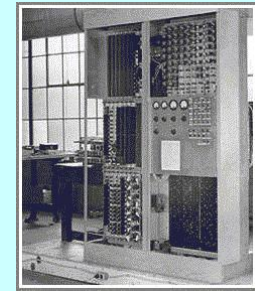


Operációs Rendszerek



- Első alkalmazások: katonai lőpálya elemek, **analóg gépek**
- Neumann János, 1945-46: általános célú számítógép elve, *digitális tárolja az adatokat ÉS a programkódot is a memóriában, mely kódot szekvenciálisan értelmezi (kontrol eszköz) és hajtja végre (aritmetikai és logikai végrehajtó egység).*
- EDVAC, 1952, első Neumann féle számítógép (0.00083 MIPS) (1943-ban már működik a Colossus, az Enigma rendszer feltörője)
- Sokáig a gépek nem Neumann félék, és manapság is a célgépek egy része más elven dolgozik
- Általános gép: „lassú”, célgép: „gyors” (egyres videókártya processzorok számítási teljesítménye meghaladja a főprocesszorét)
- Csöves gépek: „debuggolás” szó eredete
- IBM elnöke szerint a világnak legfeljebb 3-4 számítógépre lesz szüksége
- Neumann: *a tudomány a jövőben inkább a szabályozás és vezérlés, programozás, adatfeldolgozás, kommunikáció, szervezés és rendszerek problémáival törődik majd*



Bevezetés

- A modern, általános célú (személyi, illetve szerver) számítógépek rendelkeznek **operációs rendszerrel**, mely képes futtatni az **alkalmazásokat**. (az első számítógépeknek nem volt operációs rendszere, legfeljebb könyvtáraik). Manapság már operációs rendszerek futnak a számítógépeken, PDAkon, mobiltelefonokon, és a legváratlanabb helyeken (pl. az Airbus szórakoztató rendszere Linux alatt fut).
- A legelterjedtebb rendszerek a Microsoft Windows, a Unix (Linux), a Mac OS és a VMS.
- Az op. rendszer legalsó rétege a **kernel**. Igazából a kernel maga az operációs rendszer, de néhány rendszerben összekeveredik más komponensekkel. Az operációs rendszer feladatai:
 - Az alkalmazások, rendszerfeladatok időzítése
 - Erőforrás menedzsment (memória-kezelés, adathordozó elérés, stb.)
 - Jogosultságok kezelése
- Manapság az operációs rendszerek a grafikus felhasználói felület nevű (GUI: graphical user interface) rendszerszoftverrel együtt jönnek (a Windows és a MacOS beleintegrálta ezeket az operációs rendszerbe).
- Igen gyakran a segédprogramok (utility software) is beépülnek az operációs rendszerbe.
- Végül egyes alkalmazások is az operációs rendszerrel kerülhetnek terjesztésre (Windows Internet Explorer)

Alkalmazások

Az alkalmazások a számítógépes szoftverek egy alosztálya, melyek felhasználják a számítógépet arra, hogy megvalósítsanak a felhasználó által kívánt feladatokat.

Az operációs rendszer és az alkalmazás közti határvonal nem éles, és sokszor vitára ad alkalmat:

USA – MicroSoft per az Internet Explorer helyzetéről,

GNU/Linux elnevezés: Linux op. Rendszer – vagy Linux kernel?

Tipikus alkalmazások:

Szövegszerkesztők

Táblázatkezelők

Média lejátszók

Alkalmazáscsomagok: több, összefűzött alkalmazás együttese, általában közös indítófelülettel, és kölcsönhatással. Pl. egy táblázatkezelő beépülhet egy szövegszerkesztőbe.

MicroSoft Office

OpenOffice.org

Felhasználói programok:

Pl. táblázatkezelő sablonok, szövegszerkesztő makrók, tudományos szimuláció, grafika, animáció

Embedded system (beépített rendszer)

Amikor az alkalmazás annyira beépül az operációs rendszerbe, hogy már megkülönböztethetetlen attól (pl. mikrohullámú sütő).

Kernel

A számítógép indulásakor az első, a memóriába töltődő program, mely a többi rendszer vagy felhasználói program számára hozzáférést biztosít a közös alapszolgáltatásokhoz (erőforrásokhoz).

