Forrás: http://almafan.hu (Az oldal forrása részben a pcguide.com oldal)

A merevlemez logikai szerkezete

A partíciók

A lemezen kialakított fizikai szerkezet önmagában még nem teszi lehetővé, hogy állományokat is tároljunk rajta. Ehhez a következő lépéseket kell még megtenni:

1. *A lemez particionálása*
2. *A partíción a logikai fájlszerkezet kialakítása (logikai formázás)*

A partíció fogalma

A partíció a merevlemez egy önálló logikai egysége, amely fájlrendszer tárolására alkalmas. Ahhoz, hogy egy merevlemezt használni tudjunk, legalább egy formázott partíciót kell tartalmaznia.

Egy lemez összefüggő fizikai szerkezetét tehát logikailag egymástól logikailag független részekre bontjuk. Ezeket a részeket nevezzük partícióknak, a felosztás folyamatát pedig particionálásnak. Háromféle partíciót különböztetünk meg. Vegyük sorban.

A partíciók fajtái PC-s rendszerekben

Az egyik (és manapság leggyakrabban használt) partíció az ***elsődleges (primary) partíció***. Az elsődleges partíción formázás után állományok tárolhatók. Legfontosabb tulajdonsága, hogy a különböző operációs rendszerek szeretnek az elsődleges partícióról indulni. Egy merevlemezen az elsődleges partíciók száma (PC-s rendszerekben) 0, 1, 2, 3 vagy 4. Az elsődleges partíciókhoz a Windows operációs rendszer általában betűt rendel (C:, D: stb.), de lehetőség van már itt is arra, hogy egy partíció egy könyvtárban jelenjen meg. A UNIX típusú operációs rendszerek a partíciókat (feltéve ha leformáztuk őket) egy-egy alkönyvtárba csatolják.

A másik partíciótípus a ***kiterjesztett (extended) partíció***. A kiterjesztett partíció nem formázható, csupán újabb, úgynevezett logikai partíciók hozhatók rajta létre. Extended partíció száma (PC-s rendszerekben) 0 vagy 1 lehet.

A harmadik partíciótípus tehát a ***logikai partíció***, mely az előzőek szerint az extended partíción található. Számuk nem korlátozott. A Windows ezekhez is betűt rendel, míg a UNIX tipusúak alkönyvtárba csatolják. Általában az operációs rendszerek logikai partíciókról nem tudnak elindulni.

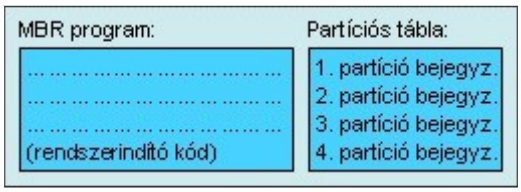
Partíciók

Honnan tudja a rendszer, hogy hány partícióra van felosztva a merevlemez, illetve hol vannak ezek a partíciók? A különböző rendszerek más-és más megoldást alkalmaztak.

MBR - Master Boot Record

PC-s rendszerekben a lemez 0. szektora (melynek neve MBR, azaz Master Boot Record) tartalmaz egy partíciós táblát és egy kis programot. A partíciós tábla felfogható a merevlemezen levő partíciók

'tartalomjegyzékének'. Ahogy egy könyv tartalomjegyzéke megmutatja, hogy a könyv bizonyos fejezetei a könyv hányadik oldalán kezdődnek, úgy a Master boot record-ban levő partíciós tábla is megmutatja, hogy a partíciók a merevlemezen hol találhatóak. A partíciós táblának 4 sora van, és ezekben vannak eltárolva az elsődleges partíciók és a kiterjesztett partíció adatai. A kiterjesztett partíció által tartalmazott logikai partíciók táblázata nem a MBR-ben, hanem a kiterjesztett partícióban található. Mivel 4 sor van, ezért lehet maximum 4 partíció. Ha ebből mind a 4 elsődleges, akkor nem lehet kiterjesztett (így logikai sem), ha meg van 1 kiterjesztett (több nem lehet), akkor maximum mellette 3 elsődleges található.



Az MBR felépítése

A gép indulásakor miután a BIOS elvégezte a POST (Power on Self Test) feladtait, elindítja az MBR kis programját, mely beolvassa a partíciós táblát, ellenőrzi, melyik partíció az aktív (azaz boot-olható), és beolvassa annak az első szektorát, amit boot szektornak nevezünk. (Az MBR szintén egy boot szektor, de különleges státusza és emiatt neve van.) Ez a boot szektor egy másik kis programot tartalmaz, mely beolvassa az operációs rendszer első részeit az adott partícióról (ha boot-olható), és elindítja azt.

