

Szoftverismeret



Gál Zoltán

Szoftverismeret

2018. július 26.

Tartalomjegyzék

1. A szerzői jog.....	9
1.1. A hardver és a szoftver.....	9
1.2. Az adatvédelmi törvény.....	10
1.3. A szerzői jogi törvény.....	11
1.4. Szoftverek szerzői joga, a licenc fogalma.....	11
1.5. Szoftverek jogi csoportjai.....	12
1.5.1. Kereskedelmi szoftverek.....	13
1.5.2. Shareware.....	13
1.5.3. Freeware.....	14
1.5.4. Nyílt forráskódú szoftverek.....	14
1.5.5. Public domain.....	16
1.6. Creative Commons.....	16
2. Szoftverek funkció szerinti csoportosítása.....	19
2.1. Az operációs rendszer.....	19
2.2. Segédprogramok.....	21
2.2.1. Biztonsági szoftverek.....	21
2.2.2. Karbantartó szoftverek.....	21
2.2.3. Tömörítő programok.....	21
2.2.4. Archiváló szoftverek.....	22
2.2.5. Állománykezelők.....	22
2.3. Alkalmazások.....	23
3. Operációs rendszerek.....	27
3.1. Az operációs rendszer feladatai.....	27
3.1.1. A számítógép hardverelemeinek kezelése, működtetése.....	27
3.1.2. Alkalmazások futtatása.....	27
3.1.3. A feldolgozás ütemezése, az erőforrások megosztása.....	28
3.1.4. Adatok kezelése, átvitele.....	29
3.1.5. Párbeszédés kapcsolattartás a felhasználóval.....	31
3.1.6. Működési zavarok kezelése, jelzése.....	31
3.2. Elterjedtebb operációs rendszerek.....	31
3.2.1. UNIX.....	33
3.2.2. Disc Operating System = DOS.....	35
3.2.3. Apple Macintosh.....	36
3.2.4. IBM: OS/2.....	37
3.2.5. Microsoft: Windows.....	37
3.2.6. Bezárult a kör: UNIX a PC-n.....	38
3.2.7. Mobil operációs rendszerek.....	39
4. Számítógépes kártevők.....	41
4.1. A kártevők fajtái.....	41
4.1.1. Önszaporító vírusok.....	41
4.1.2. Programférgék.....	41
4.1.3. Kémprogramok.....	42
4.1.4. Trójai programok.....	42
4.1.5. E-mail vírusok.....	42
4.1.6. Halászó oldalak.....	42
4.1.7. Hoax, álhír.....	42
4.2. A kártevők elkerülése.....	43
4.3. Miről ismerjük fel a bekövetkezett fertőzést?.....	43

4.4. Mit tegyünk fertőzés esetén?.....	44
Felhasznált irodalom.....	45

1. A szerzői jog

Bár a lecke címe a szerzői jogra vonatkozik, ennél kicsit több fogalom tisztázása a célja. Először is érdemes véglegesen definiálni a hardvert és a szoftvert, hiszen ez a témakör ez utóbbival foglalkozik elég részletesen. Ezután következhet az adatok és a szoftverek használatának jogi szabályozása.

1.1. A hardver és a szoftver

Tehát mielőtt továbbmegyünk, tisztázzuk, hogy mi a hardver és mi a szoftver! A hardver definíciója már szerepelt a Hardverismeret témakörben. Íme itt az ottani definíció megismételve:

Hardver (angolul *hardware*): A számítógép fizikai, elektronikai alkatrészeinek összessége által alkotott műszaki eszköz(ök).

Ehhez képest már definiálható a szoftver is:

Szoftver (angolul *software*): A hardvert működtető vagy használó szellemi termék.

Ez a definíció egyrészt megadja a szoftver és a hardver közötti kapcsolatot is, másrészt tartalmazza a *szellemi termék* kifejezést, amelyről ennek a leckének a további szakaszai szólnak majd.

