezený počet záznamů.

Tab. IV. Počty záznamů v kořenovém adresáři

Typ disku	Maximální počet záznamů v kořenovém adresáři
Pevný disk	512
Disketa o kapacitě 1,44 MB	224
Disketa o kapacitě 2,88MB	448
Médium Iomega Jaz a Zip	512
Disketa LS-120	512

4.1.5 Alokační tabulky souborů (FAT)

Tato tabulka popisuje přiřazení každého clusteru v oddílu, tj. datová část oddílu má pro každý cluster jeden záznam v této tabulce. Sektory, nacházející se v té oblasti oddílu, která není určena pro data (spouštěcí záznamy, tabulky FAT a kořenový adresář) nejsou v této tabulce popsány.

Přitom základní jednotkou pro ukládání dat je sektor. Aby však FAT nemusela popisovat každý sektor oddílu, alokuje se prostor na disku po určitých skupinách sektorů, nazývaných clustery či alokační jednotky. Velikost clusteru (tj. počet sektorů, tvořících cluster) je určen programem FDISK v průběhu vytváření oddílů; základním hlediskem přitom je velikost oddílu. To také znamená, že nejmenším prostorem, který může být obsazen daným souborem je jeden cluster. Jinými slovy řečeno, bude-li soubor o jediný byte větší než jeden cluster, budou k jeho uložení použity clustery dva.

Tabulku FAT si lze představit jako velkou tabulku, jejíž každá buňka odpovídá jednou clusteru. Hodnota, uložená v buňce, pak určuje, zda je daný cluster použit nějakým souborem a pokud ano, pak také určuje, ve kterém dalším clusteru se nachází pokračování

tohoto souboru. Údaj o počátečním clusteru daného souboru je v adresáři, v popisu počátečního clusteru, uloženém ve FAT, pak je i údaj o následujícím clusteru atd. Takto lze vysledovat všechny clustery, použité daným souborem - budou tvořit řetězec, nazývaný řetězec FAT. Čísla, uložená ve FAT, jsou buď 12-bitová nebo 16-bitová (proto FAT12 nebo FAT16); 16-bitová čísla se v programu pro přímou editaci disku (např. DISKEDIT) sledují poměrně snadno, neboť každé číslo zabírá 2 byty. Poněkud horší je to při práci s 12-bitovými čísly - ta zabírají pouze 1,5 bytu. Většina editorů však provádí automatický převod hexadecimálních čísel na čísla decimální, se kterými se většině uživatelů pracuje podstatně lépe [8].

4.1.6 Clustery (alokační jednotky)

Cluster (od operačního systému DOS 4.0 alokační jednotka) je nejmenší jednotkou diskového prostoru, kterou operační systém může využít pro zápis dat nějakého souboru. Cluster je vždy celým násobkem sektorů, majících velikost obvykle 512 bytů; samotný násobek pak je mocninou čísla 2. Ačkoliv cluster může být roven jednomu sektoru, většinou jich obsahuje více: snižuje se tím mimo jiné režie samotného souborového systému. Nevýhodou je však ztráta určitého prostoru na disku.

Standardní velikost clusteru pro diskety naleznete v tabulce V. Pro pevné disky a jejich oddíly o různé velikosti naleznete tyto údaje v tabulce VI.

Protože velikost souboru je málokdy celým násobkem velikosti clusteru, je poslední cluster, použitý pro uložení konce souboru, jen zřídka zaplněn úplně. Z toho ale vyplývá určitá ztráta diskového prostoru. Přitom platí, že diskový oddíl s clusterem o větší velikosti má vždy větší ztrátu než oddíl s clusterem o menší velikosti. Navíc u FAT16 platí, že pole bloku parametru disku, obsahující údaj o velikosti clusteru, má velikost pouze dva byty. Lze tedy do něj uložit maximální hodnotu 65 535, odpovídající clusteru o velikosti 32 kB. Uvážíme-li takto veliký cluster, disk se zhruba 5 000 soubory a průměrnou zaplněnost posledního clusteru každého souboru 50 %, pak výsledná ztráta diskového prostoru na tomto disku je více než 78 MB ($5\,000 \times 0,5 \times 32$).

Velikost clusteru v kombinaci s dalšími parametry FAT také určuje maximální možnou velikost diskového oddílu, využívajícího danou verzi souborového systému. Protože FAT16 používá pro vyjádření čísla clusteru v alokační tabulce pouze 16-bitová čísla, znamená to, že na daném oddíle může být nejvíce 65 536 clusterů (2^{16}). Současně

maximální velikost jednoho clusteru je 32 kB. Z toho vyplývá, že maximální možná velikost diskového oddílu se souborovým systémem FAT16 je 2 GB (2 096 832 kB) [8].

Tab. V. Standardní velikost clusteru pro diskety

Typ diskety	Velikost clusteru	Hustota záznamu
3.5", 720 kB	2 sektory (1 024 bytů)	Nízká
3.5", 1.44 MB	1 sektor (512 bytů)	Vysoká
3.5", 2.88 MB	2 sektory (1 024 bytů)	Velmi vysoká

Tab. VI. Standardní velikost clusteru pro různé velikosti pevných disků a jejich oddílů

Velikost svazku (oddílu)	Velikost clusteru	Typ FAT
0 MB až 16 MB	8 sektorů (4 096 bytů)	FAT12
16 MB až 128 MB	4 sektory (2 048 bytů)	FAT16
128 MB až 256 MB	8 sektorů (4 096 bytů)	FAT16
256 MB až 512 MB	16 sektorů (8 192 bytů)	FAT16
512 MB až 1 024 MB	32 sektorů (16 384 bytů)	FAT16
1 024 MB až 2 048 MB	64 sektorů (32 768 bytů)	FAT16
32 MB až 260 MB	1 sektor (512 bytů)	FAT32
260 MB až 8 GB	8 sektorů (4 096 bytů)	FAT32
8 GB až 16 GB	16 sektorů (8 192 bytů)	FAT32
16 GB až 32 GB	32 sektorů (16 384 bytů)	FAT32
32 GB až 2 TB	64 sektorů (32 768 bytů)	FAT32

4.1.7 Diagnostické cylindry pro čtení a zápis

Program FDISK vždy vyhrazuje poslední cylindr disku pro diagnostické a testovací účely. Z tohoto důvodu FDISK vždy uvádí menší počet cylindrů než výrobce disku. Operační systémy nejsou schopny tuto oblast disku využít, neboť leží mimo rozdělenou oblast disku.

Tato vyhrazená oblast může být využita pouze softwarem, dodaným výrobcem disku, a sloužícím pro testování čtení a zápisu disku bez rizika poškození dat.

Na discích s rozhraním IDE či SCSI si disk a řadič mohou vyhradit další oblasti disku, nacházející se za jeho logickým koncem. Do těchto oblastí jsou pak ukládány tabulky s informacemi o vadných sektorech apod. Tím vznikají další rozdíly mezi údaji programu FDISK a výrobce disku [8].

4.2 Souborový systém NTFS

Tato kapitola uvádí podrobné informace o vnitřní struktuře a funkci souborového systému Microsoft Windows NT (dále jen NTFS). Nejdříve shrneme cíle vývojářů a uvedeme hlavní charakteristiky NTFS, pak se budeme věnovat struktuře dat na disku a to jak logické, kde si například popíšeme službu logového souboru, tak i fyzické, která pojedná o svazcích, clusterech, zrcadlených sadách a ostatních vlastnostech NTFS.

4.2.1 Vývojové cíle a charakteristiky NTFS

Už v roce 1988 podporoval Microsoft dva souborové systémy - FAT v MS-DOS a Microsoft Windows a HPFS (high-performance file system) v OS/2. Oba tyto systémy mají omezení, která znamenají menší spolehlivost, než by měl mít souborový systém Windows NT a nejsou schopné zvládnout rozsáhlé HW konfigurace systému, pro které jsou určený Windows NT. Po důkladném zvážení se vývojový tým rozhodl vyvinout nový souborový systém - NTFS. I když vývoj NTFS byl zcela nový, byl přesto ovlivněn systémy FAT, HPFS a některými požadavky norem POSIX [2].