Mint minden szektor, a MBR is 512 bájt terjedelmű. A Master boot record (MBR) partíciós táblájában mindegyik partíció bejegyzése a következő információk leírásából áll:

- aktív állapotot jelző bájt: azt jelzi, hogy a partíció aktív-e vagy sem (Az aktív partícióról próbálja meg betölteni az operációs rendszert)
- partíció kezdete és vége (vagyis a partíció melyik cilinder melyik lemezoldalának melyik szektoránál kezdődik, illetve melyiknél végződik)
- partíció mérete (szektor darabszámban kifejezve) - maximum 2 TerraByte méretű lehet egy partíció
- partíció típusa (ami lehet elsődleges vagy kiterjesztett)

A MBR épsége nagyon fontos. Ha a MBR-ben levő partíciós tábla megsérül, akkor a számítógép indítását követően a MBR program nem találja meg az indítandó operációs rendszer partícióját, következésképpen az operációs rendszer nem fog elindulni. Amennyiben a MBR program a partíciókat nem találja meg, a merevlemez partíción tárolt fájlok és könyvtárak sem lesznek elérhetőek. Ha a MBR program sérül meg, akkor a merevlemezről nem fog az operációs rendszer bootolni.

A PC-s rendszerekben a BOOT-olás folyamata a következő

A bootolás az a folyamat, mely a számítógép bekapcsolásától az operációs rendszer betöltődéséig tart. A bootolás lépései a következők:

1. A bootolás során először az alaplap BIOS rendszere kapja meg az vezérlést. A BIOS különféle ellenőrzéseket végez annak megállapítására, hogy a számítógép hardver eszközei (RAM memória, billentyűzet, merevlemez, stb.) rendben vannak-e. Ezt a néhány másodpercig tartó ellenőrzési folyamatot power on self test-nek vagy rövidítve POST-nak nevezik.
2. Miután a POST ellenőrzések sikeresen végrehajtódnak, a BIOS megnézi, hogy milyen hardver eszközről kell végrehajtani a bootolást (lehetséges esetek: floppy, merevlemez,

CD/DVD, sőt újabban USB flash drive). A továbbiakban azt az esetet vizsgáljuk, amikor merevlemezről indul az operációs rendszer bootolása.
3. A BIOS a merevlemezről a MBR-t (vagyis az ebben levő MBR programot és a partíciós tábla tartalmát) beolvassa a RAM memóriába, majd itt átadja a vezérlést a MBR programnak.
4. A MBR program leellenőrzi, hogy a partíciós táblában melyik az aktív partíció, és a partíciós táblából "kinézi", hogy az aktív partíció a merevlemezen hol helyezkedik el.
5. A MBR program a merevlemezen megkeresi az aktív partíciót, majd ennek az első szektorát betölti a memóriába. Egy bootolható partíción belül az első szektor(oka)t (ez nem azonos a MBR-vel!) rendszerbetöltő szektornak (boot sector vagy boot record) nevezik.
6. Az aktív partíció boot rekordja egy másik, saját kis boot programot (bootstrap code) tartalmaz. A MBR program átadja ennek a kis programnak a vezérlést.
7. Ez a bootstrap program elkezdi az aktív partíción található operációs rendszert a merevlemezről betölteni a memóriába. Ettől kezdve már az operációs rendszer saját boot lépései következnek.

GPT - Globally Unique Identifier Partition Table (Forrás: lacy.hu)

A merevlemezek méretének drasztikus növekedése szükségessé tette egy újabb partíciós rendszer kifejlesztését. Ez a GPT. Tulajdonságai:

- Merevlemezenként 128 partíció támogatása.
- Egy partíció maximális mérete elméletben 18 exabájt. (18 millió TB)
- A GPT a működéshez szükséges adatokat a partíciókban tárolja.
- Tartalék elsődleges és biztonsági partíciók használata a partíciós adatok struktúrájának jobb helyreállíthatósága érdekében.
- Következő rendszerek támogatják: Windows XP x64, Windows Server 2003/2008, Windows Vista. Látszik, hogy a listában nem szerepel a 32bites XP, ezért a lemezkezelőben ilyenkor a GPT - védőpartíció kiírás fog szerepelni!

