Adattárolás és Perifériák

verzió: 0.8

Tuza Zoltán

2011. szept. 30.

Tartalomjegyzék

1. fejezet

Adattárolás és Perifériák

1.1. Bevezetés

A számítógép futása során az egyes számításokból (rész) eredmények keletkeznek, amit a korai számítógép-generációktól kezdve már jelen lévő illékony[1] memória tárolt el. Később született meg az igény arra, hogy ezeket az eredményeket eltároljuk a számítógép két bekapcsolása között, egy program két futtatása között. Szükséges tehát, hogy tároláshoz ne legyen szükségünk elektromos áramra. Erre kezdetben nyomtatót, illetve lyukkártyákat használtak (ez utóbbi annyival volt szerencsésebb, hogy egy lyukkártya-olvasóval könnyen vissza lehet tölteni az információt a memóriába). Később megjelentek a szalagos tárolási módszerek, melyek segítségével nagy mennyiségű adatot tudtunk lineárisan elmenteni (emlékezzünk a magnókazettákra, ahol ha egy számot ki szerettünk volna hagyni, azt a szalag gyors tekeréssel tudtuk csak átlépni). Az áttörést a cserélhető lemezes olvasó jelentette, ahol egy mágnesezhető korongot forgattunk egy mozgatható mágnenes olvasó/író fej előtt, így szemben a mágneses szalaggal, ahol az a olvasó fej fixen volt tartva. Ezzel a fejet különböző a korong különböző részein lévő adatsávok fölé tudtuk helyezni. Ezt a módszert - melyben a tárolón lévő adatok bármelyikét a többi adat érintése/átlépése nélkül érhetjük el - hívjuk véletlen hozzáférésnek (Random Access). Ezen a ponton két irány indult el, az egyik mentén a cserélhető lemezek fejlesztették, még a másik vonalon létrejöttek fix- vagy merevlemezek. Mivel ezek az eszközök nem közvetlenül vannak a számítógép alaplapjába építve, hanem valamilyen csatlakozón keresztül kapcsolódnak hozzá, így adattároló perifériák gyűjtőnévvel hivatkozunk rájuk. Például az egeret és a monitort is perifériának tekintjük, az egyiket adatbeviteli perfiériának, még a másikat adatmegjelenítő perifériának hívjuk. Három legfontosabb fontos jellemzője egy adattároló eszköznek a következőek:

- Hozzáférési idő (seek time): az az idő ami az adat megcímzése és az adat kiolvasása között eltelik.

[1] tápfesztültség megszünésével a bennetárolt információ elveszik

- Adatátviteli sebesség: egy időegység alatt hány byte információt tudunk az eszközre írni, vagy arról olvasni. Ez a jellemző alapvetően két elemből tevődik össze: az eszköz hozzáférési ideje és a csatolófelület átviteli sebessége (például: SATA 3.0 szabvány: max. 6 Gbit/s, USB 3.0 szabvány: max. 5 Gbit/s).

- Tárolókapacitás: hány bit információt tudunk eltárolni rajta. Fontos tisztázni, hogy ez egy bruttó érték, mivel nem a "nyers" winchesztert használjuk a nettó érték ettől eltérő lehet hiszen a különböző logikai fájlrendszerek más-más módon osztják fel a merevlemezt - Lásd a fájlrendszerek részt.

További fontos jellemző még, az adattárolás élettartalma, azaz meddig képes egy eszköz a ráírt információt megőrizni. A mágneses elven működő eszközök általában előbb szenvednek mechanikai hibából kifolyó adatvesztést, mint hogy a mágnesen elven tárolt adat elveszne. További, külső tényezők miatt is sérülhet az adatintegritás, például mechanikai behatás vagy hőhatás[2]. Míg a Compact Diskek (CD) esetében például a felület elgombásodása jelent veszélyt az információra nézve. Az Információ- és kódelmélet c. tárgyban részletesen tárgyalásra kerülnek az információ tömörítéséhez illetve az adatvesztéssel szembeni részleges rezisztenciához szükséges módszerek.