Folyamat (processz) menedzsment

A számítógépen minden feladat (rendszer, vagy alkalmazás) processzben fut

A Neumann struktúrában CPUként egy processz futhat. A régebbi operációs rendszerek (pl. MS-DOS) meg is elégedtek ezzel.

Modern gépeken multitasking van: az operációs rendszer szimulálja, mintha számos processz futna egyszerre egy CPU-n. A processz menedzsment ennek a folyamatnak a megvalósítása: az operációs rendszer gyorsan vált a processzek között, és mindig a soron következőt juttatja a CPU-ra. Ahogy növekszik a processzek száma, egyre kisebb időszeltek jutnak az egyes processzeknek (és tipikusan ilyenkor hagy ki a hang, vagy ugrál az egér, és az operációs rendszernek már csak arra van ideje, hogy saját magát menedzselje ...)

A processz menedzsment feladata az egyes folyamatokra eső időszeltek megállapítása és szétosztása.

A legtöbb operációs rendszer lehetővé teszi a processzek prioritásának kezelését.

Az interaktív operációs rendszereket továbbá prioritást adnak azoknak a folyamatoknak, melyekkel a felhasználó éppen dolgozik.

Multitasking:

Az operációs rendszerek különböző ütemezési stratégiákat alkalmazhatnak:

Multiprogramming: a futó folyamat vagy önként adja át a vezérlést (pl. olyankor, amikor várni kell egy másik folyamatra: beolvasás adathordozóról), vagy az operációs rendszer ütemezője ki nem dobja. Ez az eljárás maximalizálja a CPU használatot (feleslegesen senkitől nem vesszük el a vezérlést).

Időosztásos rendszer (time-sharing): a futó folyamat vagy önként, vagy pl. hardver interrupt hatására átadja a CPU-t a következő folyamatnak. Az eljárás a folyamatok párhuzamos futásának érzetét kelti. Manapság ezt nevezzük **multitasking**nak.

Valós-idejű rendszer (real-time): az operációs rendszer garantálja, hogy bizonyos várakozó folyamatok garantáltan megkapják a CPUt, amikor a külső esemény, amire várnak bekövetkezik. Fő alkalmazási területük pl. az ipari robotok.

Fejlődés:

- Kezdetben a CPU idő nagyon drága, a perifériák nagyon lassúak. A 60-as évektől ezért elterjed a *multiprogramming*. Hátránya, hogy nem lehet tudni, mit csinál a többi folyamat, teljesen kiszámíthatatlan processzünk várható futási ideje. Háttérmódú (batch) futtatáshoz használták.
- *Cooperative multitasking* a következő lépcsőfok: a folyamatok önként adják tovább a CPUt egymásnak. Nem használták nagy rendszereken, a Microsoft Windows 95, valamint Mac OS X előtti rendszerek használják. Elvben az MS-DOS is lehetővé tette ezt a TSR (Terminate and Stay Resident rendszerhívással, de nem használták). Hátránya, hogy egy rosszul tervezett program lefagyasztja a gépet.

- A *preemptive multitasking* volt hivatott az előbbi problémát: a processzeket két csoportra osztja: az input/outputra várókra és a CPU-ra várókra. Az utóbbi csoport (a prioritások) alapján jogosult egy időszelre a CPU-n, és nem kell megvárnia, míg az előző folyamat önként átadja azt. A külső események bekövetkezése megszakítást vált ki, mely lehetővé teszi a blokkolt folyamatok szóhoz jutását.
- A *valós idejű* rendszer a hierarchikus megszakításokra épül, melynek során a feltehetőleg korrelálatlan külső aktivitásokat egyetlen processzor kezeli le. Az egyes megszakításokhoz prioritásokat lehet rendelni fontosságuk sorrendjében.
- A *multithreadinget* a javuló multitasking hatására felvetődő igény hozta létre, hogy a párhuzamosan futó folyamatok kommunikáljanak egymással (pl. az egyik eljárás gyűjti az adatokat, a második feldolgozza, a harmadik kiírja azokat egy adathordozóra). A *thread* azon folyamatok gyűjteménye, melyek közös memóriaterületen futnak, így könnyen tudnak adatokat cserélni. A folyamatok közötti váltás nem igényel memóriatartalom váltást.
- A multithreading során a folyamatok preemptív módon váltakoznak a CPU-n. Egyes operációs rendszerek lehetővé teszik az egy threadbe tartozó folyamatok között a kooperatív multitaskingot (melyet bárki szoftveresen is megvalósíthat), ilyenkor a kooperatív multitaskinggal működő folyamatcsoportot *fiber*nek hívjuk. Ez a megoldás többprocesszoros rendszereken azonban általában káros.