A GPT használatának korlátai

- A Microsoft bevezetett egy NTFS limitálást a Windowsokban, azaz a kötetek maximális mérete nem lehet nagyobb 256 TB-nál. Tehát az NTFS fájlrendszer soha nem fogja kihasználni a 18 exabyte-os partícióméretet.
- Vista alatt csak akkor lehet GPT meghajtóról bootolni, ha a rendszer Extensible Firmware Interface (EFI) -t használ, BIOS helyett. Adattárolásra természetesen használhatunk GPT-t attól még, hogy a rendszerünk MBR lemezről bootol. (A Machintos rendszerek EFI-t használnak, tehát ezek tudnak innen bootolni.)
- Jó tudni, hogy a partíciók óriási mérete kecsegtető, ám azok hibaellenőrzésére kitalált chkdsk segédprogram nem tud a korral ennyire haladni. Ha probléma van a fájlrendszerrel és elindul a chkdsk ellenőrző könnyen elképzelhető, hogy egész nap vizsgálódni fog egy terrabájtos partíció esetén. Nem beszélve ha 18 millió terrabájtos és rajta van a teljes internet tegnapi biztonsági mentése. Erre a DFS (Distributed File System) adhat megoldást, ahol több kisebb kötetünk van és ezeket egy névtérbe foglalva gyakorlatilag csak 1 nagyobb kötetet látunk. Így hiba esetén elég csak a kisebb köteteket felülvizsgálni.

A BSD rendszerek

A BSD rendszerek szintén a lemez egy rekordjában (melynek neve Disklabel) tárolják a partíciók adatait. A disklabel 16 bejegyzést tartalmazhat, így 16 partíció lehet a lemezen, melyből lényegében 15 használható.

Logikai szerkezet - fájlrendszerek

Egy háttértár logikai szerkezete a rajta kialakított fájlrendszert jelenti. A számítástechnika egy fájlrendszer alatt a számítógépes fájlok tárolásának és rendszerezésének a módszerét érti, ideértve a tárolt adatokhoz való hozzáférést és az adatok egyszerű megtalálását is. Az adatokat fájlokban tároljuk, és könyvtárakban rendszerezzük. A fájlrendszer a logikai formázás során alakul ki.

A lemezen a fizikailag elérhető legkisebb adategység a szektor, melynek mérete 512 byte. A logikai szerkezetben a legkisebb elérhető adategység a klaszter, mely mindig egész számú szektorokból épül fel, tipikusan 4, 8, 16 vagy 32 kbyte mérettel. A fájlokat a rendszer tehát klaszterek sorozatában tárolja el, és azt, hogy egy fájl végigolvasásához melyik klasztert melyik után kell olvasni, egy táblázatban tárolja. Azt, hogy a fájl első klasztere hol van (tehát hol kell elkezdeni olvasni), a fájl nevével és egyéb adataival együtt külön tárolja.

FAT fájlrendszerek (FAT12, FAT16, FAT32)

A Microsoft operációs rendszerei számára a FAT (Fat Allocation Table, Fájl Allokációs Tábla) fájlrendszereket fejlesztette ki. A FAT tábla tartalmazza azokat az adatokat, hogy egy fájl tartalmának végigolvasásához melyik klaszter után melyik klasztert kell olvasni. A FAT tábla minden bejegyzése tehát egy klaszterre mutat. A 12 bites FAT tábla 12 bites számokkal azonosította a klasztereket, tehát maximum 4096 klasztert tudott kijelölni. A hajlékonylmezeknél alkalmazták. A 16 bites FAT tábla már 65536 klasztert azonosíthatott, maximum 2GB-os partíciókat lehetett vele kezelni. A FAT táblából fontossága miatt két példányt is eltároltak a lemezen, így az egyik megsérülése esetén a másikat lehett használni. Manapság a FAT típusú rendszerekből csak a 32 bitest használjuk, mely a 32 bitből 26-ot használ a klaszterek kijelölésére.

A klasszikus (12 és 16 bites) FAT táblában háromféle bejegyzés lehet:

- *egy másik klaszter sorszáma* (ezen a klaszteren kell folytatni a fájl olvasását)
- *EOF* (ez a klaszter a fájl utolsó klasztere)
- *0* (a klaszter egyik fájl tárolásában sem vesz részt, vagyis szabad helyet jelöl)

A FAT tábla bejegyzéseiből nem derülnek ki a következő információk:

- melyik klaszteren kezdődik a fájl
- a fájlok neve és egyéb nem a tartalmukban tárolt adataik
- az EOF-fal jelölt klaszeter hány bájt tartozik a fájlhoz

Ezek az információk a directory (könyvtár) részben találhatók. Kétfajta könyvtár rész van, a gyökérkönyvtár és az alsóbb szintű könyvtárak. Gyökérkönyvtár például a C:\, míg alkönyvtár a C:\GAMES. Alapvetően mindkét könyvtárterület a benne található fájlok tartalmukon felüli adatait tartalmazzák. Ezek a következők:

- a fájl neve (8+3 karakter)
- a fájl mérete byte-okban
- a fájl létrehozásának dátuma és ideje
- a kezdő klaszter sorszáma
- attribútumok

A fájl hosszából (mivel a rendszer tudja, hogy egy klaszterben hány byte van eltárolva) kiszámítható, hogy az utolsó, EOF-fal jelölt klaszterből még hány bájt tartozik a fájlhoz.