1.2. Adattároló perifériák

1.2.1. Merevlemez (Hard Disk Drive - HDD) felépítése

Ahogyan a cserélhető lemezes adattárolást, úgy a "fix" vagy "merevlemezes" adattárolást is az IBM mérnökei találták az 1950-es években[3]. Maga az adattárolás elve az elmúlt ötven évben nem sokat változott, egyedül a megbízhatóság és a tárolható adatmennyiség mértéke nőtt drasztikusan. A légmentesen vagy porszűrővel lezárt ház a következő részeket tartalmazza:

- mindkét oldalán mágnes anyagot tartalmazó korongok

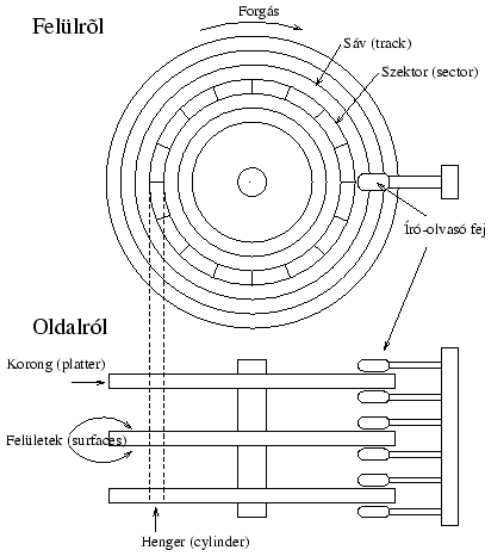
- olvasó/író fejek, amelyek a mágnes felület felett pár mikrométerrel - légpárnán siklanak.

- vezérlő elektronika, amely pozicionálja a fejeket, olvasás esetén elvégzi az analóg mágneses mérés digitalizálását, illetve a csatoló felületnek megfelelő adatátviteli protokollt megvalósítja.

[2]Minden mágneses anyagnak létezik egy úgynevezett Curie-pontja, ezen hőmérséklet felett az anyag elveszti mágneses tulajdonságát

[3]Az első merevlemezt IBM 350 RAMAC néven forgalmazták és 5 megabyte tárolókapacítással rendelkezett, ezt ötven darab 24"-os lemezzel érték el

- forgatómotor, jellemzően 5400 illetve 7200 forulat/perc sebességgel forgatja a lemezeket, tehát az olvasó fejek - átmérőtől függően - kb. 270 km/h-val száguld a lemezek felett.



1.1. ábra. A merevlemez vázlatos képe

A lemezeket a különböző kerületek mentén sávokra osztják, a sávokat pedig szektorokra, ez a legkisebb címezhető egység egy merevlemezen ezek mérete régebben 512 byte volt, jelenleg elérhető 4096 byte is. Mivel az olvasófejek együtt mozognak, ezért a különböző lemezeken azonos sávon állnak minden idő pillanatban, ezeket a sávokat együttesen cilindernek nevezzük. Látható tehát, hogy az összetartozó adatokat szomszédos szektorokra ill. azonos cilinderekre érdemes írni. Fontos fogalom még a klaszter, ami az azonos sávon egymásután elhelyezkedő szektorok gyűjtő neve. Lásd a 1.1. ábrán. A merevlemez hatékony felosztása és az adatok tárolása a merevlemezre telepített fájlrendszer feladata, melyet részletesen tárgyalunk.