4.2.2 Hlavní přednosti souborového systému NTFS

Souborový systém NTFS je oproti souborovým systémům FAT a HPFS vhodný pro nové aplikace vyžadující zotavení, násobné proudy dat, jména souborů ve znakové sadě Unicode, bezpečnost, schopnost všeobecné indexace, dynamické přemapování vadných clusterů, zdvojení nebo redundance (nadbytečnost) dat, odolnost vůči selhání a podporu stále větších médií. O některých z těchto vlastností bude dále pojednáno podrobněji.

Zotavení

Pokud se soustředíme na diskové V/V operace, uživatelé osobních počítačů mají největší zájem o rychlost. Obvykle chtějí jen rychle udělat práci. Windows NT však posunuly využití osobních počítačů do oblastí, kde se spolehlivost uložených dat stala důležitější než rychlost, s jakou lze k datům na disku přistupovat. Jinými slovy, když systém selže, disk se poškodí nebo je nepřístupný, pokud je rychlost V/V operací je nezajímavá.

Požadavky na spolehlivé uložení dat a spolehlivý přístup k nim řeší souborový systém NTFS na základě transakčního modelu zpracování. Transakční zpracování je technika zpracování modifikací databází, při které selhání systému neovlivní správnost

nebo integritu databáze. Základním principem transakčního zpracování je, že databázové operace, nazývané transakce, mají logickou hodnotu vše nebo nic (pravda nebo nepravda). Transakce je definována jako operace, která mění data v souborovém systému nebo strukturu adresářů na disku. Jednotlivé změny na disku, které vyvolají transakci, musí být provedeny atomicky (jako nedělitelné operace). To znamená, že když začne zpracování jedné transakce, všechny změny na disku musí být dokončeny. Když selhání systému přeruší transakci, část, která již byla vykonána musí být zrušena (vrácena zpět). Operace návratu zpět vrátí databázi do dřívějšího známého a konzistentního stavu tak, jako by k transakci nikdy nedošlo.

NTFS využívá transakční model zpracování ke splnění požadavku zotavení souborového systému. Když program iniciuje V/V operaci, která změní strukturu NTFS (změní adresářovou strukturu, rozšíří soubor, vyhradí prostor pro nový soubor atd.), NTFS provede tuto operaci jako atomickou transakci. To zajistí, že je buď dokončena, nebo dojde-li k selhání systému během zpracování transakce je vrácena zpět. Kromě toho NTFS používá zdvojené ukládání životně důležitých informací souborového systému. Pokud je jeden prostor uložení těchto informací porušen, NTFS má stále k dispozici všechna důležitá data. Toto zdvojení v souborovém systému kontrastuje se strukturou uložení dat na svazku u souborových systémů FAT a HPFS, které ukládají kritická data souborového systému na jediné místo. Pokud se objeví chyba čtení v jediném sektoru těchto dat, data na celém svazku jsou ztracena.

Bezpečnost

Bezpečnost dat je rozhodující pro zákazníky, kteří zpracovávají soukromé nebo citlivé informace, jako jsou banky, nemocnice a agentury spojené s národní bezpečností. Tito zákazníci potřebují záruky, že jejich data budou bezpečná proti neautorizovanému přístupu. O bezpečnosti bylo již pojednáno v kapitole 3. v obecném pojednání o souborových systémech.

Bezpečnost v NTFS je odvozena přímo z objektového modelu Windows NT. Otevřený soubor je považován za objekt. Informace o bezpečnosti (deskriptor zabezpečení) je uložena na disku jako součást souboru. Pokud nějaký proces přistupuje k jakémukoli objektu, včetně souborového objektu, bezpečnostní systém Windows NT zkontroluje, zda proces má právo to udělat. Deskriptor zabezpečení kombinovaný s autorizovaným přihlášením uživatele do systému (zadáním hesla) zajišťuje, že k souboru

může přistoupit jen proces, který k tomu má příslušné právo od správce systému nebo vlastníka souboru [2].

Redundance dat a odolnost vůči haváriím

Mimo vlastnosti zotavení souborového systému někteří zákazníci vyžadují, aby jejich data nebyla ohrožena výpadkem napájení nebo selháním systému. Schopnost zotavení NTFS zajišťuje, že souborový systém na médiu zůstane přístupný, ale nedává žádné záruky na úplné zotavení uživatelských souborů. Aplikacím, které nemohou riskovat ztrátu dat uložených v souborech, poskytuje redundance dat další úroveň ochrany.

Vrstvový model ovladačů Windows NT poskytuje podporu diskům odolným vůči haváriím. Při zápisu dat na disk NTFS komunikuje s ovladačem disku odolným vůči havárii, který zase komunikuje s řadičem pevného disku. Tato komunikace umožňuje systému Windows NT instalací dodatečného ovladače vytvořit disková zařízení odolná vůči haváriím.

Velkokapacitní disky a velké soubory

Technické a vědecké aplikace často ukládají a zpracovávají extrémně velká množství informací. Disky s kapacitou 8 GB a disková pole s kapacitou 100 až 500 GB již nejsou výjimkou. NTFS podporuje disky s velkou kapacitou a velké soubory efektivněji než souborové systémy FAT nebo HPFS.

NTFS alokuje clustery a používá k jejich číslování 64 bitů, to znamená 2^{64} možných clusterů (16 000 000 000 000 000 000), každý velikosti až 64 KB. Každý soubor může mít velikost až 2^{64} a to bude nějakou dobu vyhovovat požadavkům na ukládání dat. Tak jako u FAT je velikost clusteru v NTFS volitelná, ale nemusí růst úměrně s velikostí disku. NTFS používá clustery velikosti 512 bytů na malých discích a maximální velikost clusteru je 64 KB na velkokapacitních discích. Ačkoli NTFS používá 64 bitů (8 bytů) pro adresu vyhrazeného (alokovaného) prostoru, kóduje adresu tak, že zabere jen 3 až 5 bytů [2].

Šifrování

Šifrování je užitečné tehdy, potřebujeme-li, aby každý soubor složky byl zašifrován. Pak není možné, aby ho někdo přečetl jinou instalací NT. V kombinaci se standardní a velice bezpečným heslem do samotného systému tato možnost nabízí bezpečnost vámi vybraných důležitých dat pro většinu aplikací.

Kvóty

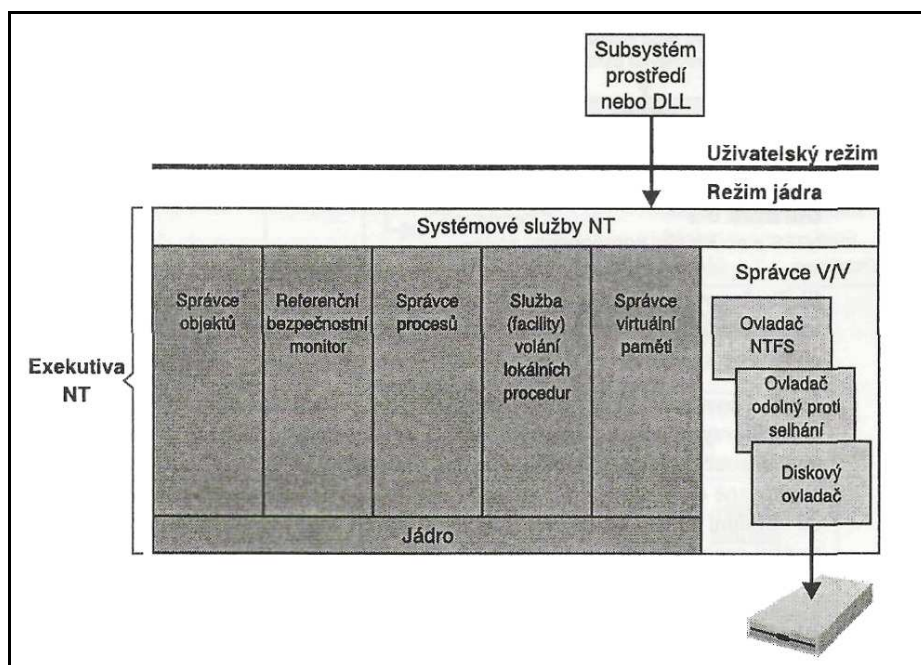
Další vlastností, kterou přinesla až NTFS 5.0, bylo kvótování. To znamená, že administrátor může každému uživateli nastavit, kolik může do dané složky uložit dat. Výhodné je to zvláště pro servery nebo stanice s přístupem více lidí. Tato vlastnost se používá stejným způsobem jako u serveru Novell Netware [9].