A klaszterek használatának következménye az, hogy egy fájl tárolásakor a merevlemezről a fájl byteokban számolt méreténél általában több szabad helyet veszítünk. Egy klasztert csak egy fájlhoz tud az allokációs tábla hozzárendelni. Bármekkora is a fájl, a rendszer csak a klaszterek egész számú többszörösét rendeli hozzá. Tegyük föl, hogy a klaszter mérete a partíción 4 kbyte. Legyen a tárolandó fájl mérete 1 kbyte. A rendszer lefoglal számára 1 klasztert, így veszendőbe megy 3 kbyte. Legyen a tárolandó fájl mérete 4kbyte+1byte. Az 1 byte miatt a rendszer lefoglal 2 klasztert, azaz 8 kbyte-ot, így veszendőbe ment majdnem 4 kbyte. Látható, minél több kis méretű fájlt tárolunk a partíción, annál több lesz a veszteség.

A FAT fájlrendszer korábbi változataihoz képest a FAT32 az alábbi továbbfejlesztett funkciókat tartalmazza:

- A FAT32 fájlrendszerben használható meghajtók maximális kapacitása akár 2 terabájt is lehet.
- A FAT32 hatékonyabban használja ki a rendelkezésre álló lemezterületet. Mivel kisebb (4 KB méretű) szektorcsoportokat használ (akár 8 GB kapacitású meghajtókon), 10–15 százalékkal hatékonyabban kezeli a lemezterületet, mint a nagyméretű, FAT vagy FAT16 fájlrendszerű meghajtók.
- A FAT32 sokoldalúbb, mint elődje. A FAT32 képes a gyökérmappa áthelyezésére, és a fájlallokációs tábla biztonsági másolatának használatára (az alapértelmezett példány helyett). Emellett a FAT32 rendszerű meghajtókon a rendszertöltő rekord kibővült a létfontosságú adatstruktúrák biztonsági másolatával. Ennélfogva az ilyen meghajtók a FAT16 rendszerű meghajtókhoz képest kevésbé hajlamosak arra, hogy egyetlen hiba miatt működésképtelenné váljanak.
- A FAT32 rugalmasabb. A FAT32 fájlrendszerű meghajtókon a gyökérmappa egy szokásos szektorcsoport-sorozat, ezért a meghajtó tetszőleges helyére áthelyezhető. A gyökérmappa-bejegyzések számára vonatkozó korábbi korlát megszűnt. A fájlallokációs tábla tükrözése letiltható, így az első példány helyett a tábla újabb példánya is aktívvá tehető. Ez a tulajdonság lehetővé teszi a FAT32 fájlrendszerű partíciók dinamikus átméretezését. Érdemes azonban megjegyezni, hogy bár a FAT32 kialakítása lehetővé teszi ezt a funkciót, a Microsoft nem építi bele a fájlrendszer első verziójába.

A FAT fájlrendszerek az állományokról nem tartalmaznak semmiféle felhasználóspecifikus adatot, tehát sem tulajdonosi, sem csoportjogokat nem rendel hozzájuk. Hiányzik belőlük a naplózási lehetőségek is, ezért a félbehagyott lemezműveletek (áramszünet, RESET esetén) hibákat hagytak a rendszerben. Ilyen hibák lehetnek a következők:

Árva klaszterek (Lost cluster): Olyan lefoglalt klaszterek, melyek egyetlen fájlhoz sincsenek hozzárendelve. Vagyis csak foglalják a helyet, de senki nem használja az adataikat.

Keresztcsatolt klaszterek (cross linked cluster): Olyan klaszterek, melyek több állományhoz is tartoznak. Ha egy-egy állomány klaszterei egy-egy láncolt listának tekinthetők, akkor ezek a láncolt listák valamelyik láncszemnél összeakadtak.