1.2.2. Compact Disk (CD) ROM felépítése

A cserélhető lemezes fejlődési vonalat az utóbbi évtizedig a mágneses tárolási elven működő eszközök határozták meg, de ezek tárolókapacitása nem nőtt és/vagy hozzáférési ideje nem csökkent olyan mértékben, mint a Compact Diskeké, ezért a továbbiakban nem is foglalkozunk velük. A CD-s adattárolás egy - a merevlemezes tárolástól eltérően - egy alapvetően optikai elven működő tárolási módszer. Egy lézerdióda által kibocsátott koherens fénysugár tapogatja le a CD lemez felületét, melyen apró gödrök és púpok váltják egymást, melyekről másként verődik vissza a lézersugár - ezzel reprezentálva a bitek értékeit. Gyártás során a CD lemezen - mint a merevlemeznél - sávokat és

szektorokat hoznak létre, amelyben a biteket a vájatok reprezenetálják. Pl.: ha van mélyedés akkor az logikai egyes jelent, ha nincs akkor logikai nullát. Amikor az olvasó fej mindig azonos szögből megvilágítja a felületet, akkor a vájatba beeső lézer fény máshova verődik vissza, mint az lézer fény, mint ami nem esett bele a vájatba. Mivel a lemez fixen síkban forog és az olvasófej is meghatározott szögben világítja meg a felszínt, ezért a várható visszaverődési helyekre fényérzékeny szenzorokat helyeznek el. Értelemszerűen tehát a logikai egyes és nullás értékek más-más szenzorból váltanak ki jelet. Ebből a felépítésből látható, hogy minél fókuszáltabb a lézersugár, illetve minél kisebbek a vájatok a lemez felületén, annál nagyobb az elérhető adatsűrűség (természetesen további fontos paraméterek is vannak - anyagtechnikai jellemzők, a lézer hullámhossza, valamint több adattároló réteggel rendelkező lemezek is léteznek). CD ROM elnevezésben a ROM (Read Only Memory/Media) rövidítés arra utal, hogy ezeket az eszközöket, csak egyszer lehet ír, utána már csak olvashatóak, léteznek újra írható CD lemezek is, ahol egy az olvasástól eltéri tulajdonságú lézersugárral visszállítják az eredeti felületet - természetesen ebben az esetben nem vájatokkal dolgozunk, hanem a felülnek változtatjuk meg a visszaverődési tulajdonságait az írás során.

1.2.3. Pendrive, Flashdrive

Mivel ezen eszközök működésének megértéséhez komoly elektronikai háttér tudásra van szükség, ezért a technikai részleteket nem tárgyaljuk. A működési elvről elegendő annyit megjegyeznünk, hogy ezek a tárolók olyan különleges áramkörök, amelyekben lévő tranzisztorok tápfeszültség jelenléte nélkül is képesek megtartani azt az állapotot, amit tápfeszültség jelenlétében megadtunk nekik, ezzel reprezentálva a logikai 1-0 értékeket. Fontos még megjegyezni, hogy ebből a technikai megvalósításból kifolyóan az ilyen típusú eszközök írási és olvasási sebessége jelentősen eltér egymástól. Az érdeklődők a következő kulcs szavak mentén tudnak további információhoz jutni: Flash memory, Floating gate, EEPROM.

1.3. Adatátviteli technológiák

A programok futtatásához az adattároló perifériákról az adatokat a memóriába kell átmásolni, ezért a megbízható, gyors adat továbbítás kulcs fontosságú.

- Paralell ATA (PATA) / IDE

egy kábelre két eszköz master, slave
- Serial ATA (SATA)
- I/O vezérlő, DMA

1.4. Adattároló eszközök felosztása, fájlrendszerek

1.4.1. Partíciók

Lehetőségünk van arra, hogy a merevlemez lemezén lévő területeket felosszuk és különböző méretű, de egybe tartozó területeket egységbe foglaljunk. Egy ilyen területet partícióknak hívunk. Minden partíció külön kezelhető a többitől, törölhető, formázható, másolható valamint saját fájlrendszerrel rendelkezik. Ezért fizikailag egy lemezen tárolhatunk különböző operációs rendszereket, anélkül, hogy zavarnák egymást.