Kvóta je v MFT realizována souborem \$QUOTA a to pod záznamem 9.

4.2.3 Logická struktura NTFS

Souborový systém NTFS, ale i jiné souborové systémy jsou vlastně ovladače zařízení, které běží v režimu jádra systému (kernel mód). Na Obr. 7. je znázorněno, jak subsystém prostředí Windows NT volá systémovou službu, která najde správný zavedený ovladač zařízení a vyvolá jej.

Vrstvové ovladače si předávají V/V požadavek jeden druhému voláním správce V/V exekutivy Windows NT. Zprostředkující funkce správce V/V umožňuje nezávislost jednotlivých ovladačů, takže mohou být zavedeny nebo odinstalovány, aniž by ovlivnily ostatní. Ovladač NTFS navíc spolupracuje se třemi dalšími exekutivami NT, které jsou v levé části Obr. 7. a jsou pevně svázány se souborovým systémem [2].

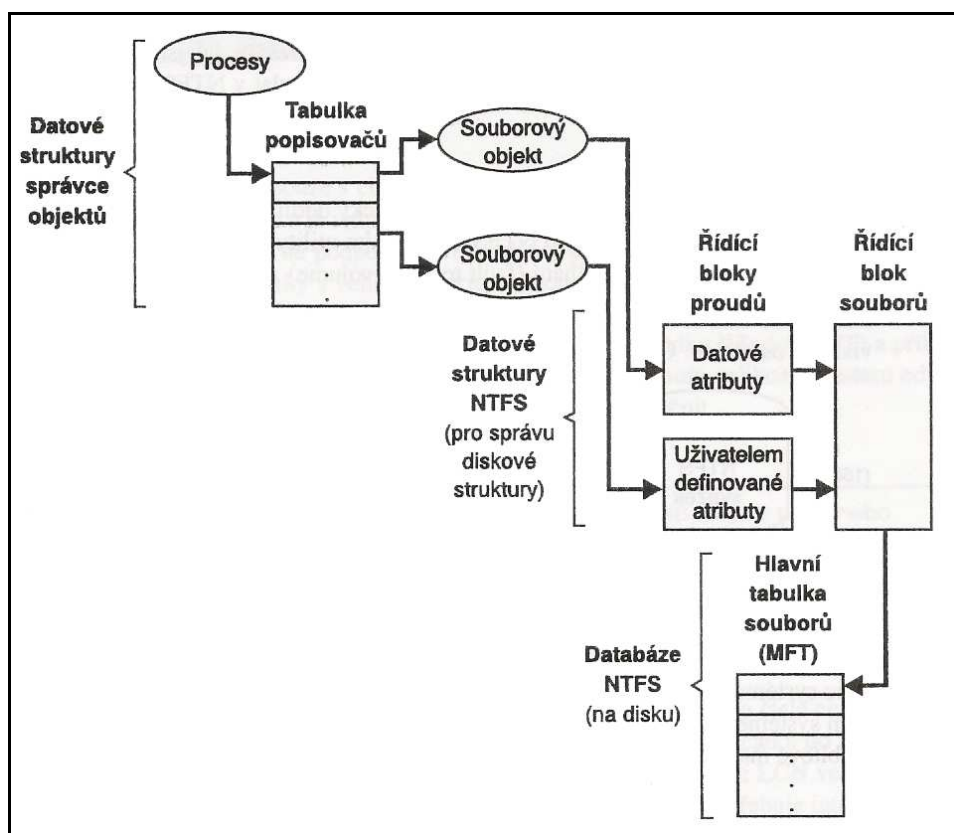


Obr. 7. Jednotlivé části V/V systému Windows NT

Služba logového souboru (log file service LFS) je částí NTFS poskytující služby pro správu záznamů o zápisech na disk. Záznamy jsou využity při zotavení NTFS disku v případě havárie systému.

Správce vyrovnávací paměti (cache manager) je částí exekutivy Windows NT poskytující služby vyrovnávací paměti pro NTFS a jiné ovladače souborových systémů včetně síťových souborových systémů (servery a redirektory).

Datové struktury spojující návěští souboru se strukturou souborového systému na disku jsou na Obr. 8. Je zde zobrazen souborový objekt, který reprezentuje jedno volání systémové služby otevřených souborů, ukazuje na řídicí blok proudu (stream control block SCB) s atributy souboru, které zkouší volající číst nebo zapisovat. Podívejme se znovu na Obr. 8. Proces zde otevřel datové atributy i uživatelem definované atributy souboru. SCB reprezentují jednotlivé atributy souboru a obsahují informace jak je najít v souboru. Všechny SCB jednoho souboru ukazují na společnou datovou strukturu řídicí blok souboru (file control block FCB). FCB obsahuje ukazatel na záznam v hlavní tabulce souborů (MFT) na disku [2].



Obr. 8. Datové struktury NTFS

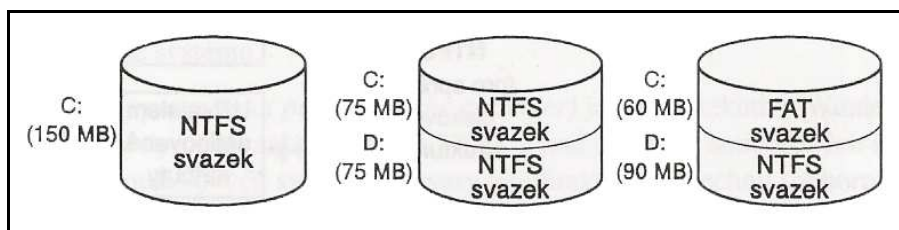
4.2.4 Fyzická struktura NTFS

Tato část popisuje diskovou strukturu NTFS svazků. Uvádí, jak je prostor členěn a organizován do clusterů, jak jsou soubory umísťovány do adresářů, jak jsou data a atributy souborů ukládány na disk a konečně jak pracuje komprese dat v NTFS.

Svazky

Struktura NTFS začíná u svazků. Svazek je logický oddíl na disku vytvořený při formátování disku nebo části disku pro NTFS. S použitím utility **Disk Administrator** lze vytvořit také svazek odolný proti selhání (fault tolerant volume) zahrnující několik disků.

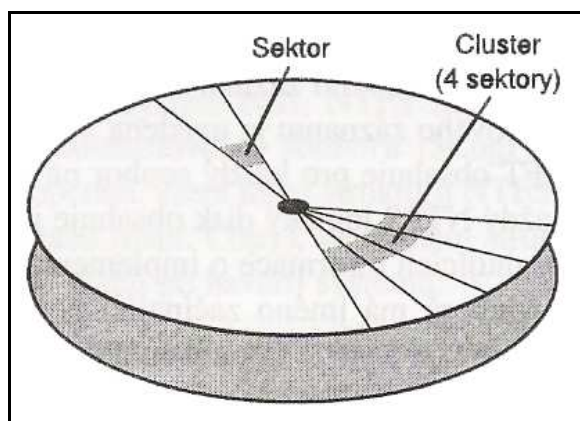
Disk může zahrnovat jeden nebo několik svazků. NTFS pracuje s každým svazkem nezávisle na ostatních. Na Obr. 9. jsou znázorněny tři možnosti konfigurace pevného disku o velikosti 150 MB [2].