Ugyan nem hiba, de a rendszer teljesítményét jelentősen csökkentheti a töredezettség (fregmentáció) jelensége. A töredezettség annyit jelent, hogy a FAT (vagy más láncolási) táblában a folyamatos használat miatt (fájlok írása, törlése, módosítása) az ugyanazon fájlokhoz tartozó klaszterek fizikailag nem egymás után helyezkednek el, hanem össze-vissza. (Olyan jelenség, mintha egy könyv lapjai nem egymás után lennének fűzve, hanem állandó lapozásra kényszerítenék az olvasót.) Töredezettségmentesítő programokkal lehet a logikailag összetartozó klasztereket fizikailag is egymás után írni a merevlemezen. Meg kell jegyezni, hogy a merevlmezbe épített pár Mbyte-os cache hatékonysága drasztikusan csökken töredezett fájlrendszer esetén.

A fájlrendszer létrehozásakor (vagyis a formázáskor) lehetőség van klasterméret megadására, illetve meg lehet vizsgáltatni azt, hogy a szektorok hibátlanul írhatók és olvashatók. Ez a formázás idejét rendkívül megnöveli.

Az NTFS fájlrendszer

Az NTFS vagy New Technology File System (új technológiájú fájlrendszer) a Microsoft Windows NT és utódainak (Windows 2000, Windows XP, Windows Server 2003, stb.) szabványos fájlrendszere.

Az NTFS a Microsoft korábbi FAT fájlrendszerét váltotta le, melyet az MS-DOS és a korábbi Windows verziók esetén használtak. Az NTFS több újdonsággal rendelkezik a FAT fájlrendszerrel szemben, mint például a metaadatok támogatása, fejlettebb adatstruktúrák támogatása a sebesség, a megbízhatóság és lemezterület-felhasználás érdekében, valamint már rendelkezik hozzáférésvédelmi listával és megtalálható benne a naplózás is. A fő hátránya a korlátozott támogatottsága a nem-Microsoft operációs rendszerek oldaláról, mivel a pontos specifikáció a Microsoft szabadalma.

Az NTFS-en belül minden fájlokkal kapcsolatos információt (fájlnév, létrehozás dátuma, hozzáférési jogok, tartalom) metaadatként tárolnak. Ez teszi lehetővé, hogy az NTFS fájlrendszer rugalmasan továbbfejlszthető, hiszen az új funkciókhoz szükséges információk újabb metaadatként jelenhetnek meg.

Egy fájlrendszer naplót használnak magának a fájlrendszer integritásának (de nem az egyes fájloknak) a biztosítására. Az NTFS-t használó rendszerek biztonságosabbak, ami egy kiemelten fontos követelmény a Windows NT-k korábbi verzióinak instabil mivolta miatt.

Mivel az NTFS megvalósításának részletei nem nyilvánosak, így külső gyártóknak nagyon nehéz NTFS-t kezelő eszközöket előállítani.

Az NTFS fájlrendszer főbb tulajdonságai a következők:

- kvóta használata - a felhasználók csak bizonyos mennyiségű helyet használhatnak el
- titkosítás (Encrypting File System) - szimmetrikus titkosítással titkosítja a fájlrendszer adatait
- fájltömörítés - a fájlrendszer adatait automatikusan tömöríti
- kötet csatolási pont - más fájlrendszer gyökerét képes egy könyvtárba becsatolni, tehát nem kell arra betűjellel (pl D:) hivatkozni
- könyvtár csatolás - egy könyvtárba egy másik könyvtár jelenik meg
- hard link - hivatkozást készíthetünk egy másik fájlra (csak azonos köteten)

Az NTFS használatakor találkozhatunk pár korlátozással: A fájlrendszer legfeljebb 32 000 Unicode karakter hosszúságú útvonalakat támogat, melyek minden komponense (könyvtár vagy fájlnév) legfeljebb 255 karakter hosszúságú lehet, bizonyos neveket a rendszer fenntart saját használatra. Ennek oka, hogy az NTFS metaadatok szabályos (bár rejtett, és általában nem hozzáférhető) fájlokban tárolódnak, ezért a felhasználói adatállományok nem használhatják fel ezeket a neveket. Ezek a rendszerfájlok mindig a kötet gyökerében tárolódnak (és kizárólag a gyökérkönyvtárban vannak fenntartva). A fájlnevek: $Mft, $MftMirr, $LogFile, $Volume, $AttrDef, . (pont), $Bitmap, $Boot, $BadClus, $Secure, $Upcase és $Extend; a . és az $Extend mindkettő könyvtár, a többiek fájlok.