Memória menedzsment

- Parkinson törvénye szerint “A programok arra törekszenek, hogy kitöltsék a rendelkezésükre álló memóriát”.
- A programozók mindig azt gondolják, hogy a memória végtelen nagy és végtelen gyors.
- A mai számítógépek memóriája többszintű:
 - Register
 - Cache
 - (Gyorsmemória, pl. CRAY solid-state tárolóeszköz), Memória
 - Merevlemez
 - Szalag, DVD, stb.
- Az operációs rendszer memóriakezelő modulja koordinálja a különböző memóriák elérhetőségét, foglal bennük, illetve szabadít fel belőlük területeket, valamint elvégzi a swap-olást a fő és a háttér memória között (paging).
- *Virtuális memória menedzsment*: a háttértárolók igénybevételével a számítógép látszólagos memóriája jóval a fizikai memória fölé növelhető. Ennek ára a sebesség jelentős csökkenése, mely akár a számítógép kritikus lelassulásához is vezethet.
- *Virtuális címkezelés (memóriavédelem)*: az operációs rendszer a megosztott memóriát nem igénylő programokat saját memóriacímterületre teszi. Ezáltal a program által kért memória sohasem a fizikai címre mutat, hanem azt az operációs rendszer leképzi a valódi memóriacímre. Ezáltal elkerülhető, hogy egy program a másik memóriájába írjon, hátránya, hogy olvasni sem tud belőle, azaz két külön program nem tudja megosztani memóriáját. Az operációs rendszer memóriatáblákban tárolja a konverziós adatokat.

- *Memóriavédelem*: Ha több program is van a memóriában, egy (szándékosan vagy véletlenül) rosszul viselkedő program felülírhatná a másik, de akár az operációs rendszerhez tartozó memóriát. A védelem többszintű:
 - Minden programhoz tartozik egy memóriaterület, melyet elérhet. Ha a program ebből kimutat, akkor az operációs rendszer azonnal leállítja, mielőtt még megváltoztatná a védett memóriát.
 - *Privilegiumok*: az alacsony privilégiummal rendelkező programok számára tilosak meghatározott típusú memóriaelérések, illetve bizonyos CPU műveletek. Amennyiben tiltott műveletre kerül sor, a vezérlés azonnal átadódik (trap) egy felügyelő programnak, mely eldönti, hogy mi a teendő.
 - *Virtuális gép*: minden folyamat olyan környezetet kap, melyben a szükséges erőforrások számára látszólag kizárólagosak, és nem keverednek más folyamatok erőforrásaival.
- *Swap*: a fizikai memória látszólagos megnövelésének módja a swap-file. Ez egy lassabb adathordozón (merev lemez) lefoglalt terület, melyben a fizika memóriában tárolt, de pillanatnyilag nem használt tartalmat az operációs rendszer kimentti, és ezáltal felszabadítja a folyamatok számára a gyorsabb memóriát. A multitasking és a swap két teljesen különálló technika, de a legtöbbször együtt használatosak. A multitasking rendszerben az operációs rendszer egy másik folyamatot enged futni, míg az első folyamat arra vár, hogy az általa használt memóriatartalom visszatöltődjön a másodlagos tárolóból (swap file).
- *Többprocesszoros gépek* esetében két technológia létezik:
 - *Standalone* gépek: minden processzornak saját, önálló memóriája van (klaszterek)
 - *Shared memory* rendszerek, amikor a processzorok megosztják a memóriát, így a különböző processzoron futó folyamatok megoszthatják egymással adataikat.

Device driver / Eszközkezelő

A modern gépekben, mivel a különböző perifériák nagyon sok gyártótól jöhetnek, nem várható el az operációs rendszertől, hogy azokat mind kezelni tudja.

Ezért az operációs rendszer nyitva hagyja a lehetőséget, hogy egy egységesített felületen keresztül a különböző eszközök kommunikáljanak vele (vagy a buszon, vagy egyéb kommunikációs alrendszeren keresztül).

Az eszközök gyártóinak biztosítaniuk kell azokat a meghajtó programokat, melyek az operációs rendszer szabványműveleteit átfordítják az adott eszköz parancsaira, és az eszköz által visszaadott adatokat értelmezhetővé teszi az operációs rendszer számára, valamint megvalósítja az aszinkron működésű hardver interrupt-kontrollját. Ezek az eszközkezelő programok.