Mint a definíció mutatja, a szoftver lehet a hardver működéséhez elengedhetetlen programteremék (operációs rendszer, rendszerszoftverek) vagy lehet olyan program, amellyel a számítógépet valamilyen számunkra hasznos tevékenységre alkalmazhatjuk (alkalmazás). A szoftverek ez alapján történő csoportosítása majd egy későbbi lecke feladata lesz.

Érdekes viszont, hogy a határ a hardver és a szoftver között nem olyan egyértelmű, mint amilyennek a fenti definíciók mutatják. Ugyanazt a feladatot el lehet végezni szoftverrel és hardverrel is. Hogy mikor melyik megoldásra kerül sor, azt a hardver gyártója dönti el. Mint a processzorok ismertetésénél szó volt róla, a CISC processzorok belsejében van egy kis memória, rajta egy mikroprogrammal, amely egyszerű lépésekre bontja a processzor összetett utasításait. Ez a mikrokód a hardver része vagy szoftver? A hardver része, de egyrészt szellemi termék, másrészt nem fizikailag létező, hanem program, tehát inkább szoftvernek nevezhető.

Ugyanakkor vannak olyan célszámítógépek, amelyek egy bizonyos feladatra készültek, amelyet esetleg külön szoftver nélkül el tudnak végezni. Ezek egy részében a szoftver egy ROM-ban van elhelyezve, ezért tűnik úgy, mintha nem lenne szoftver. Más részük azonban lehet, hogy tényleg nem tartalmaz szoftvert, hanem olyan speciális processzorral rendelkezik, amely az egyébként például egy PC-n szoftverrel végrehajtható feladatot hardveresen megold. Ilyen lehet például egy switch, egy DVD- vagy Blu-ray lejátszó, egy TV set-top-box, amely a videó dekódolást hardveresen elvégzi. Ugyanezt a dekódolást egy személyi számítógép esetén a videólejátszó program végzi – sokkal több erőforrást lefoglalva a gépnek, mint amit a hardveres dekódolás igényel. Viszont egy DVD-lejátszón aligha lehet például egy szövegszerkesztő programot futtatni, mivel nem általános célú processzort, hanem kifejezetten a videó dekódolására alkalmas processzort tartalmaz.

Fontos még megjegyezni, hogy a szoftver fenti definíciója *nem tartalmazza az adatot*, csak magát a számítógép által futtatandó programkódot! **Az adatot nem tekintjük szoftvernek.** Ez alól

csupán azok az adatok kivételek, amelyek egy adott program részét képezik. Éppen ezért külön kell tárgyalni a szoftverre és az adatokra vonatkozó felhasználási jogokat.

1.2. Az adatvédelmi törvény

Magyarországon az adatok felhasználását szabályozó jogszabályt sokszor módosították, sokszor fogják még módosítani; és sokszor sértik meg – sajnos nem csak magánszemélyek, cégek, hanem maga az állam is. Ha figyelembe vesszük, hogy a 21. században az adatok, információk birtoklása milyen fontos értékévé vált, akkor mindenképpen meg kell ismernünk az erre vonatkozó jogszabályokat, különben nem tudjuk a saját érdekeinket sem érvényesíteni.

A jogszabály, amelyek az adatok felhasználását szabályozza: *1992. LXIII. törvény a személyes adatok védelméről és a közérdekű adatok nyilvánosságáról*.¹

A törvény két fontos fogalmat definiál:

Személyes adat: bármely meghatározott természetes személlyel kapcsolatba hozható adat. Az adatból levonható, az érintettre vonatkozó következtetés.

Közérdekű adat: az állami vagy önkormányzati feladatot, valamint jogszabályban meghatározott egyéb közfeladatot ellátó szerv vagy személy kezelésében lévő, valamint a tevékenységére vonatkozó, a személyes adat fogalma alá nem eső adat.