A Linux fő fájlrendszerei és tulajdonságaik *(Forrás: novell.com)*

Két vagy három évvel ezelőtt a fájlrendszer kiválasztása egy Linux-rendszerhez nem tartott tovább néhány másodpercnél (vagy Ext2, vagy ReiserFS). A 2.4 és újabb kernelek esetén azonban már egész sokféle fájlrendszer közül lehet választani. Az alábbiakban áttekintjük a fájlrendszerek alapvető működését és az általuk kínált előnyöket.

Fontos megjegyezni, hogy nincs olyan fájlrendszer, amely tökéletesen megfelelne mindenféle alkalmazáshoz. Minden fájlrendszernek vannak erősségei és gyengéi, amelyeket figyelembe kell venni. Emellett még a legkifinomultabb fájlrendszer sem helyettesíthet egy józan mentési stratégiát.

Az adatintegritás és adatkonzisztencia kifejezések ebben a fejezetben nem a felhasználói területen lévő adatok (az alkalmazások által a fájlokba írt adatok) konzisztenciájára utalnak. Az adatok konzisztenciájáról magának az alkalmazásnak kell gondoskodnia.

ReiserFS

A 2.4-es kernelkiadás egyik fontos eleme, a ReiserFS fájlrendszer kerneljavításként már a 2.2.x SUSE kernelek, a SUSE Linux 6.4 változat óta rendelkezésre áll. A ReiserFS-t Hans Reiser és a Namesys fejlesztőcsapat tervezte. Azóta bebizonyította, hogy hatékony alternatívát kínál a régi Ext2-vel szemben. Legfontosabb előnyei: jobb lemezterület-kihasználás, jobb lemezhozzáférési teljesítmény és gyorsabb helyreállás összeomlás után.

A ReiserFS előnyeiről részletesebben:

Jobb lemezterület-kihasználás

A ReiserFS fájlrendszerben az összes adat kiegyensúlyozott B*-fastruktúrába van szervezve. A fastruktúra jobban ki tudja használni a lemezterületet, mivel a kis fájlok közvetlenül a B*-fa levélcsomópontjaiban kerülnek tárolásra, nem pedig egy másik helyen és csak egy mutató mutat a tényleges tárolási helyre. Ezen felül a tárterület nem 1 vagy 4 kilobájtos egységekben kerül lefoglalásra, hanem az adatok pontosan a szükséges méretet foglalják el. Másik előnye az inode-ok dinamikus lefoglalása. Így a rendszer rendkívül rugalmas, szemben például az Ext2-vel, ahol az inode-ok sűrűségét a fájlrendszer létrehozásakor kell megadnunk.

Jobb lemezhozzáférési teljesítmény

Kis fájlok esetén az adatok és a „stat_data” (inode) információ általában egymás mellett kerül tárolásra. Ez az információ egyetlen lemez I/O-művelettel kiolvasható, azaz csak egy lemezhozzáférés szükséges a kívánt információ lekéréséhez.

Gyors visszaállítás rendszerösszeomlás után

A legutolsó metaadat-módosításokat nyomkövető napló segítségével a fájlrendszer ellenőrzése nagyon nagy fájlrendszerek esetén is csak néhány másodpercet vesz igénybe.

Megbízhatóság az adatnaplózás használatával

A ReiserFS az adatnaplózást és a sorbarendezett adatmódokat is támogatja, hasonló elven, mint ahogy az Ext3-ról szóló részben írjuk. Az alapértelmezett mód a data=ordered, amely az adatok és metaadatok integritását egyaránt biztosítja, de naplózást csak a metaadatokhoz használ.

Ext2

Az Ext2 eredete a Linux történetének első napjaira nyúlik vissza. Az eredeti Extended File System 1992 áprilisában készült el és lett beépítve a Linux 0.96c-be. Az Extended File System számos módosításon ment keresztül és Ext2 néven évekig a legnépszerűbb Linux-fájlrendszer volt. A rendkívül rövid helyreállítási idejű naplózó fájlrendszerek megszületése miatt azonban az Ext2 mára sokat vesztett fontosságából.

Az Ext2 előnyeinek rövid összefoglalása segít annak megértésében, hogy miért volt számos Linux-felhasználó kedvenc linuxos fájlrendszere (és néhány esetben miért még mindig az).