Az eszközkezelők függenek mind az adott hardver eszköztől, mind az operációs rendszertől.

A fejlesztőknek viszont nem kell érteniük az adott hardverhez, azok specifikumait az operációs rendszer elrejtja előlük. Természetesen lehetőség van a dokumentáció ismeretében az eszközhöz való közvetlen hozzáférésre is.

Továbbra is az operációs rendszer feladata az erőforrások elosztása a folyamatok között, ezeket az eszközkezelőnek nem kell tudnia.

Ez a moduláris felépítés nagyon flexibilissé teszi az operációs rendszereket: a kiadásuk után forgalomba kerülő eszközökkel is képesek együttműködni.

Rendszerszoftver

A rendszerszoftver a számítógép működéséhez szükséges rendszer általános neve. Ennek lehetséges részei:

- a) az operációs rendszer,
- b) adatbázis könyvtárak
- c) grafikus könyvtárak (pl. OpenGL – videókártya meghajtás)

A szoftverek másik nagy családja az alkalmazások, mely közvetlenül a felhasználó által megkívánt tevékenységet hajtják végre.

A rendszerszoftver állandó tárolóban helyezkedik el (merevlemez, ROM, CD, stb. – általánosságban firmware).

Grafikus felhasználói felület

Manapság a legtöbb operációs rendszer grafikus felhasználói felületet (Graphical User Interface – GUI) tartalmaz. Néhány régebbi rendszer szorosan integrálta a GUIt a kernellel (pl. az eredeti Windows és Mac OS implementációk).

A modern operációs rendszerek modulárisak, és a grafikus felület szét van választva a kerneltől (pl. Unix, Windows NT – WNT – technológiára épülő rendszerek, illetve a Mac OS X).

Számos operációs rendszer megengedi a felhasználónak a grafikus felület megválasztását. A Unix alapú rendszerekben az X Window System nyújtotta alapokon üzemel a GNOME és a KDE rendszer (illetve számos egyszerűbb ablakkezelő). A felhasználók különböző GUIkat is futtathatnak egy időben ugyanazon a gépen.

A grafikus felületek időben változhatnak. Pl. a Windows minden új verziója más felülettel jön ki, és az eredeti Mac OS GUI is dramatikusan megváltozott a Mac OS X-ben.

A grafikus felületek teszik lehetővé a felhasználó számára, hogy a lényeges műveleteket egérmozgatással és kattintással hajtsák végre, és ne kelljen az azokat megvalósító parancsokat a konzolon begépelni (pl. Linux konzol, vagy MS-DOS).

Az egér és a grafikus felület is a Xerox Palo Alto-i központjának találmánya (mint az Ethernet kártya, az integrált dokumentumkezelés, és számos más hasznos dolog).

Segédprogram (Utility software)

Számítógépes szoftver, mely segít a számítógépes hardver, operációs rendszer, illetve alkalmazások karbantartására és finomhangolására szolgál, és nagyon korlátozott feladatot hajt végre. A legtöbb operációs rendszer tartalmaz segédprogramokat, pl.

- defragmenter
- profile kezelés (mentések)
- víruskeresők
- bináris vagy hexa szerkesztők
- archiváló parancsok
- tömörítők
- kriptálók

Állományrendszerek

Az operációs rendszerek az állományokat állományrendszereken tárolják. Az állományrendszer független a hordozótól (merevlemez, CD, pen-drive, memória), és leírja az állományok elhelyezkedését a hordozón a könnyebb megtalálás és elérhetőség céljából.

Állományrendszerek létrehozhatók virtuálisan is, virtuális adatok eléréséhez (pl. procfs a folyamatok adatainak megjelenítéséhez, illetve paramétereik változtatásához, vagy hasonló állományrendszer építhető hálózati adatokhoz is).

Az állományrendszer egy absztrakt adattípus halmaz, melyet az adatok tárolására, hierarchikus szervezésére, manipulációjára, elérésére és a köztük levő navigálásra terveztek – ennek következtében sok köze van az adatbázisokhoz.