Obr. 9. Možné konfigurace disku

Clustery

NTFS používá clustery jako základní jednotky vyhrazení (alokace) diskového prostoru (obdobně jako souborový systém FAT). Velikost clusteru na svazku (cluster factor) určuje uživatel při formátování svazku příkazem Format nebo v utilitě Disk Administrator. Velikost clusteru se mění s velikostí logického disku, ale vždy je 2^n násobkem velikosti fyzického sektoru (1 sektor, 2 sektory, 4 sektory, 8 sektorů atd. viz Obr. 10.).

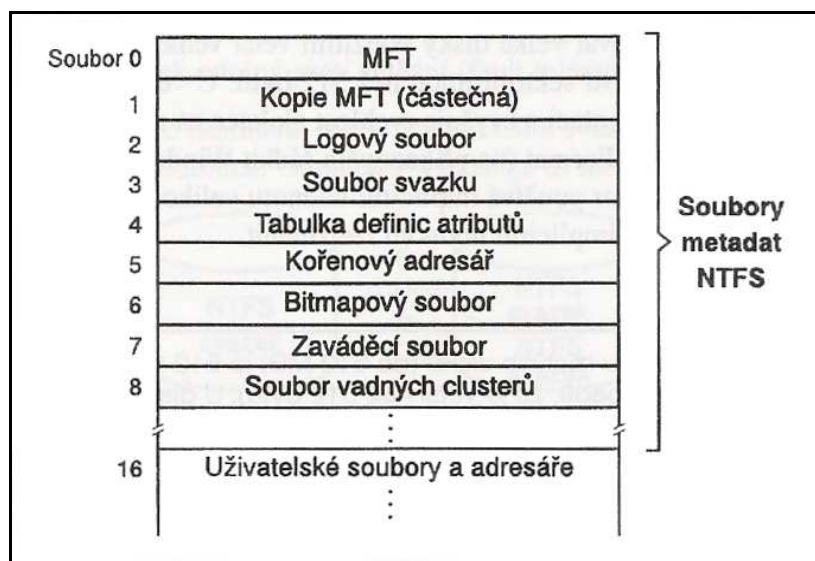


Obr. 10. Sektory a clustery na disku

Velikost clusteru je vyjadřována jako počet bytů v clusteru (512 bytů, 1 KB, 2 KB). Implicitní hodnota velikosti clusteru malých disků (do 512 MB) je 512 bytů (nebo hardwarová velikost sektoru v případě, že je větší než 512 bytů). U disků do 1 GB je implicitní velikost clusteru 1 KB, u disků 1 GB - 2 GB je 2 KB a u disků nad 2 GB je 4 KB. Tyto hodnoty jsou kompromisem mezi fragmentací disku, ke které může dojít u disků s příliš malou velikostí clusteru, a nevyužitým diskovým prostorem (vnitřní fragmentace) u příliš velkých clusterů.

Hlavní tabulka souborů (Master file table MFT)

V NTFS jsou všechna data uložená na disku obsahem nějakého souboru, včetně datových struktur používaných k nalezení a přečtení souboru, zavaděče a bitových map zobrazujících stav alokace na celém disku (metadata NTFS). Uložení všech dat v souborech umožňuje souborovému systému snadno alokovat a udržovat data, každý samostatný soubor lze chránit popisovačem zabezpečení. Pokud se stane část disku nepoužitelnou, NTFS může metadata přemístit a zabránit tak nepřístupnosti disku [2].



Obr. 11. Souborový záznam NTFS v MFT

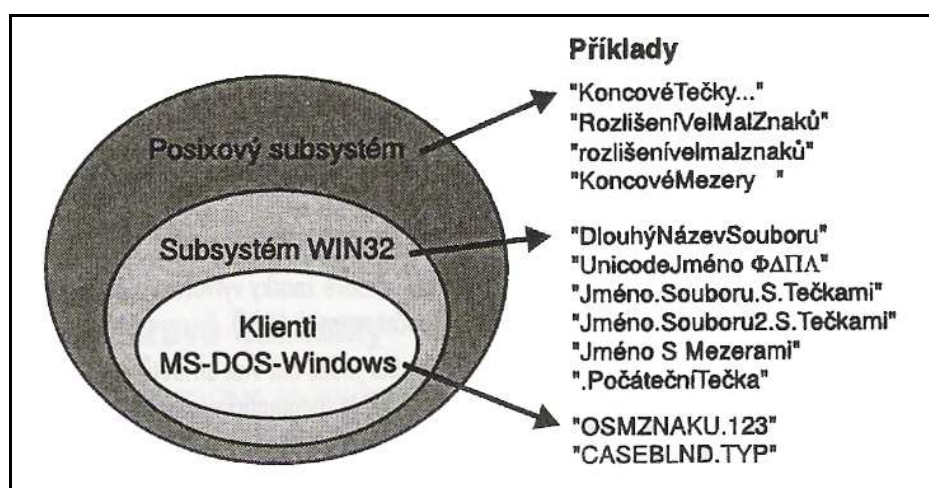
Hlavní tabulka souborů (MFT) je srdcem struktury NTFS na disku. MFT je implementována jako pole souborových záznamů. Velikost souborového záznamu je pevná (1 KB), nezávislá na velikosti clusteru. Logicky MFT obsahuje pro každý soubor na disku jeden řádek, včetně jednoho řádku pro sebe. Každý NTFS logický disk obsahuje ještě kromě MFT několik dalších souborů metadat obsahujících informace o implementaci struktury souborového systému. Každý z těchto souborů má jméno začínající dolarem (\$) a

jsou proto skryté. Například jméno souboru MFT je \$MFT. Ostatní soubory na NTFS disku jsou běžné uživatelské soubory a adresáře, jak je uvedeno na Obr. 11.

Obvykle každý záznam MFT odpovídá jednomu souboru. Pokud má soubor velký počet atributů nebo je velmi fragmentovaný, může nastat situace, že potřebuje více než jeden záznam. V těchto případech je první záznam obsahující umístění ostatních nazýván základní souborový záznam (base file record).

Jména souborů

Oba souborové systémy, NTFS i FAT, umožňují jména souborů s délkou 255 znaků (včetně cesty). Jména souborů mohou obsahovat znaky sady Unicode, několik teček a mezery. Souborový systém FAT v MS-DOS je však omezen u jmen souborů na 8 znaků (nikoli Unicode) následovaných tečkou a třemi znaky extenze. Obr. 12. znázorňuje vzájemnou působnost různých typů jmen souborů podporovaných Windows NT [2].



Obr. 12. Typy jmen souborů ve Windows NT

Indexování souborů

V NTFS si lze adresář představit jako index jmen souborů, jinými slovy souhrn jmen souborů (včetně jejich souborových odkazů) organizovaný tak, aby k nim byl rychlý přístup. Při vytváření adresáře NTFS indexuje atributy jmen souborů v adresáři.

Uložení jmen souborů pomocí indexování má několik výhod. Prohledávání adresářů je rychlé, protože jména souborů jsou uložena uspořádaně. Když software vyšší úrovně vypisuje soubory v adresáři, NTFS poskytuje již seřazená jména. Proto se při rozrůstání adresářů doba prohledávání výrazně neprodlužuje.

Datová komprese

NTFS podporuje kompresi souborů, adresářů nebo disku (v současné době se komprimují pouze uživatelská data, nikoli metadata souborového systému). Uvědomme si, že nastavení komprese u souboru způsobí okamžitou změnu komprese nebo dekomprese, nastavení u adresáře se týká pouze nově vytvořených souborů a podadresářů.