Megbízhatóság

Igazi „old-timer”-ként az Ext2 számos javításon és komoly tesztelésen ment keresztül. Ez lehet annak az oka, amiért az emberek gyakran sziklaszilárdnak nevezik. Rendszerkimaradás után, amikor a fájlrendszer nem szabályosan lett lecsatolva, az e2fsck elkezdi elemezni a fájlrendszer adatait. A metaadatok konzisztens állapotba kerülnek és a függőben lévő fájlok vagy adatblokkok egy kijelölt könyvtárba íródnak (ennek neve lost+found). A naplózó fájlrendszerrel ellentétben az e2fsck a teljes fájlrendszert végigvizsgálja, nemcsak az utoljára módosított metaadatbiteket. Ez lényegesen tovább tart, mint a naplózó fájlrendszer naplóadatainak ellenőrzése. A fájlrendszer méretétől függően az eljárás fél óráig, vagy még sokkal tovább is tarthat. Éppen ezért magas rendelkezésre állást igénylő kiszolgálóhoz nem ajánlatos Ext2 fájlrendszert választani. Mivel azonban az Ext2 nem tart karban naplót és lényegesen kevesebb memóriát használ, néha gyorsabb a többi fájlrendszernél.

Egyszerű frissíthetőség

Az Ext2 kódja jó alapot biztosít, amelyre építve az Ext3 könnyen az egyik legnépszerűbb következő generációs fájlrendszerré válhat. Megbízhatósága és szilárdsága elegánsan párosul a naplózó fájlrendszer előnyeivel.

Ext3

Az Ext3 Stephen Tweedie fejlesztése. Szemben a többi új generációs fájlrendszerrel, az Ext3 nem vadonatúj tervezési elvekre épül, hanem az Ext2-re. A két fájlrendszer szorosan kapcsolódik egymáshoz. Az Ext3 fájlrendszer egyszerűen ráépíthető egy Ext2 fájlrendszerre. A legfontosabb különbség az Ext2 és Ext3 között, hogy az Ext3 támogatja a naplózást. Röviden összefoglalva, az Ext3-nak három nagy előnye van:

Egyszerű és nagyon megbízható frissítés Ext2-fájlrendszerekről

Mivel az Ext3 az Ext2 kódjára épül, valamint a lemezen lévő formátum és a metaadatok formátuma is egységes, az Ext2-ről Ext3-ra frissítés hihetetlenül egyszerű. A más naplózó fájlrendszerekre – például ReiserFS, JFS vagy XFS fájlrendszerre – való áttéréssel szemben, ami ugyancsak körülményes lehet (a teljes fájlrendszert el kell menteni, majd nulláról létrehozni), az Ext3-ra áttérés néhány perc alatt végrehajtható. Biztonságos is, mivel elképzelhető, hogy a teljes fájlrendszer újbóli létrehozása nem működik hibátlanul. A meglévő, naplózó fájlrendszerre frissítendő Ext2-rendszerek számát figyelembe véve könnyű kitalálni, hogy miért fontos az Ext3 számos rendszeradminisztrátor számára. Az Ext3 Ext2-re visszaállítása a frissítéshez hasonlóan egyszerű. Az Ext3 fájlrendszert egyszerűen le kell választani, majd újra fel kell csatolni Ext2 fájlrendszerként.

Megbízhatóság és teljesítmény

Több más naplózó fájlrendszer „csak metaadatokat” naplóz. Ez azt jelenti, hogy a metaadatok mindig konzisztens állapotban maradnak, de a fájlrendszerben tárolt adatokra ugyanez nem feltétlenül igaz. Az Ext3 a metaadatokra és az adatokra is vigyáz. E „törődés” mértéke szabályozható. Maximális biztonságot az Ext3 data=journal módba állítása nyújt (adatintegritás), de a metaadatok és adatok együttes naplózása a rendszert nagyon lelassíthatja. Viszonylag új megközelítés a data=ordered mód használata, amely az adatok és metaadatok integritását egyaránt biztosítja, de csak a metaadatokat naplózza. A fájlrendszer-illesztőprogram összegyűjti az egy metaadat-frissítéshez tartozó összes adatblokkot. Ezek az adatblokkok a metaadatok frissítése előtt a lemezre íródnak. Ennek eredményeképp a metaadatok és adatok konzisztenciája egyaránt megmarad, a teljesítmény romlása nélkül. A harmadik lehetőség a data=writeback mód, amely módban az adatok a fő fájlrendszerre íródnak, miután a metaadatok a naplóba kerültek. Teljesítmény szempontjából általában ez a legjobb választás. Így azonban előfordulhat, hogy egy rendszerösszeomlást követően a visszaállításkor régi adatok jelennek meg a fájlokban a belső fájlrendszer-integritás fenntartása mellett. Ha a rendszergazda nem állítja át, akkor az Ext3 alapértelmezés szerint data=ordered módban működik.