A legelterjedtebb állományrendszerek az olyan hordozókon találhatók, melyek adott hosszúságú blokkokat (tipikusan 512 bájt) – más néven szektorokat – szolgáltatnak. Az állományrendszer kezelő szotverek felelős azért, hogy a szektorokat állományokba illetve könyvtárakba szervezze, és nyomon kövesse, melyek szektor melyik állományhoz tartozik, illetve, hogy mely szektorok szabadok.

Az állományrendszerek általában tartalmaznak könyvtárakat, melyek minden állománynévhez hozzárendelik az állományt – legtöbbször egy index állományhoz, mely megadja az állomány foglaltsági táblát. Ilyen tábla a FAT az MS-DOS rendszerekben, illetve az inode a Unix rendszereken. A könyvtárakat maguk is tartalmazhatnak egész könyvtárrendszereket. Néhány állományrendszerben az állománynevek strukturáltak, meghatározott szabvánnyal a kiterjesztésre és a verziószámra. Más rendszereken az állománynevek egyszerű karakterláncok, és a metainformáció másutt tárolódik.

A hagyományos állományrendszerek megengedik állományok és könyvtárak

- létrehozását
- mozgatását
- törlését

A Unixban jelennek meg a az utalások, a szülő utalás (..) átnevezése.

A hagyományos állományrendszerek megengedik továbbá állományok

- csonkítását (végéből)
- megnövelését az állomány végéhez való hozzáírással
- helyettesítését (törlés – létrehozás).

Nem engedik meg az állományok

- csonkítását (elejéből)
- növelését az állomány elejéhez való írással
- törlést / betoldást tetszőleges helyen

Ennek eredménye, hogy pl. a folyamatok közötti kommunikációt a Unixban az állományrendszeren kívül kell megoldani (mivel a pipe – cső nem lenne csonkolható az elejéről).

Az állományrendszer feladata jogosultságok számontartása is:

- csoportos jogosultságok (felhasználó, csoport, többiek)
- hozzáférési lista (ACL: Access Control List)

Állományrendszerek típusai:

- Lemez állományrendszerek: az állományok adathordozón való tárolására szolgál
 - Típusai: FAT, NTFS, HFS+, ext2, ext3, ISO 9660, ODS-5, UDF, stb.
 - Egyéb csoportosítás: naplózó, illetve verziókövető (pl. VAX/VMS) állományrendszerek.
- Adatbázis állományrendszerek: a hierarchikus struktúra helyett az állományokat metainformációval látjuk el (típus, szerző, témakör, stb.)
 - Típusai: BFS, WinFS
- Tranzakciós állományrendszerek: főleg a pénzügyi világban használatosak, változás nyomonkövetésre használatosak. Egy változás több állományt is módosíthat egyszerre. Üzemzavar esetén minden tranzakció visszaállítható a státuszával.
- Hálózati állományrendszer: kliensként működik, és hozzáférést biztosít a szerveren levő állományokhoz.
 - Típusai: NFS, SMB, AFS, valamint az FTP és WebDAV kliensek.
- Speciális célú állományrendszerek: minden nem lemez és hálózati alapú állományrendszer. A szoftverek generálják le dinamikusan a tartalmukat.
 - Típusai: folyamatok közötti kommunikáció, ideiglenes tárhelyek.
 - Unixon (állományrendszer orientált operációs rendszer): procfs (/proc) a folyamatok paramétereinek elérése.
 - Voyager I és II szalagos rendszer alapú állományrendszert használt
 - Cassini-Huygens és a Mars szonda (Rover) valós idejű operációs rendszer állományrendszerét (RTOS) használja (flash memória).