A partíciók MS Windows alatt betűvel címkézve jelennek meg, pl. C, D. - fontos megjegyezni, hogy ha fizikailag másik lemezen van egy partíció az a meghajtó betű jeléből nem kideríthető. (pl. elképzelhető, hogy a C meghajtó egy partíció, amely a teljes merev merevlemezt elfoglalja, míg a D, illetve E meghajtók fizikailag egy lemezen helyezkednek, de valamilyen arányban megosztják a lemez területét). GNU/Linux alatt már tisztább a helyzet, a *dev* könyvtár tartalmazza a számítógéphez csatolt perifériák eszközfájljait, így a merevlemezekét is. A IDE csatolóval rendelkező lemezeket HDA, HDB, HDC, HDD névvel találjuk a könyvtárban, míg a SATA csatolóval renelezőeket SDA, SDB, SDC, stb. Általában két IDE csatlakozó van egy alaplapon, amelyre két-két eszközt lehet csatlakoztatni, ezért a HDA az elsőleges IDE csatlakozó master eszköze, a HDB ugyanezen kábelen lévő slave eszköz. A HDC, HDD a másodlagos IDE csatlakozóra felfűzőtt eszközöket jelzik. Ha a HDA lemez partíciókat tartalmaz, akkor az elsőleges partíció HDA0 néven fog szerepelni a *dev* könyvtárban, míg a második partíció HDA1 néven. SATA eszközök esetén az SD után következő A,B,C,D betűjelek és a 1,2,3,4 számok ugyanezt jelentik.

Természetesen az operáció rendszert informálni kell arról, hogy milyen partíciók léteznek az adott lemezen, ezeket az információkat tartalmazza a Master Boot Record (MBR). Ezen bejegyzés tartalmazza a partíciók méretét, kezdő és vég értékét valamint azt, hogy melyik partíció tartalmaz operációs rendszer elindításához szükséges adatokat - ezt/ezeket a partíciókat hívjuk bootolható partíciónak. A MBR-t az 1980-as években találták ki. Továbbfejlesztése a GUID Partion Table (GPT), amely számos kiterjesztést tartalmaz MBR-hez képest, például az MBR esetében a legnagyobb partíció mérete maximum 2.19 TByte lehet, míg az GPT esetén ez 9.4 ZetaByte (Zeta = $10^2$1). Tipikus háztartási méretekben ezek a számok megmosolytatónak tűnhetnek, de vegyük figyelembe, hogy ezeket a technológiákat nagyvállalai rendszerekben is használják, ahol egy adatbázis gond nélkül elérhet 2TB méretet.

1.4.2. Fájlrendszerek

A fájlrendszer feladata, hogy az eltárolandó fájlokat és könyvtárakat a winchester egy partícióján a megfelelő helyen elhelyezze, garantálja annak visszaolvashatóságát, valamint a változásokat adminisztrálja. Tehát a fájlrendszer

funkciója kettős: egyrészt tárolja egy adott partíción lévő adatsávok (fájlok) helyét, másrészt kezeli az ezekhez kapcsolódó metaadatokat[4]. Minden, a fájlrendszerben tárolt adathoz (egy adott fájl fizikai elhelyezkedése a lemezen) tartozik egy metaadat bejegyzés is, ez tartalmazza a fájl vagy könyvtár nevét, létrehozás, módosítás dátumát, tulajdonos adatait, valamint a hozzáférési jogosultságokat. Fontos tudni, hogy a könyvtárak fizikailag nem jelennek meg a lemezen, azok csak a fájlrendszer adatbázisában lévő bejegyzésként vannak tárolva (tehát amennyiben egy lemezről elveszítjük a fájlrendszert leíró adatbázist, úgy a nyers adatok visszanyerhetőek, de: 1) nem fogjuk tudni, hogy melyik fájl hol kezdődött és hol van vége 2) nem fogjuk tudni rekonstruálni a könyvtárrendszert.) Látható, hogy a fájlrendszer helyes működése létfontosságú a tárolt adatok használhatóságának szempontjából, éppen ezért a fájlrendszer adatbázisa a lemezen általában több példányban, sokszorosítva kerül eltárolásra, csökkentve a megsérülés valószínűségét.