A személyes adat erősebb a közérdekű adatnál, így ha közérdekű adat lenne, de egy adott személlyel kapcsolatos személyes adat is egyben, akkor annak ellenére tilos nyilvánosságra hozni, hogy a közérdekű adatot mindenkinek joga van megismerni.

A különböző szervezetek a működésükhöz természetesen kénytelenek nyilvántartani rólunk adatokat. Ennek módját és időtartamát is szabályozza a törvény. Ilyen szükséges nyilvántartás például egy cégnél a munkavállalókról azok adószáma, TAJ-száma, hiszen adót, egészségbiztosítási, nyugdíjjárulékot kell fizetniük utánuk. Egy kereskedő cég nyilvántarthatja a vásárlóinak adatait – például a számlák kiállításához –, de ezen nyilvántartások nem nyilvánosak és csak a szervezet működésével kapcsolatban szükséges adatok tárolhatóak benne.

Az állami és önkormányzati szervek elsősorban a polgároktól szerzik be a munkájukhoz szükséges adatokat. Amennyiben a polgártól nem szerezhető be a számukra szükséges adat, akkor fordulhatnak az állami nyilvántartásokhoz (mint például a lakcímnnyilvántartás, az autók tulajdonosainak nyilvántartása stb.). Magáncégek azonban nem fordulhatnak az állami nyilvántartásokhoz. Ebből is következik például, hogy bármennyire is macerás tud lenni, ha az adatainkban változás áll be, akkor azt nekünk személyesen kell jelezni az összes közüzemi szolgáltató, bank felé, amellyel kapcsolatban állunk.

A polgár jogosult megtiltani a róla nyilvántartott adatok kiadását harmadik fél részére akkor is, ha valamely cég ezt automatikusan megtenné. Emiatt azok a cégek, amelyek például reklámajánlatokat akarnak ügyfeleiknek küldeni, hivatalos nyilatkozatot kérnek az ügyfelektől arról, hogy ehhez hozzájárulnak.

Hasonló okból kell minden diáknak az iskolába beiratkozáskor nyilatkoznia arról, hogy hozzájárul a róla valamely iskolai rendezvényen készült kép- és videofelvétel nyilvános közzétételéhez. A törvény ugyanis szabályozza azt is, hogy az ilyen felvételek alapesetben személyes adatként keze-

¹ <https://mkogy.jogtar.hu/?page=show&docid=99200063.TV>

lendőek. Kivétel ez alól, ha a felvételen sokan láthatóak, mert ebben az esetben egy nyilvános tömegrendezvény felvétele már szabadon felhasználható.

1.3. A szerzői jogi törvény

A másik fontos törvény az 1999. LXXXVI. törvény a szerzői jogról.² Ez a törvény nemzetközi szabályozásokat is alkalmaz a szellemi termékekkel kapcsolatos jogok egységes kezelésére, de magyar sajátosságokat is szabályozhat. A törvény szintén tartalmaz egy meghatározást a szoftverre, amely a fogalmat kifejezetten a szerzői jog szempontjából próbálja meg definiálni:

„Szoftver: A számítógépes programalkotás és a hozzá tartozó dokumentáció. Annak bármely formája vagy fajtája védelmet élvez.”

Ez a definíció gyakorlatilag azt a részét fejt ki a mi definíciónknak, hogy mit értünk azon, hogy a *szoftver szellemi termék*. Mint látható, a szoftverhez hozzáveszi annak dokumentációját is, amely nélkül nem volna szabad szoftvernek léteznie.