Kompresa "řídkých" souborů (s volným místem)

Řídké (sparse) soubory jsou často velké soubory obsahující ve srovnání s velikostí pouze malé množství nenulových dat. Příkladem řídkého souboru je nulová matice uložená na disku. Pokud uživatel vybere na NTFS disku soubor pro kompresi, jednou kompresní technikou NTFS je odstranění dlouhých nulových řetězců ze souboru. Pokud je soubor řídký, zmenší se natolik, že zaujímá jen zlomek prostoru na disku. Následně zapíše do souboru, že NTFS alokuje prostor pouze pro nenulové extensity.

Kompresa "hustých" (nonsparse) dat

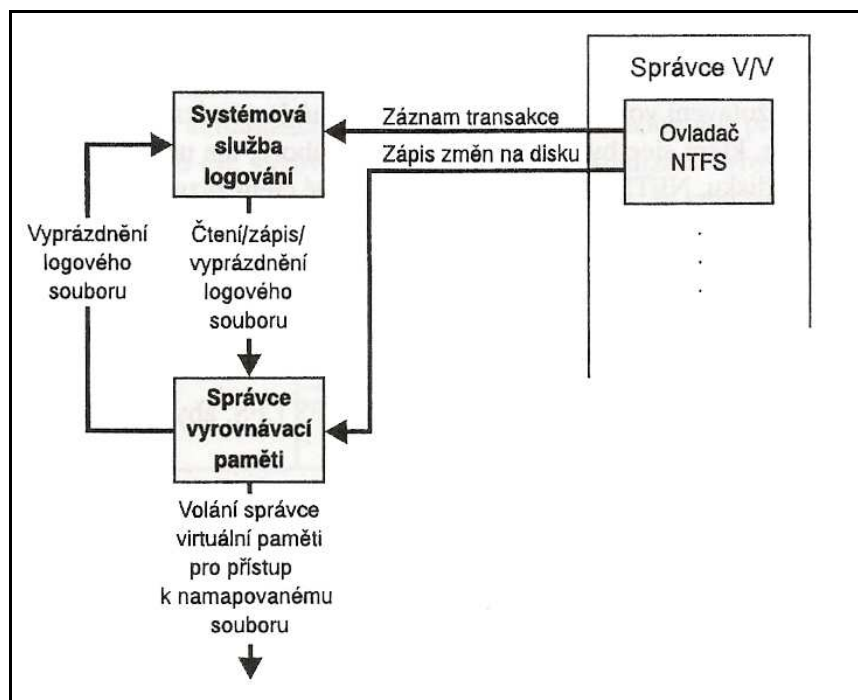
Předcházející příklad komprese řídkého souboru je poněkud vyumělkovaný. Popisuje "kompresi" v případě, že celá oblast souboru je vyplněna nulami a zbylá data nejsou kompresí dotčena. Data ve většině souborů nejsou nulová, ale mohou být přesto komprimována nějakou aplikací s kompresním algoritmem.

V NTFS může uživatel nastavit kompresi na jednotlivý soubor nebo všechny soubory v adresáři. Při kompresi souboru NTFS rozdělí souborová data do kompresních jednotek po 16 clusterech (to odpovídá 8 KB u clusterů po 512 bytech). U některých dat v souboru nemusí přinést komprese výraznou úsporu, proto NTFS zkoumá, zda komprese jednotlivých kompresních jednotek přinese úsporu aspoň jednoho clusterů. Pokud úspora aspoň jednoho clusterů není splněna, NTFS alokuje extend délky 16 clusterů a zapíše do něj data bez komprese.

NTFS udržuje souvislý prostor vyhrazený pro komprimované soubory vždy, pokud je to možné. Pokud jsou dva nebo více extendů souvislé, NTFS je načte dopředu stejně, jako to dělá u jiných souborů. Čtení a dekomprese souvislých dat na disku probíhá asynchronně dříve než si program vyžádá data, a následná operace čtení má data přístupná ve vyrovnávací paměti, což významně zvyšuje rychlost čtení [2].

Systémová služba logování (LFS - Log File Service)

LFS je řada rutin v režimu jádra systému v rámci ovladače NTFS, které NTFS používá při přístupu k logovému souboru. V principu je navržena pro logové a zotavovací služby více klientů, ale v současnosti ji využívá pouze NTFS. Volající (v tomto případě NTFS) předává LFS ukazatel na objekt otevřeného souboru - logový soubor. LFS inicializuje nový logový soubor nebo volá správce vyrovnávací paměti Windows NT a přistupuje k existujícímu logovému souboru ve vyrovnávací paměti (viz Obr. 13.).



Obr. 13. Systémová služba logování (LFS)

Následující přehled uvádí postupy, kterými systém zaručuje zotavení svazku:

- Před provedením jakékoli změny na svazku volá NTFS LFS, aby zaznamenala transakci do logového souboru.
- NTFS mění data na svazku (také ve vyrovnávací paměti).
- Správce vyrovnávací paměti volá LFS, aby zapsala logový soubor na disk. (LFS realizuje zápis na disk zpětným vyvoláním správce vyrovnávací paměti s informací, které stránky paměti má zapsat na disk (viz Obr. 13.).
- Po zápisu logového souboru na disk správcem paměti se provedou samotné změny na disku (vlastní transakce).

Tyto kroky zajišťují, že neúspěšné změny v souborovém systému lze při zotavení souborového systému opakovat nebo zrušit podle záznamů v logovém souboru.

Zotavení

Po každém startu systému, při prvním přístupu k NTFS disku provádí NTFS automaticky zotavení. Pokud zotavení není nutné, je tento proces bezvýznamný. Zotavení závisí na dvou tabulkách, které si NTFS udržuje v paměti:

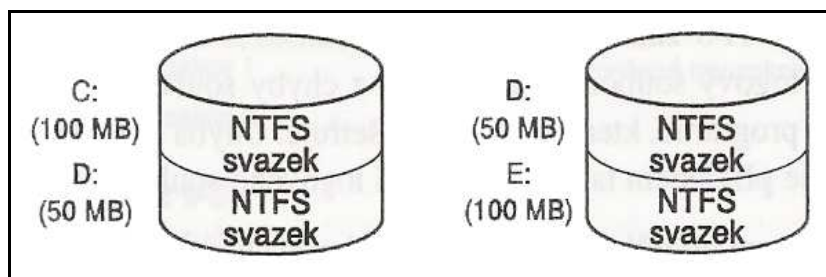
- **Tabulka transakcí** (transaction table) obsahuje transakce, které byly započaty, ale ještě nebyly dokončeny. Částečné operace těchto transakcí musí být během zotavení zrušeny.
- **Tabulka modifikovaných stránek** (dirty page table) obsahuje stránky vyrovnávací paměti, které obsahují změny souborového systému a nebyly ještě zapsány na disk. Tato data musí být během zotavení zapsána na disk.

NTFS každých 5 vteřin zapisuje do logového souboru záznam kontrolního bodu. Při zotavení NTFS prohlíží logový soubor celkem třikrát. Při prvním průchodu jej načte do paměti, aby minimalizoval V/V operace [2]. Každý průchod má zvláštní účel:

- 1) Průchod analyzuje
- 2) Průchod zaznamenává transakce
- 3) Průchod ruší transakce

Sady svazků

Sada svazků je jeden logický disk složený z maximálně 32 oblastí na jednom nebo více discích. Na Obr. 14. je diskový svazek 100 MB označený písmenem D:. Tato sada svazků byla vytvořena z poslední třetiny prvního disku a první třetiny druhého disku. Sady svazků se využívají pro spojení menších oblastí volného diskového prostoru do větších svazků nebo pro vytvoření rozsáhlých svazků ze dvou nebo více menších disků.