Mivel a számítógépeknek számos különböző feladatra használhatóak (családi személyi számítógép, bankszámlakezelő rendszer, egy tőzsdei kereskedőrendszer vagy egy milliós forgalmú webkiszolgáló), ezért az adatok tároláskor is különböző igények léphetnek fel (mind teljesítmény, mind biztonság tekintetében). Ezen igényekre kielégítésére rengeteg fájlrendszer megvalósítás létezik ebből ebben a jegyzetben részletesen a FAT, illetve ext2 fájlrendszerrel fogunk megismerkedni.

Fájlrendszereket általában az operációs rendszerrel együtt szokták szállítani, például MS Windows operációs rendszerhez két fájlrendszer érhető el: a FAT, illetve az NTFS. Míg Linux alatt a legelterjetebb az ext3 fájl rendszer, de számos egyedi igényéket kielégítő megvalósítás érhető el (xfs, jfs, Google file system, stb). Továbbá léteznek hálózati fájlrendszerek is, amelyek az operációs rendszer szempontjából átlagos partíciónak tűnnek, de fizikailag az adatok nem az adat számítógép merevlemezén tárolódnak. Például az egyetemen lévő tárhelyeinket (turdus, users) is felcsatolhatjuk meghajtóként az általunk használt operációs rendszerben. Érdekesség, hogy linux alatt lehetőség van a Google által üzemeltetet gmail levelező szolgáltatáshoz tartozó postafiók tárhelyét meghajtóként felcsatolni, így arra bármilyen adatot menthetünk a szabad tárhely függvényében.

A fájlrendszerek egy fontos tulajdonsága a legkisebb foglalási egység, azaz ha létrehozunk egy üres fájlt, akkor fizikailag mekkora tárterület foglalódik le ennek a fájlnak a winchesteren.

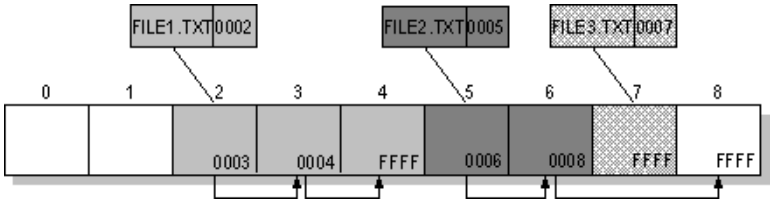
Fontos megjegyezni, hogy a fájlrendszerek fájl- és könyvtárelhelyezési implementációja nem azonos az általunk megszokott könyvtárstruktúrával! Tehát a felhasználók számára látható könyvtárstruktúra mögött a fizikai tárolási módja ettől teljesen eltérő, ahogy ezt látni fogjuk.

[4] adatokat leíró adatok

1.4.3. Fájlrendszer implementációk

FAT fájlrendszer

A File Allocation Table (FAT) fájlrendszer a Microsoft fejlesztette és a Windows XP operációs rendszerig, régen ez volt a Windows operációs rendszerek által kizárólagosan használt fájlrendszer. A FAT fájlrendszer klasztereket tart számon, és a különböző FAT verzió főként abban különbözőnek, hogy hány biten tárolják a klaszterek sorszámait (FAT12, FAT16, FAT32). Ez bitszám határozza meg, hogy összesen mekkora lehet egy FAT partíció mérete. Az operációs rendszert is tartalmazó FAT partíciók tartalmaznak boot sectort is, ezt a szektor kerül beolvasásra memóriába az operációs rendszer bootolásának első lépéseként. Az alábbiakban részletesen megnézzük, hogyan tárolja a fájlokat a FAT fájlrendszer. A 1.2 ábrán látható a partíció egy darabja: a tárolt fájlok, valamint a hozzájuk tartozó allokációs tábla-beli bejegyzések. Az allokációs tábla - többek között - tartalmazza a fájl nevét és annak a klaszternek a számát, ahol a fájl kezdődik. Minden klaszter végén található egy cím amely a fájl többi darabját tároló klaszterre mutat vagy egy 0xFFFF jelzés, ami azt jelenti, hogy ez az utolsó klaszter, amiben a fájl részlete volt eltárolva. Fontos észrevennünk, hogy a rendszer nem követeli meg, hogy egy nagyobb méretű fájl egymás utáni klaszterek sorozataként legyen a lemezen: az operációs rendszer utasításaitól függően akár rengeteg, a lemez különböző pontjain elhelyezkedő klaszterbe is kerülhet a fájl egy-egy darabja. A FAT fájl rendszerekben a legnagyobb tárolható fájl 4GB.