A törvény szerint a szerzői jog azt illeti, aki a művet megalkotta. Őt nevezzük szerzőnek. Fontos, hogy egy átdolgozott mű is szerzői jogi védelem alá esik, *„ha annak egyéni, eredeti jellege van”* (idézet a törvényből). Amennyiben több szerzője van a műnek, akkor *„ha annak részei nem használhatók fel önállóan, a szerzői jog együttesen és – kétség esetén – egyenlő arányban illeti meg a szerzőtársakat, a szerzői jog megsértése ellen azonban bármelyik szerzőtárs önállóan is felléphet.”* Amennyiben a mű különböző szerzők által készített részei önállóan is felhasználhatóak – például ha egy könyv egy-egy fejezetét más-más szerző írta –, akkor a törvény szerint az egyes részek felett a szerzői jogokat a megfelelő szerző önállóan is gyakorolhatja.

Együttesen létrehozott műnek nevezi viszont a törvény az olyan műveket, amelyek *„megalkotásában együttműködő szerzők hozzájárulásai olyan módon egyesülnek a létrejövő egységes műben, hogy nem lehetséges az egyes szerzők jogait külön-külön meghatározni.”* Ebben az esetben a szerzői jog azt a természetes vagy jogi személyt illeti meg, akinek vagy aminek a kezdeményezésére és irányításával a művet létrehozták és aki vagy amely azt a saját nevében nyilvánosságra hozta. Ha a törvény mást nem mondana, akkor ez alapján egy film vagy egy szoftver esetében ezt a szabályt kellene alkalmazni (de a törvény ezt később megerősíti és részletesebben is kifejti).

A szerzői jog egy része átruházható, más része nem. Nem átruházható például a szerző személyéhez fűződő joga. Ezekről nem is mondhat le, nem öröklődik. Tehát például az Arany János által írt művek esetében minden körülmények között Arany János marad a szerző. Szintén személyhez fűződő jog az is, hogy a mű nyilvános-e: a szerző dönthet a mű nyilvánosságra hozataláról és annak visszavonásáról is.

A vagyoni jogok örökölhetők, azokról halál esetén rendelkezni kell. A vagyoni jogokat öröklés útján megszerző személyek azokról egymás javára rendelkezhetnek. A vagyoni jogok át is ruházhatóak. A szerzői jog ezen része tehát pontosan úgy viselkedik, mint bármely más vagyontárgy.

1.4. Szoftverek szerzői joga, a licenc fogalma

A szoftverekre vonatkozó szerzői joggal kapcsolatban először a szerzői jogi törvény 30.§-át célszerű megnézni: *„(1) Eltérő megállapodás hiányában a mű átadásával a vagyoni jogokat a szerző jogutódjaként a munkáltató szerzi meg, ha a mű elkészítése a szerző munkaviszonyból folyó kötelessége.”* Bár ez általában minden munkaviszony keretében születő műre vonatkozik, figyelembe véve, hogy a szoftvereket általában nem egy ember készíti, hanem egy szoftvercégnél dolgozók programozók együtt, a fenti rendelkezés a legtöbb szoftverre is vonatkozik!

² https://net.jogtar.hu/jr/gen/hjegy_doc.cgi?docid=99900076.TV

A **licenc** vagy más néven a **felhasználói szerződés** határozza meg, hogy a szoftver felhasználójának *milyen feltételek mellett mire van joga* az adott szoftver felhasználása során. Az egyes felhasználói szerződések eltérőek lehetnek abban, hogy a szoftver felhasználását milyen célra, milyen felhasználói kör számára és hány példányban engedélyezik.

Ez alapján létezik olyan megoldás, amely például magán célú felhasználásra ingyen, de kereskedelmi célú felhasználásra már díj fizetése ellenében engedélyezi csak a felhasználást. Hasonló módon a licenc meghatározhatja azon számítógépek számát, amelyekre a szoftver egyidejűleg feltelepíthető. Manapság egyre terjed az a megoldás is, hogy a szoftver használatáért évente egy meghatározott díjat, egyfajta bérleti díjat kell fizetni.