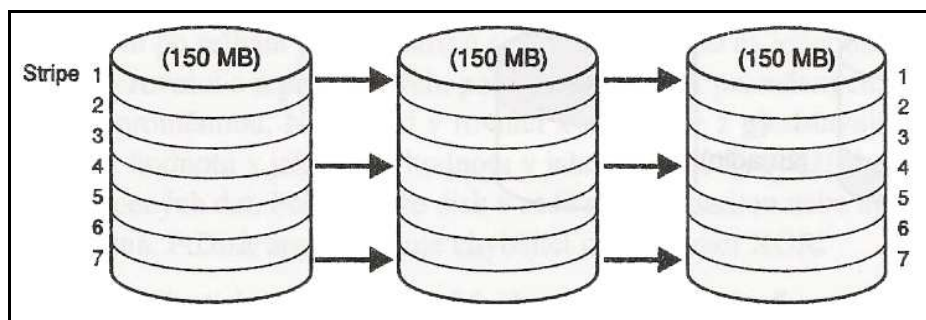
Obr. 14. Sada svazků

Sada svazků naformátovaná v NTFS může být rozšiřována o další volné oblasti nebo disky aniž by byla ovlivněna uložená data. Toto je jedna z největších výhod, kterou

umožňuje implementace všech dat na NTFS disku ve tvaru souborů. NTFS může dynamicky zvětšovat velikost logického svazku, protože bitová mapa, která zachycuje stav vyhrazeného prostoru na disku je bitmapový soubor. Bitmapový soubor může být rozšířen o jakýkoli prostor připojený do diskového svazku. Na druhé straně dynamické rozšíření FAT disku by vyžadovalo rozšíření FAT tabulky, což by mělo za následek ztrátu dat na disku.

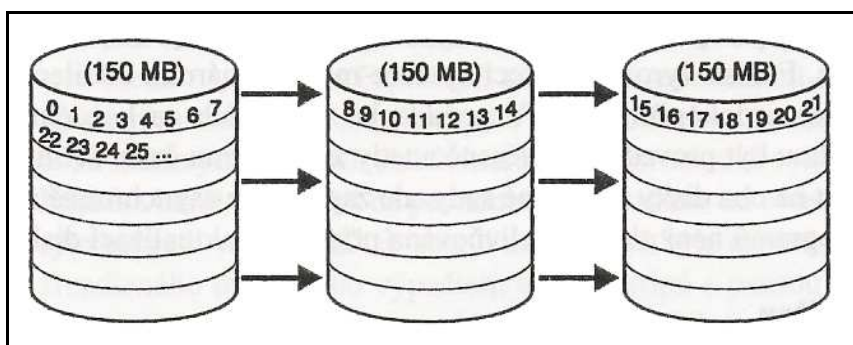
Sady stripů

Sada stripů (proužků) je řada oddílů, jedna na fyzický disk, které utilita Disk Administrator spojuje do jednoho logického svazku. Na Obr. 15. je sada stripů složená ze tří oddílů, na každém disku jeden oddíl. Oddíl v sadě stripů nemusí zahrnovat celý disk, jedinou podmínkou je, že oddíly na všech discích musí mít stejnou velikost [2].



Obr. 15. Sada stripů

Souborovému systému se jeví sada stripů jako jeden logický svazek 450 MB. NTFS přistupuje k fyzickým sektorům disků tak, jako by byly číslovány postupně na jednotlivých stripech (viz Obr. 16.)



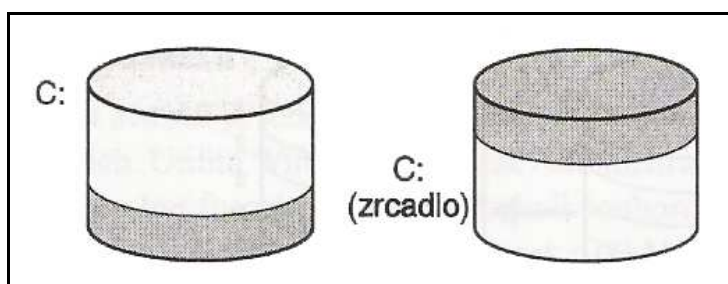
Obr. 16. Logické číslování fyzických sektorů v sadě stripů

Každý stripe je relativně malý - 64 KB (hodnota je zvolena tak, aby se omezila individuální čtení a zápisy na jediný disk) a data jsou rovnoměrně distribuována mezi disky. Stripy tedy zvyšují pravděpodobnost, že jednotlivé čekající operace čtení a zápisů

jsou směřovány na různé disky. Protože jsou data na všech třech discích přístupná současně, redukuje se, zvláště u výrazně zatížených systémů, časy V/V operací [2].

Zrcadlené sady

U **zrcadlených sad** je obsah oddílu na jednom disku duplikován v oddílu stejné velikosti na druhém disku. Zrcadlená sada je na Obr. 17. Zrcadlená sada může být formátována v jakémkoli souborovém systému podporovaném Windows NT.



Obr. 17. Zrcadlená sada

Zrcadlené sady mohou zlepšit průchodnost V/V operací velmi zatížených systémů. Když je V/V aktivita vysoká, FtDisk vyrovnává čtecí operace mezi primárním oddílem a zrcadleným oddílem (počet nedokončených V/V požadavků čekajících na každém disku). Dvě čtecí operace mohou být provedeny současně a tedy za polovinu času. Při modifikaci disku se musí zapsat na oba disky zrcadlené sady, ale zapisuje se asynchronně a výkonnost uživatelských programů není obecně ovlivňována přídatnou aktualizací disků [2].

4.2.5 Shrnutí

Cílem vývoje NTFS bylo vytvoření souborového systému, který bude nejen spolehlivý ale i rychlý. Přesto výkon V/V operací ve Windows NT není zajištěn výhradně implementací NTFS. Rychlost V/V operací je ve velké míře dána spoluprací NTFS a správce vyrovnávací paměti Windows NT. Touto spoluprací bylo dosaženo také výtečné úrovně spolehlivosti uložených dat jak na systémech pracovní stanice, tak i na serveru.

5 SOUBOROVÉ SYSTÉMY NA BÁZI OPERAČNÍCH SYSTÉMŮ UNIX

Souborový systém tvoří metody a struktury dat, pomocí kterých operační systém udržuje záznamy o souborech na discích a diskových oblastech. Jde tedy o způsob, jakým jsou soubory na disku organizované. Tento termín se ale používá i k označení diskové oblasti nebo disku, na kterém se ukládají soubory, nebo ve významu typu souborového systému. Takže když někdo řekne „mám dva souborové systémy“, může mít na mysli to, že má disk rozdělen na dvě diskové oblasti, nebo to může znamenat, že používá „souborový systém ext2“, tedy určitý typ souborového systému. Rozdíl mezi diskem či diskovou oblastí a souborovým systémem, který je na nich vytvořený, je dost podstatný. Jen některé programy (mezi nimi – zcela logicky – programy, pomocí kterých se souborové systémy vytváří) pracují přímo se sektory disku nebo diskovými oblastmi. Jestli na nich byl předtím vytvořen systém souborů, bude po spuštění takovýchto programů zničen nebo vážně poškozen. Většina aplikací ale pracuje se souborovým systémem. Proto je nelze použít na diskové oblasti, na které není vytvořen žádný systém souborů (nebo na které je vytvořen souborový systém nesprávného typu). Předtím, než bude možné diskovou oblast nebo disk použít, je potřeba je inicializovat – musí se na ně zapsat určité datové struktury. Tento proces se označuje jako vytvoření souborového systému. Většina souborových systémů na bázi UNIX má podobnou obecnou strukturu, v dalších podrobnostech se ale celkem dost liší. Mezi ústřední pojmy patří superblok, inode, datový blok, adresářový blok a nepřímý blok. **Superblok** obsahuje informace o souborovém systému jako celku, například jeho velikost (zrovna u této položky závisí přesné hodnoty na konkrétním souborovém systému). **Inode** obsahuje všechny informace o souboru kromě jeho jména. Jméno souboru je uloženo v adresáři společně s odpovídajícím číslem inode. **Adresářová položka** obsahuje jména souborů a čísla inodů, které tyto soubory reprezentují. Inode dále obsahuje čísla **datových bloků**, v nichž jsou uložena data souboru, který daný inode zastupuje. V inode je ale místo jenom pro několik čísel datových bloků. Když je jich potřeba víc, je dynamicky alokováno víc místa pro další ukazatele na datové bloky. Tyto dynamicky alokované bloky jsou uloženy v **nepřímém bloku**. Jak jejich název naznačuje, v případě, že je potřeba najít datový blok, musí se nejdřív najít jeho číslo v nepřímém bloku.