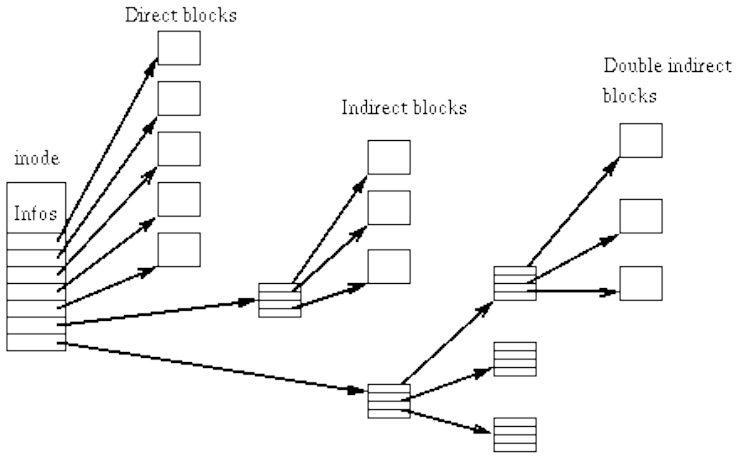


1.2. ábra. Egy FAT partíció darabja. Forrás: http://www.ntfs.com/images/recover-FAT-structure.gif

Ext2 fájlrendszer

Az ext2 fájlrendszer alapegysége a blokk, amelynek mérete tipikusan 1-8 Kb-ig terjed (ez a fájlrendszer létrehozásakor beállítható érték, de később nem változtatható és minden blokk ekkora méretű lesz). Így ha létrehozunk egy fájlt amiben elhelyezünk 2 karaktert az minimum 1 Kb-ot (vagy épp 8Kb-ot, ha akkora a blokkméret) fog elfoglalni a lemezen. Létezik egy szuperblokk, amely partíció elején helyezkedik el és többek között az operációs rendszer bootolásához szükséges információkat tartalmazza. A blokkokat csoportokban tárolják, ezzel is csökkentve a töredezettség mértékét.

A ext2 fájlrendszerben minden fájlt és könyvtárat egy úgynevezett inode ír le. Az inode tartalmazza a fájlal vagy könyvtárral kapcsolatos adminisztratív információkat: fájlnevét, létrehozás, módosítás dátumát, tulajdonost, jogosultságokat, stb. Az inode többi része 12-15 linket (blokk címeket) tartalmaz, amely egy csoportot címez meg, ezek a direkt blokkok (lásd a 1.3. ábrán). Amennyiben a fájl mérete meghaladja a direkt blokkokban tárolható adatmennyiséget, akkor az utolsó link helyére nem egy direkt blokk címet helyezünk az inode-ban, hanem egy másik csoportleírót, ami további blokkokra vagy csoportleírókra mutat. Ezzel a módszerrel a legnagyobb tárolható fájl mérete 1KB-os blokk méretnél 16 GB, míg 8 KB blokkméret esetén 2 TB.