A szerzői jogi törvény szerint (59.§ (2)): *„A felhasználási szerződésben sem zárható ki, hogy a felhasználó egy biztonsági másolatot készíthessen a szoftverről, ha az a felhasználáshoz szükséges.”*

Vannak speciális licencek, amelyek a nagy mennyiségű szoftverfelhasználóknak kedvezményeket adnak. Más speciális megoldás az OEM (Original Equipment Manufacturer) licenc, amely a számítógépforgalmazóknak teszi lehetővé, hogy a számítógépeket előtelepített szoftverrel forgalmazzák.

Egy felhasználó nem tehet mást: vagy elfogadja a szoftverhez tartozó licenct, vagy nem használja a szoftvert. **A licencben leírtaktól eltérő módon történő felhasználás jogellenes és ennek megfelelő következményeket vonhat maga után!**

1.5. Szoftverek jogi csoportjai

Az alapján, hogy a licenc pontosan milyen jogokat ad a felhasználásra, több csoportba lehet sorolni a szoftvereket. Ezen csoportok ma már nem teljesen átjárhatatlanok, de egyértelműek a szoftverek felhasználására vonatkozó jogok szempontjából. A csoportosításnak több változata is található az interneten, könyvekben, de ezek a főbb csoportok szempontjából megegyeznek. Egy rövid, de jó összefoglalója a csoportoknak az interneten GROSS BALÁZS jogi megközelítése ([1]).

Az egyik fontos, amit sokan elfelejtenek: **a felhasználói szerződés nem ruház át a felhasználóra tulajdonjogot**, csupán a felhasználás feltételeit határozza meg, azaz felhasználási jogot ad. A szoftverek továbbadására nincs lehetőség!

Egy szoftverlicenc többek között megadja a választ a következőkre:

- A szoftver beszerzése ingyenes vagy kizárólag pénz ellenében lehet azt legálisan felhasználni?
- A felhasználó tevékenységére van-e valamilyen korlátozás?
 - Lehet időkorlát
 - Lehet a végezhető tevékenységek fajtájára (magán célú, oktatási célú, céges tevékenység)
- Megismerhető-e a szoftver forráskódja, módosítható-e az?
- Amennyiben a szoftver forráskódja megváltoztatható, az így létrejövő új szoftverre milyen jogosultságok érvényesek?

A fenti kérdésekre adott válaszok alapján a szoftverek négy fő csoportját hozhatjuk létre, amelyek bemutatására a következő négy alszakaszban kerül sor. Ezek közül az első három csoportot nevezzük együtt „tulajdonosi” (**proprietary**) szoftvereknek, mivel azok csak felhasználási jogot adnak a szoftverekhez, míg a negyedik csoportba a *szabad szoftverek* tartoznak.

1.5.1. Kereskedelmi szoftverek

Angolul *commercial software* a neve a legszigorúbb szabályozást alkalmazó szoftvereknek. Ezeket a szoftvereket a készítőjük kifejezetten azért hozza létre, hogy kereskedelmi forgalomba hozza. Eből következően ezeket a szoftvereket meg kell vásárolni, hogy használhassuk. Vigyázat: csak használati jogot vásárolhatunk, tulajdonjogot nem!

A kereskedelmi szoftverekhez általában dokumentációt is kapunk – leggyakrabban elektronikus formában. Jó esetben – legalább egy ideig – frissítéseket is ingyen kaphatunk és a szoftver készítője valamilyen mértékű garanciát vállal a szoftverért.

Ilyen kereskedelmi szoftver például az Apple operációs rendszere, az OS X, vagy a Microsofté a Windows. Rosszmájúan megjegyezhetem, hogy utóbbi esetében például arra kaphatunk garanciát, hogy sokat fogunk a használat során bosszankodni (1. ábra).

De nagyon sok felhasználói program is ebbe a kategóriába tartozik, mint például az Adobe Photoshop, a MS Office vagy a legtöbb játékprogram, de ilyen például a Parallel Desktop, amellyel Macintoshon egy virtuális gép hozható létre, amelyben másik operációs rendszer futtatható úgy, hogy az „azt hiszi”, hogy közvetlenül a hardveren fut.