Souborové systémy na bázi UNIX obvykle umožňují vytvořit v souboru díru, a to pomocí systémového volání `lseek()`. Vypadá to tak, že souborový systém předstírá, že je na

konkrétním místě souboru uložen blok nulových bajtů, nicméně pro něj není vyhrazen žádný sektor pevného disku (to znamená, že takovýto soubor zabírá o něco méně diskového prostoru). S tím se můžete setkat zvláště často u malých binárních souborů „sdílených“ knihoven Linuxu, některých databází a v několika dalších speciálních případech. Díry jsou implementovány prostřednictvím speciální adresy datového bloku v nepřímém bloku nebo inode. Tato zvláštní adresa znamená, že pro určitou část souboru není alokován žádný datový blok, tedy že je v tomto souboru díra.

Výběr konkrétního souborového systému závisí na dané situaci. V případě, že jsou kompatibilita nebo jiná omezení nutnou podmínkou výběru některého souborového systému jiného operačního systému, nezbyvá než použít tento nepůvodní systém souborů. Pokud nejste ve výběru omezení, bude pravděpodobně nejrozumnější používat ext2, protože má ze všech uvedených systémů souborů nejvíce možností a nemá nedostatky z hlediska výkonu [7].

5.1 Souborový systém Ext2

Název znamená **Second Extended File System (ext2)**. Je to souborový systém používaný OS Linux, Hurd a dalšími. Je celkem oblíbený, protože je velmi rychlý. Dokonce se tvrdí, že je rychlejší než Fast File System z FreeBSD. Je ovšem mnohem rychlejší než všechny ostatní běžně známé. Obsahuje algoritmus na inteligentní alokaci volného místa, proto dochází k fragmentaci jen velmi málo. Např. při defragmentaci disku většinou uživatel zvláštní zrychlení ani nepozná. Přesto, že na disku fragmentace je, jsou jednotlivé fragmenty souboru tak blízko u sebe, že není zpomalení tak drastické jako v OS Windows.

5.1.1 Základní specifikace:

- navrhnul Rémy Card a Wayne Davison
- implementoval Rémy Card, založen na ext, uvolněn v lednu 1993
- vylepšen několika dalšími programátory (Stephen Tweedie - kernel code, Theodore Ts'o - utility)
- maximální velikost celého souborového systému je 4 TB
- maximální velikost souboru 2 GB
- maximální délka jména 255 znaků

- maximální velikost bloku 4 KB (používá se 1 KB)

Souborový systém ext2 podporuje standardní typy souborů Unixu: běžné soubory, složky, speciální soubory zařízení a symbolické odkazy. Dokáže pracovat na souborových systémech opravdu velkých oddílů. Zatímco originální kód kernelu omezoval maximální velikost souborového systému na 2 GB, zapracování na vrstvě VFS zvýšilo tento limit na 4 TB. Takže je možné používat velké disky, aniž by bylo třeba je dělit na mnoho oddílů.

Ext2 nabízí použití dlouhých názvů souborů. Používá položky složky s proměnnou délkou, maximální délka je 255 znaků a tento limit by mohl být rozšířen na 1012, pokud by bylo třeba. Souborové atributy dovolují uživateli modifikovat chování kernelu, když manipulují se sadou souborů. Kdokoli může nastavit atributy souboru nebo složky. Ext2 umožňuje administrátorovi zvolit při vytváření systému souborů velikost logických bloků. Velikost může být 1024, 2048 nebo 4096 bytů. Na druhou stranu použitím velkých bloků dochází zase k většímu plýtvání místa na disku. Bloky jsou rozděleny na skupiny bloků, skupina bloků obsahuje:

- **superblok** (základní popis souborového systému)
- **deskriptor skupiny bloků** (každá skupina bloků má svůj deskriptor který ji popisuje)
- **bitmapu bloků**
- **bitmapu inodů**
- **tabulku inodů**
- **datové bloky**

Základní stavební kámen v souborovém systému ext2 představuje inode. Nazýváme tak strukturu, kterou je zastoupen každý soubor. Každá inoda obsahuje popis souboru:

- **mód** (určuje typ souboru a přístupová práva)
- **informace o vlastníkovi** (uživatelský a skupinový identifikátor vlastníka souboru)
- **časové značky** (čas, kdy byl inode vytvořen a kdy byl naposledy modifikován)
- **velikost souboru**
- **ukazatel na blok dat**

5.1.2 Symbolický odkaz

Další funkcí souborového systému ext2 je realizace rychlých symbolických odkazů. 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 zachovává stopu se stavem souborového systému. V superbloku je použita speciální položka kódem kernelu k indikování stavu systému souborů. Když se souborový systém vytvoří v módu read/write, jeho stav se nastaví na "Not Clean". Když se odmountuje nebo namountuje v módu read-only, jeho stav je nastaven na "Clean". V čase bootování, ověřování souborového systému tuto informaci použije pro rozhodování, má-li se systému souborů ověřovat. Kód kernelu také v této položce zaznamenává chyby. Jestli se najde nějaká neshoda, systém souborů je označen jako "Erroneous". Při ověřování systému souborů dochází k tomuto testování, aby se spustila kontrola souborového systému bez ohledu na zřejmý stav "Clean". Každé přeskočení ověřování systému souborů může někdy být nebezpečné, proto ext2 poskytuje dva způsoby spouštění ověřování v pravidelných intervalech. Dále nabízí nástroje k vyladění chování systému souborů, k tomu slouží program tune2fs.

Souborový systém ext2 je velmi rozšířený a oblíbený, ačkoliv nemá podporu malých souborů jako ReiserFS. Některé souborové systémy ext2 jsou opravdu velké, u nich pak trvá kontrola integrity dat v případě výpadku velmi dlouhou dobu (i několik hodin). Během doby, kdy se souborový systém kontroluje, jsou aplikace na počítači mimo provoz. Z těchto důvodů byl ext2 zdokonalen [13].

5.2 Souborový systém Ext3

5.2.1 Charakteristika

Ext3 byl napsán Dr. Stephem Twediem jako nadstavba standardního souborového systému Linux - ext2, který je již dlouhou dobu stabilním bodem v OS Linux a se svým

předchůdcem je také obousměrně kompatibilní. To je také největší trumf ext3 oproti jeho konkurentům a možná i proto zůstává nejrozšířenější na platformách OS Linux.

V dnešní době jsou některé souborové systémy ext2 opravdu velké, u nich pak trvá kontrola integrity dat v případě výpadku proudu velmi dlouhou dobu (i několik hodin). Během doby, kdy se souborový systém kontroluje, jsou aplikace na počítači mimo provoz. Hlavním důvodem vzniku ext3 je co nejvyšší dostupnost služeb i v případě výpadku proudu. Nedisponuje ovšem vlastnostmi jako podpora extrémně rozsáhlých souborů. Jedním z nejdůležitějších cílů byla absolutní kompatibilita mezi ext2 a ext3 a to v obou směrech. Můžeme použít existující souborový systém ext2, vytvořit žurnál a připojit ho jako ext3 a dostáváme žurnálovací souborový systém. Naopak pokud máme souborový systém ext3 a předtím došlo k bezpečnému odpojení tohoto souborového systému, můžeme ho připojit znovu jako ext2 a vše funguje, jak má.

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 [12].

5.2.2 Výkon

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 [11].