Állományrendszerek fejlődése

- A legtöbb operációs rendszer magába foglal egy állományrendszert – a modern operációs rendszerek alapvető magja az állományrendszer kezelés.
- Az első mikroszámítógépek operációs rendszerének egyetlen igazi feladata volt: az állományrendszer kezelése (pl. DOS, melynek a neve is erre utal).
- Néhány kezdeti operációs rendszer egy külön komponensként, lemez operációs rendszerként kezelte az állományrendszer menedzsmentet, sőt némelyik mikrogép külön is töltötte azt be.
- A régi operációs rendszerek csak egy, saját állományrendszert támogattak (pl. CP/M), így annak nem is volt külön neve.
- Lapos (flat) állományrendszer: nincsenek benne könyvtárak, minden állomány ugyanazon a szinten tárolódik (akár merev, akár hajlékony lemez, stb.) Előnye, hogy egyszerű, hátránya, hogy sok állomány esetén már áttekinthetetlen. A kezdeti Apple MacIntosh ilyennel rendelkezett (MFS), de az állománykezelő szoftver képes volt ezeket csoportokba rendezni, a hierarchia látszatát keltve.
- Ma az állományrendszerek zöme a Linux (Unix), Mac OS és Windows operációs rendszerekhez köthetők (ext2, ext3, ReiserFS, Reiser4, GFS, GFS2, OCFS, OCFS2, NILFS, XFS, JFS, HFS, FAT, NTFS). Ezek egy része naplózható (ext3, reiserfs, JFS, NTFS).
- A Linux rendszereken a könyvtárakat a / jel választja el, és a rendszer megkülönbözteti a kis- és nagyűket, míg a Microsoft rendszereken a \ az elválasztó karakter, és a kis- és nagybetű nem számít különbözőnek.
- Az egyetemi központi szerver (ludens) OpenVMS-t futtat, mely hierarchikus, támogatja a hozzáférési listát, rekord orientált és nyomon követi az állományok verzióit.

MicroSoft állományrendszerek

A windows állományrendszerek együtt fejlődtek az operációs rendszerrel:

- CP/M a legkedveltebb rendszer 1979-1980-ban, de az IBM PC megjelenésével (1980 vége) az IBM nem tud egyezsége jutni a licenszelésben, és a CP/M átírásában 16 bitre.
- A MicroSoft ekkor rendelkezik a MicroSoft BASICkel (1975-ben az Altair 8800 gépre kifejlesztett értelmezővel), de nincs operációs rendszere.
- 1980-ban Tim Paterson megírja a QDOS (Quick and Dirty Operating System) 4 hónap alatt az új 8086-os processzorra. A QDOS sokat merít a CP/M-ből és a Digital VAX/VMS rendszeréből (86-DOS néven kerül forgalomba). A QDOS nem a CP/M állományrendszerét használja, hanem a MicroSoft BASIC által is támogatott FAT állományrendszert (1977, Bill Gates és Marc McDonald, létező technikák alapján).
- A MicroSoft megkötí a licensszerződést az IBM-el a QDOS implementálására az IBM PC-re és felbérlí Tim Pattersont. 1981-ben az IBM megveszi a 86-DOS teljes jogát, teszteli a PC-DOS-t, és megírja hozzá a manuált, és nem kizárólagos licensszerződés keretében megveszi a PC-DOS 1.0 használati jogát. Az IBM \$60-ért árulja a szoftvert (a CP/M ára ekkor \$240). A CP/M birtokosa, a Digital Research be akarja perelni a MicroSoftot, mivel az gyakorlatilag lemásolta a CP/M rendszerhívásait, de aztán eláll a pertől, mivel realizálja, hogy az IBM-et is be kéne perelnie.
- 1983-ban kijön a PC-DOS 2.0, mely teljesen újra lett írva. Egészen 1991-ig, az MS-DOS 5.0-ig az egyetlen MicroSoft szolgáltatta szövegszerkesztő az EDLIN (sorszerkesztő), melyet egy teljes képernyős szerkesztő vált. Az MS-DOS (1982 már a Unixból is sokat átvesz).
- A ...DOS operációs rendszereket a FAT állományrendszert használták.

FAT: File Allocation Table:

- FAT12 / FAT16 / FAT32: 12, 16, illetve 32 bites verzió

| | FAT12 | FAT16 | FAT32 |
|-------------------|----------------------|-----------------|----------------|
| megjelenés | 1977 (MS Disk BASIC) | 1988 MS-DOS 4.0 | 1996 Windows95 |
| Max. file méret | 32 Mb | 2 Gb | 4 Gb |
| Fileok max. száma | 4077 | 65517 | 268 435 437 |
| Max. lemeznagyság | 32 Mb | 2 Gb (4 Gb) | 2 Tb |

- A FAT12 nem hierarchizált állományrendszer, viszont a kontroll információ belefért az első sávba a hajlékony lemezen, ezért nem kellett állandóan mozgatni a fejet, és elég jó várakozási időt eredményezett.
- A DOS csak egy FAT partíció kezelésére volt felkészítve. 1986, ban az MS-DOS 3.2-ben jelenik meg a kiterjesztett (extended) partíció, és a logikai meghajtók (először egy, majd a 3.3 verzióban 24). A logikai meghajtók is a lemezen definiált struktúrák, az MBR (master boot record)-hoz hasonlóan, és láncszerűen vannak felfűzve. Csak egy kiterjesztett partíció lehet, és nem bootolható, és csak az elsődleges partíciók után definiálható (melyek az MBR-ben vannak leírva).
- Az állományok formátuma: 8.3. Ezt a később bevezetett kiterjesztések (VFAT) megnövelik.
- Mivel a FAT nagyon egyszerű, és minden operációs rendszer támogatja, természetesnek tűnt, hogy a flash kártyák, digitális kamérák, PDA-k is ezt alkalmazzák. Ezen felbuzdúlva a MicroSoft 2003 decemberében bevezette rájuk a licenszdíjat.