XFS

Az eredetileg az IRIX operációs rendszerhez tervezett XFS fejlesztését az SGI az 1990-es évek elején kezdte meg. Az XFS mögötti elképzelés egy olyan nagy teljesítményű 64 bites naplózó fájlrendszer létrehozása volt, amely a mai extrém feldolgozási igényeknek is megfelel. Az XFS kiválóan kezeli a nagy fájlokat és jól működik csúcsminőségű hardveren is. Azonban még az XFS-nek is van hátránya. A ReiserFS-hez hasonlóan az XFS is nagy gondot fordít a metaadatok integritására, de az adatok integritására már kevesebbet.

Az XFS fő funkcióinak áttekintéséből kiderül, hogy miért erős versenytársa más naplózó fájlrendszereknek a felsőkategóriás számítástechnikában.

Kiváló méretezhetőség allokációs csoportok használatával

Az XFS fájlrendszer létrehozásakor a fájlrendszer alapjául szolgáló blokkeszköz nyolc vagy több egyenlő méretű lineáris részre van osztva. Ezeket allokációs csoportoknak hívjuk. Minden allokációs csoport maga kezeli a saját inode-jait és szabad lemezterületét. Az allokációs csoportok gyakorlatilag afféle fájlrendszeren belüli fájlrendszerek. Mivel az allokációs csoportok egymástól függetlenek, a kernel egyszerre többet is megcímezhet. Ez a funkció a lelke az XFS jó méretezhetőségének. A független allokációs csoportok alapelve jól illeszkedik a többprocesszoros rendszerek igényeihez is.

Nagy teljesítmény a lemezterület hatékony kezelésével

A szabad területet és inode-okat az allokációs csoportokon belül B+-fák kezelik. A B+-fák használata nagyban hozzájárul az XFS jó teljesítményéhez és méretezhetőségéhez. Az XFS késleltetett lefoglalást használ. A lefoglalást a folyamat két részre osztásával kezeli. A függőben lévő tranzakciót RAM-ban tárolja, és a szükséges mennyiségű terület lefoglalásra kerül. Az XFS továbbra sem dönti el, hogy az adatokat pontosan hol kell tárolni (már ami a fájlrendszer konkrét blokkjait illeti). Ez a döntés az utolsó lehetséges pillanatig késleltetve van. Lehet, hogy lesz olyan nagyon rövid életű, ideiglenes adat, amelyik sosem kerül a lemezre, mivel megszűnik, mire az XFS eldönti, hogy valójában hova is kellene elmenteni. Ezáltal az XFS javítja a teljesítményt és csökkenti a fájlrendszer töredezettségét. Mivel azonban a késleltetett lefoglalás kevesebb írási eseményt eredményez, mint más fájlrendszerek, valószínű, hogy az írás közbeni rendszerösszeomlás utáni adatvesztés súlyosabb.

Előzetes foglalás a fájlrendszer-töredezettség elkerülése érdekében

Az adatok fájlrendszerre írása előtt az XFS fenntartja (előre lefoglalja) a fájlhoz szükséges szabad területet. Ezáltal a fájlrendszer töredezettsége nagyban csökken. A teljesítmény javul, mert a fájl tartalma nem oszlik szét a teljes fájlrendszeren.

Kérdések

1. Milyen műveleteket kell végrehajtani a merevlemezen, hogy hogy állományokat is tárolhassunk rajta?
2. Mi a partíció? Mi a partícionálás?
3. Sorolja fel, milyen partíciókat különböztetünk meg a PC-s rendszerekben?
4. Mi jellemző az elsődleges partícióra?
5. Mi jellemző a kiterjesztett partícióra?
6. Mi jellemző a logikai partícióra?
7. Rajzoljon példát egy partíciós szerkezetre!
8. Mi az MBR? Mit tartalmaz? Rajzolja fel a szerkezetét!
9. Milyen bejegyzéseket tartalmaz az MBR egy partícióról?
10. Milyen lépéseken keresztül indul el az operációs rendszer a gép bekapcsolása után?
11. Mi a GPT?
12. Hogyan tárolják a partíciókat a BSD rendszerek?
13. Mit értünk logikai fájlrendszer alatt?
14. Mi a klaszter? Mire használjuk?
15. Mit tárol a FAT tábla?
16. Milyen információkat tárol a 12 és 16 bites FAT tábla egy bejegyzése?
17. Mit tárol a könyvtárrész?
18. Mik az árva klaszterek?
19. Mik a keresztbecsatolt klaszterek?
20. Mi jellemző a FAT32-es fájlrendszerre?
21. Mit nevezünk töredezettségnek? Mit tehetünk ellene?
22. Mi jellemző az NTFS fájlrendszerre?