1. ábra: "We suck more and that's guaranteed"

1.5.2. Shareware

A kereskedelmi szoftvereknek sokszor létezik egy ingyenes próbaverziója, amellyel a vásárlás előtt a felhasználónak van lehetősége megismerkedni a program képességeivel. Ezután ha úgy gondolja, hogy megfelelő arra a célra, amelyre használni szeretné, akkor megvásárolja a teljes változatot.

Ez a próbaváltozat valamilyen formában korlátozza a program használhatóságát, hogy a felhasználó előbb-utóbb mindenképpen megvásárolja a fizetős teljes változatot. Ez a korlátozás többféle lehet:

- Lehetséges időkorláthoz kötni a felhasználást. Ez az első használatától számított 30, 90 nap lehet, de az is előfordulhat, hogy meghatározott számú alkalommal lehet a programot elindítani.
- Előfordulhat, hogy a program bizonyos funkciói nem érhetőek el, csak a megvásárlás után.
- Előfordulhat az is, hogy a program ugyan teljes értékű, de a felhasználást nehezítő elemeket tartalmaz. Aki használta már a Total Commander nem megvásárolt változatát, az tapasztalhatta, hogy minden indításkor egy ablak jelenik meg és az 1, 2, 3 gombok valamelyikére kattintva lehet továbbhaladni. Másik példa, hogy a program időnként véletlenszerű helyzetekben feldob egy megvásárlásra felszólító ablakot, amelynek bezárása kell a további használatához. Van olyan videoszerkesztő program is, amelynek próbaverziója az elkészített videóra helyez valamilyen emblémát, feliratot véletlenszerű pozícióba és időben. Érdemes odafigyelni, amikor az interneten egy programot úgy hirdetnek, hogy „FREE”. Ezekről kiderülhet, hogy miután elkezdjük használni, egyszer csak közli, hogy most már ideje megvásárolni elég komoly pénzért!
- A Winamp nevű népszerű zenelejátszó program egy negyedik megoldásra példa. Erről a programról a legtöbben azt hiszik, hogy freeware, pedig valójában shareware: Létezik egy kb. 5 euróért megvásárolható változata, amely az ingyenes változat szolgáltatásait egy plusz funkcióval, az audió-CD lemezekről történő közvetlen zeneátkódolás lehetőségével van kiegészítve. Sok egyéb olyan program is van, amelynek van egy adott szolgáltatáskészletű ingyenes változata és egy bővebb szolgáltatásokat nyújtó fizetős, „Pro” változata.

A shareware programok egy része esetében a vásárlással gyakorlatilag nem is kapunk új programot, csak egy kódot, amelyet a megfelelő párbeszédablakba beírva lehet a korlátozást feloldani. Ez abból a szempontból talán tisztességesebb megoldásnak tűnik, hogy így a megvásárolt program is úgy fog működni, mint a próbaverzió, kevésbé lehet a leendő felhasználót „beetetni”.

1.5.3. Freeware

Olyan szoftver, amelyet mindenféle díjfizetés nélkül lehet használni. Éppen ezért szabadon másolható, továbbadható (kizárólag ingyenesen!), de nem módosítható. Az ilyen szoftverek készítői általában más szoftvereiket pénzért kínálják vagy jobb esetben a felhasználót a fejlesztés támogatására kérik (donation), esetleg a szoftver honlapja tele van reklámokkal, amelyekből befolyó bevételből fedezik a fejlesztést. (Ne feledjük: a szoftver fejlesztése komoly munka, így ne várjuk el senkitől, hogy teljesen ingyen végezze!)

Kicsit hasonló megoldás, amikor az ingyen beszerezett programban bizonyos helyeken reklámok jelennek meg. Ezt nevezik *adware*-nek. A reklámok a felhasználást nem zavarják, viszont a fejlesztőnek bevételt hoznak, mivel a megjelenő reklámért fizetnek neki (mint a fentebb leírt, honlapon megjelenő reklámok esetén).