NTFS (New Technology File System)

- A Windows NT, 2000, XP, Server 2003 és Vista alap állományrendszere
- A FAThoz képest számos új lehetőséget tartalmaz:
 - Metaadatok tárolása (állománynév, létrehozás dátuma, hozzáférési jogok, indexálás kereséshez ...)
 - Fejlettebb adatstruktúrák a teljesítmény és megbízhatóság javítására
 - Biztonsági rendszerek (hozzáférési lista)
 - Állományrendszer naplózás
 - Kvóta rendszer
 - Más állományrendszerek beillesztése a fastruktúrában (ld. Unix csatolási pont)
 - Szimbólikus és merev utalások (ld. Unix állományrendszer)
- A pontos specifikáció nem publikus, üzleti titok. Ennek oka, hogy más operációs rendszerek nem képesek egyelőre teljes támogatást adni az NTFS állományrendszerre (tipikusan olvasás).
- 5 verziója van (NT: 1.0, 1.1, 1.2, 2000: 3.0, XP-: 3.1)
- Az állománynevek UTF-16 kódolással tárolódnak.
- A rendszerállományok indexelésére a B+ fastruktúrát használja (mely lehetővé teszi az egyszerű betoldást és törlést. Az adatok mind egy szinten vannak, egy példányban, és másutt csak utalások vannak rájuk).
- A Master File Table (MFT) metaadatokat tartalmaz minden állományról és könyvtárról, mint a helyük, méretük és védelmük. A defragmenter programok ennek alapján dolgoznak.

Unix állományrendszerek

- A Unix (Linux) operációs rendszerben minden eszköznek van egy eszközneve, de az állományokat nem ezen keresztül érjük el.
- Az operációs rendszer létrehoz egy virtuális állományrendszert, melyben az összes eszköz minden állománya egy hierarchia alatt látszik: a / (gyökér – root) könyvtár alatt.
- A gyökér könyvtárnak nem szükséges fizikai eszközön elhelyezkednie, akár hálózati eszköz is lehet.
- A különböző eszközök helyét a gyökér alatt a *csatolási hely* (mount point) határozza meg: az operációs rendszernek meg kell mondani, hogy az adott eszközt a gyökér alatt melyik könyvtárhoz rendelve találja meg (pl. /mnt). A csatolási pontok kijelölése és a csatolások végrehajtása általában a rendszergazda (root) feladata.
- A Unix rendszerek lehetőséget biztosítanak automatikus felcsatolásokra is:
 - A /etc/fstab állomány írja le a rendszerben definiált csatolási pontokat és eszközöket
 - A hordozható eszközök (CD, DVD, flash memória, ...) a rendszerhez való csatlakoztatáskor felcsatolódhat az előre megadott csatolási pont alá.
 - Supermounting: a hordozható eszközök szabadon ki-be dughatók, a rendszer felismeri, ha kicseréltük a hordozható eszközt, és újracsatolja más helyre (gondoskodik a megfelelő szinkronizációról).
 - Autocsatolás (autofs): a leggyakrabban a hálózati eszközöknél alkalmazott eljárás, a rendszer igény esetén automatikusan felcsatolja, majd tartós nem használat esetén lecsatolja.

MacIntosh állományrendszer

- A Mac OS X az operációs rendszer előző verziói által használt HFS (Hierarchical Files System) továbbfejlesztése: HFS Plus.
- Metaadatok tárolását erősen támogatja
- Megkülönbözteti a kis- és nagybetűket
- Unix eredetű, így Unix szerű jogosultságkezelésre képes
- Naplózás
- Beépített optimalizáló és defragmenter
- Unicode alapú tárolás
- Szimbolikus link, merev link és alias (követi a filel átnevezés vagy átmozgatás esetén) létezik.