5.2.3 Budoucnost

Ext3 není ani prvním ani posledním žurnálovým souborovým systémem, který se v Linuxu objeví, ovšem jeho vlastnosti a především kompatibilita z něj činí ideální volbu pro všechny uživatele, kteří doposud pošilhávali po dovednostech žurnálového souborového systému, ale odrazovaly je složité procedury s převodem dat. Nyní, když jej můžete získat bezbolestně s distribucí, nezbyvá, než dát sbohem historii v podobě ext2 a svěřit svá data dokonalejšímu ext3 [11].

5.3 Souborový systém ReiserFS

Jedná se o první souborový systém na bázi UNIX, který podporoval žurnálování. Hlavní osoba, která za tímto projektem stojí, je Hans Reiser, po kterém byl tento souborový systém pojmenován. Ovšem na vývoji s ním spolupracoval ještě velký tým spolupracovníků.

ReiserFS má spoustu vlastností, které ostatní souborové systémy neobsahují. Jak jsem již řekl, jedná se o žurnálovací souborový systém, tudíž jednou z výhod, které poskytuje, je zajištění konzistence dat při výpadku proudu a také zrychlení obnovení systému oproti fsck (program pro kontrolu stavu dat).

ReiserFS je založen na rychlém „balancovaném“ stromu. Je tak zrychlena práce např. s velkým množstvím položek v adresáři. Už 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 souborový systém ReiserFS několik do jednoho bloku. Použitím tohoto souborového systému tedy šetříme nezanedbatelnou měrou prostor na našem záznamovém médiu. 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 nevyužitě 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, tudíž 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 [10].

5.3.1 Parametry zrychlující práci FS

Jak jsem již uvedl, ReiserFS má i některé své nedostatky (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 souborového systému o posledním přístupu ke všem položkám souborového systému. Další volbou, která zrychlí práci s souborovými systémy je `notail`. Tím zrušíme ukládání konců souborů do jednoho bloku. Ztratíme tím sice nějaké místo na disku, ale dojde k zrychlení práce souborového systému [10].

5.3.2 Změna velikosti souborového systému

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`) [10].

5.3.3 Shrnutí

K dalším výhodám souborového systému ReiserFS patří podpora žurnálovacího souboru na odděleném diskovém oddílu a dynamické alokování inod. Jak je již uvedeno výše je u tohoto souborového systému také možnost zvětšit souborový systém online. Poskytuje i podporu špatných bloků a pomocí externího patche i možnost `quoty`.

ZÁVĚR

Byla vytvořena studijní pomůcka pro předmět "Operační systémy", která je zaměřena na popis elementárních principů souborových systémů používaných v dnešních počítačových systémech.

Předností této práce je způsob zpracování použitých informací, díky kterému má uživatel k dispozici přehledné materiály týkající se souborových systémů. Cílem bylo vytvořit takový zdroj informací, ze kterého si student vezme co nejvíce nových poznatků z oblasti souborových systémů, popřípadě si ozřejmí informace, které mu dosud nebyly zcela jasné.

Pro přednášejícího může být pak nemalým pomocníkem prezentace, vypracovaná ve formátu MS PowerPoint, kde se nachází zestručnění všech kapitol obsažených v této bakalářské práci. To je vypracováno takovým způsobem, aby studenti na přednášce měli dokonalý přehled o tématu, který je docílen současným použitím výstižného vyjádření konkrétního tématu a názorného obrázku. Tím se dle mého názoru stává vhodným doplňkem přednášek předmětu "Operační systémy". Studentům pak může při samostudiu pomoci zjednodušená HTML verze určená pro publikování na síti Internet. Zde si pak student může obdobně jako při prezentaci na přednáškách dohledat informace, které mu buď unikly nebo mu nejsou zcela jasné.

Tato studijní pomůcka obsahuje množství informací z oblasti souborových systémů a to jak obecného pojednání společného pro většinu souborových systémů, tak informací o konkrétních souborových systémech a to pracujících pod operačními systémy MS Windows a systémů pracujících na bázi UNIX. V úvodních dvou kapitolách jsou rozebrány informace, týkající se základů nezbytných pro další pochopení fungování souborového systému. V další kapitole jsou popsány vlastnosti společné pro různé souborové systémy, a to např.: fyzická a logická organizace souborů a adresářů na disku, šifrování souborů, odkazy, metadata, mountování a nebo způsob uspořádání souborů.

Díky této bakalářské práci lze předpokládat s nemalým přínosem pro studenty i přednášející. Lze se domnívat, že nalezne díky současné aktuálnosti informací své místo nejen mezi studenty a vyučujícími fakulty technologické, ale jistě i na jiných vysokých školách, které mají ve svých učebních osnovách předměty, které se zabývají informačními technologiemi.

SEZNAM POUŽITÝCH ZDROJŮ

Monografické publikace:

- 1) Deitel, H. M.: Operating Systems, Prentice Hall, 2004
- 2) Solomon, D. A., Windows NT pro administrátory a vývojáře, Computer Press, Brno, 1999
- 3) Čada, O.: Operační systémy, Grada, 1993
- 4) Klimeš, C.: Operační systémy 1. , Ostravská univerzita, Ostrava, 2003
- 5) Tanenbaum, A. S.: Modern operating systems, Prentice Hall, 2002
- 6) Microsoft Corporation, Resource Kit Microsoft Windows XP, Computer Press, 2002
- 7) Kolektiv autorů: Linux – Dokumentační projekt, Computer Press, 2003
- 8) Mueller, S.: Osobní počítač, Computer Press, Praha, 2003.

WWW stránky

- 9) Disk [online].[cit. 2005-02-14] Dostupný z URL:
<<http://ivt.wz.cz/vesely/disk.htm>>.
- 10) Moderní souborové systémy - ReiserFS [online].[cit. 2005-02-22] Dostupný z URL: <<http://www.abclinuxu.cz/clanky/show/12663>>.
- 11) Ext3 - žurnálování pro každého - Root.cz [online].[cit. 2005-02-22] Dostupný z URL: <<http://www.root.cz/clanky/ext3-zurnalovani-pro-kazdeho/>>
- 12) Moderní souborové systémy - ext3 [online].[cit. 2005-02-22] Dostupný z URL: <<http://www.abclinuxu.cz/clanky/show/12344/>>
- 13) Paměti a paměťová média - projekt do PCK [online].[cit. 2005-05-13] Dostupný z URL: <<http://www.umel.feec.vutbr.cz/~danekl/Archiv.PCK/!!!2001!!!/0103/>>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

FS - (File system) - souborový systém

OS - (Operating system) - operační systém

V/V - Vstupně/výstupní

SEZNAM OBRÁZKŮ

Obr. 1. Fyzická reprezentace souboru na disku	12
Obr. 2. Dvouúrovňové hierarchické uspořádání souborového systému	15
Obr. 3. Příklad hierarchického uspořádání souborového systému	16
Obr. 4. Odkazy v souborovém systému	17
Obr. 5. Mountování souborového systému	21
Obr. 6. Základní struktury souborového systému řady FAT na pevném disku	24
Obr. 7. Jednotlivé části V/V systému Windows NT	36
Obr. 8. Datové struktury NTFS	37
Obr. 9. Možné konfigurace disku	38
Obr. 10. Sektory a clustery na disku	38
Obr. 11. Souborový záznam NTFS v MFT	39
Obr. 12. Typy jmen souborů ve Windows NT	40
Obr. 13. Systémová služba logování (LFS)	42
Obr. 14. Sada svazků	43
Obr. 15. Sada stripů	44
Obr. 16. Logické číslování fyzických sektorů v sadě stripů	44
Obr. 17. Zrcadlená sada	45

SEZNAM TABULEK

Tab. I. Příklady polí u adresářů a souborů.....	13
Tab. II. Hlavní spouštěcí záznam a hlavní tabulka rozdělení disku	27
Tab. III. Struktura adresáře souborového systému FAT.....	29
Tab. IV. Počty záznamů v kořenovém adresáři.....	30
Tab. V. Standardní velikost clusteru pro diskety.....	32
Tab. VI. Standardní velikost clusteru pro různé velikosti pevných disků a jejích oddílů...	32