1.5.4. Nyílt forráskódú szoftverek

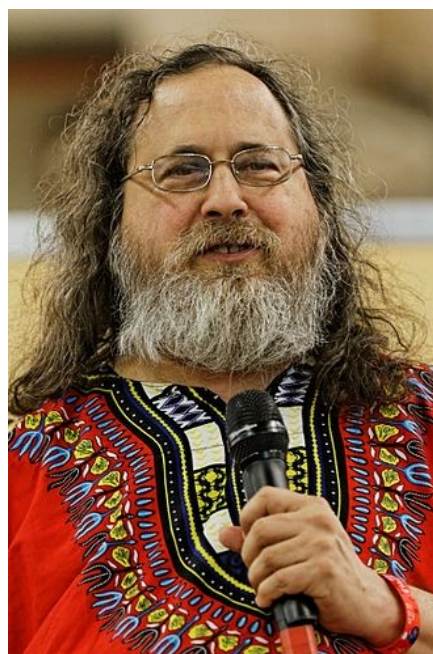
A szoftverek másik nagy csoportja az előzőekkel ellentétben, ahol részben vagy egészben a felhasználó a szoftver tulajdonosává is válhat – ha akar. Ennek érdekében ezeknek a szoftvereknek a forráskódja is nyilvános, ezáltal szabadon módosítható. Persze nem teljesen megkötések nélkül!

A nyílt forráskód megszületése RICHARD STALLMAN (2. ábra³) nevéhez köthető. Eredetileg a Harvard University és az MIT munkatársa volt programozóként abban az időben, amikor a szoftverek fejlesztésével még az amerikai egyetemek foglalkoztak. A munkája során szembesült azzal, ahogy a munkatársai az egyes szoftvereket egymás után próbálták az egyetemek közös fejlesztéséből kivonva kereskedelmi cégekbe vinni és elzárni azok továbbfejlesztési lehetőségét mások elől. Valójában ezzel szembeni mozgalomként indította el a később Szabad Szoftver Alapítvánnyá (Free Software Foundation) alakult folyamatot.

Stallman alkotta meg a *nyílt forráskódú szoftver*, azaz az **Open-Source Software** fogalmát és az ehhez kapcsolódó egyéb fogalmakat.

A kereskedelmi szoftverekkel szemben, amelyek a *copyright* fogalommal (magának a szerzői jognak az angol neve) megalkotta a *copyleft* szót. Ez azt jelenti, hogy míg a kereskedelmi szoftverek esetében a másolás jogát a jogtulajdonos megtartja, a *copyleft* alapján a nyílt forrású szoftvereknél a másolás joga **megmarad** a felhasználónál. Azaz a felhasználó a forráskódot szabadon módosíthatja, az így létrejött programot szabadon forgalomba hozhatja.

Ám ez még nem jelenti azt, hogy a nyílt forráskódú szoftverrel bármit meg lehetne tenni! Ha így lenne, akkor nem sok értelme lenne ilyen szoftvereket fejleszteni, mivel a Stallman korábbi munkatársaihoz hasonlóan gondolkodó emberek gyorsan írnának belőle egy továbbfejlesztett keres-



2. Ábra: Richard Stallman 2014-ben

³ A kép forrása: https://en.wikipedia.org/wiki/Richard_Stallman

kedelmi szoftvert, aminek már nem lenne publikus a forráskódja. Ennek elkerülésére a nyílt forráskódú programoknak is van licencük, amely például ezt a magatartást azáltal tiltja, hogy előírja a nyílt forráskódú szoftver esetében a nyílt licenc megtartását.