1.3. ábra. Az ext2 inode felépíése Forrás: http://upload.wikimedia.org/wikipedia/commons/a/a2/Ext2-inode.gif

A fájlok és könyvtárak mellett létezik egy másik típusú inode bejegyzéstípus is, ez a link. A link nem más, mint egy bejegyzés a fájlrendszerben, amely egy másik fájlrendszer-beli bejegyzésre hivatkozik. Önmagában tehát nem tárol adatot, hanem az őt megnyitó programot továbbirányítja az általa mutatott fájlra (ennek nagyságrendekkel butább utánzata az MS Windows-beli parancsikon). Két típusát különböztetjük meg, az egyik a szoft link a másik a hard link. A hard link esetében az inode-ban található linkek nem blokokkra mutatnak, hanem egy másik inode bejegyzésre. Ezzel szemben a szoft link egy speciális fájl, amely annak a fájlnak az elérési útját tartalmazza, amire mutat.

Ebből következően a hard linknél a mutatott fájl vagy könyvtár addig nem törölhető, amíg létezik rá mutató link (ezt a link számlálóból tudja - lásd feladatok).

A szoft linknél a link az elérési utat tárolja, így a mutatott fájl vagy könyvtár nem tudja, hogy létezik olyan hivatkozás, amely reá mutat. Éppen

ezért ha azt letöröljük a hivatkozott fájlt, a link "célpont" nélkül marad, és "törött" link jön létre. Másik fontos különbség, hogy hard linket csak partíción belül lehet létrehozni, mivel az inode-ra mutat, aminek a számozása partíciónként újrakezdődik. Ezzel szemben a szoft link elérési utat tárol (ahogy azt már említettük egy partíció a könyvtárstruktúra tetszőleges pontjára becsatolható), így a szoft linket nyugodtan mutathat másik partíción elhelyezkedő fájlra.

SWAP fájlrendszer

Egy adott pillanatban nem minden programot használunk, amit elindítottunk a számítógépünkön, illetve az adott programnak sem használjuk minden részét. Ebből kifolyólag a nem aktív programokat, valamint programrészeket az operációs rendszer nem a viszonylag szűkös memóriában tartja, hanem a winchesteren, az úgynevezett swap (csere) területen. A tárolás olyan formátumban történik, hogy a kiírt programrészleteket ill. adatokat szükség esetén külön keresés-konvertálás nélkül a memóriába tudja visszatölteni. (Például: amikor a tálcára leteszünk egy programot és sokáig nem foglalkozunk vele, majd később elővesszük azt tapasztaljuk, hogy elég lassan reagál a kéréseink, de a winchester nagy tempóban dolgozik, ekkor kerül vissza a swap (csere) területről a memóriába az adott programhoz tartozó adatok és program részletek). MS Windows alatt a fájlrendszerben egy reguláris fájlként jelenik meg a swap terület, amit Pagefile-nak hív a rendszer. Ez komoly hátrány, hiszen mint reguláris fájl, ez is ki van téve a töredezettségnek, hasonlóan a többi, fájlrendszer-beli fájlhoz. (Elérhetők olyan kis programok, amik a windows indulásakor töredezettségmentesiti, ezzel jelentős sebesség növekedés érhető el). GNU/Linux rendszereknél a swap tárterület egy linux-swap típusú fájlrendszerrel rendelkező külön partíció. Ez a megoldás természetesen nem szenved az előbb említett, a fájlrendszerek töredezettségéből fakadó hátránytól.

Cserélhető eszközök fájlrendszerei

Természetesen a cserélhető lemezeken is található fájlrendszer. CD és DVD esetében ISO9660 fájlrendszert szokás használni, még

Hálózati fájl rendszerek

- sshfs

Naplózó fájlrendszerek

Journaling, ext3, atomi műveletek fogalma

1.5. Könyvtárstruktúra és a fájlrendszer adminisztrációjának manipulációja

Linux alatt a BASH shell segítségével lehetőségünk van a parancssori utasítások segítségével fájlok és könyvtárak létrehozására, módosítására, valamint törlésére, tehát a könyvtárstruktúra módosítására. Továbbá lehetőségünk van a fájlrendszer adminszitrációs információk megjelenítéséra, megfelelő jogosultság esetén módosítására.

1.5.1. Könyvtárstruktúrával kapcsolatos parancsok

- könyvtárváltás: cd (change directory), aktuális pozíciónk a könyvtárfában: pwd (path to working directory), könyvtárfa megjelenítése: tree parancs. könyvtár tartalmának listázása: ls. Könyvtárakat az mkdir NÉV (make directory) parancs segítségével tudunk létrehozni.

- fájlok tartalmának megjelenítése, összefűzése: cat, szerkesztésre rengetek lehetőségünk van, például: mcedit, vim. Üres fájlokat is létrehozhatunk a touch parancs segítségével - létrejön az inode tábla, de a blokkok üresen maradnak.

- fájlokat másolni a copy, azaz cp MIT HOVA paranccsal lehet, áthelyezni a mv (move) paranccsal lehet. Könyvtárakat, és tartalmukat másolni a cp -r MIT HOVA utasítással tudunk (-r: rekurzívan, az alkönyvtárakat is másolja)

1.5.2. A fájlrendszerrel kapcsolatos parancsok

- Az egyes bejegyzések (fájlok és könyvtárak) hozzáférési jogosultságát a chmod paranccsal állíthatjuk be, a tulajdonos váltása chown parancsok segítségével történik, valamint felhasználó csoport váltása a chgrp parancs segítségével lehetséges.

- Az fsck parancs segítségével lehet ellenőrizni, hogy a winchester tartalma megegyezzik-e az adminsztrációs fájlok által leírt állapottal, azaz a fájlrendszer koherens állapotban van-e. Ilyen például akkor fordulhat elő, amikor hirtelen kikapcsol a számítógép (pl. áramszünet esetén) és valamilyen lemezművelet félbeszakad. Szintén problematikus eset a fájlrendszer koherenciájának szempontjából, ha akkor távolítunk el egy cserélhető eszközt, amikor még nem fejeződött be a lemezre írás művelet.

- Fájlrendszer egy üres partícióra a mkfs parancs segítségvel tudunk létrehozni, a parancs lefutása létrehozza az összes adminisztrációs állományt, ami szükséges a fájlrendszer menedzseléséhez. Hasonlóan, ha egy fájlrendszerrel rendelkező partíciót leformázunk, akkor a formázás létrehozza az üres

adminisztrációs fájlokat (fontos, hogy ezzel még az előző fájlrendszerben tárolt adatok megmaradnak, csak nem tartozik hozzájuk adminisztrációs állomány).

- Az érvényes, hibamentes fájlrendszereket tartalmazó partíciókat használat előtt fel kell csatolnunk a könyvtárstruktúrába. Általánban az mnt könyvtárban van egy - partícióhoz tartozó - üres könyvtár, ahová a mount paranccsal tudjuk becsatolni a partíciót.

1.6. Feladatok

- nézzük meg mit csinál a df parancs, keressük meg a man oldalán, hogy mit csinál a -T kapcsoló és futtatssuk a df -T parancsot.
- nézzük meg a stat parancs man oldalát, próbáljuk ki a következőekre: sima fájl, könyvtár, eszközfájl, szotflink, hardlink.
- szintén nézzük meg az ls -i parancsot, keressünk egy fájl a könyvtárból és nézzük meg, hogy az ls -i parancsban megadott inode szám egyezik-e a stat parancs kimenetével.
- hozzunk létre szoft és hard linkeket, fájlra, könyvtárra, figyelünk a linkcounter értékének változására. Töröljük azt a fájlt amire a link mutat mit tapasztalunk szoft illetve hard link esetén?
- nézzük meg a dumpe2fs parancsot és futtassuk az egyik partícióra. A grep parancs segítsgével (grep -i superblock) nézzük meg hány példányban tárolódik a lemezen a szuperblokk.

1.7. Ellenőrző kérdések