Ez a licenc, amit az FSF (Free Software Foundation) alkalmaz, a **GPL: General Public License**. Maga a licenc elég hosszú, megtalálható magyarul is a <http://gnu.hu/gpl.html> címen. A lényege röviden összefoglalva:

1. A GPL hatálya alá tartozó szoftver forráskódját kötelező publikálni.
2. A forráskódban a GPL szövegének szerepelnie kell.
3. A forráskódban szerepelnie kell a szoftver készítőinek nevének, a szoftver módosítása esetén ezeket kötelező megtartani, legfeljebb az új fejlesztők nevével kibővíve.
4. A GPL hatálya alá tartozó szoftver a fentiek felül szabadon módosítható, de az így létrejövő új szoftverre is automatikusa a GPL-t kötelező alkalmazni.

A 4. pont gondoskodik róla, hogy a licenc ne legyen megváltoztatható, amely alapján a Szabad Szoftver Alapítvány már nyert is bíróságon pert, amikor valamely szoftvercég kereskedelmi szoftverben akarta egy nyílt forráskódú szoftver kódját felhasználni.

A Szabad Szoftver Alapítvány egyébként célul tűzte ki, hogy teljes körű szoftvercsomagot hozzon létre kizárólag nyílt forráskódú szoftverekből. Ez lett a **GNU**, amely egy különös hagyományt is teremtett a rekurzív rövidítéssel. Ugyanis a GNU mozaikszó annak a rövidítése, hogy *GNU is Not Unix*. (Másik tipikus példa az ilyen rekurzív rövidítésre a szintén nyílt forráskódú KDE = KDE Desktop Environment).

A GNU a nagygépeken használt Unix rendszerek nyílt forráskódú változatainak létrehozását célozta eredetileg. Viszont éppen az operációs rendszer létrehozásával nem boldogultak, egészen addig, amíg a Linux meg nem jelent, amely így a GNU operációs rendszerévé vált.

Felmerül a kérdés persze, hogy honnan van pénz a nyílt forráskódú szoftverek fejlesztésére, mitől tudnak ezek fejlődni. Ez persze két egymással összefüggő kérdés, de lássuk a választ!

Egyrészt mivel a forráskód nyilvános, bárki részt vehet a fejlesztésben. Ellentétben a kereskedelmi szoftverekkel, ahol egy szűk csoport előjoga a szoftver fejlesztése, a nyílt forráskód esetében bárki módosíthatja a kódot. Persze van egy kisebb-nagyobb csapat, akik koordinálják a fejlesztést, hibajavítást.

Egy kereskedelmi szoftver esetében a felhasználónak nincs beleszólása, hogy merre fejlődjön. Ha lenne, akkor a MS a 2007-es Office után valószínűleg gyorsan felhagyott volna a hagyományos menüs felület helyett kitalált „szalagos” felület alkalmazásával, ahogyan a nyílt forráskódú konkurencia, az OpenOffice/LibreOffice esetében az láthatóan azóta sem terjedt el. A nyílt forráskódú szoftverek jobban idomulnak a felhasználói igényekhez.

A fejlesztést sokszor olyanok végzik, akiknek a hobbijuk a programozás (aki már kipróbálta, tanúsíthatja, hogy élvezetes hobbi tud lenni).

Am az is előfordul, hogy egy cégnek van szüksége valamilyen szoftverre, aminek fejlesztésére kér fel pénzért embereket. Ennek egyik módja éppen az lehet, hogy olyanokra vadásznak, akik tudását már bizonyítja valamilyen szoftver elkészítése – erre is kitűnően alkalmas lehet nyílt forrású programok készítése. Másrészt egy-egy ilyen megrendelésre készített programból lehet utána nyílt forráskódú program, amelynek fejlesztéséért esetleg az eredeti megrendelő továbbra is fizet. Neki is megérheti, ha mások is használják, hiszen így gyorsabb fejlesztésre van esély.

Megjegyzendő itt, hogy nemzetgazdasági haszna is van, ha egy-egy nagy amerikai szoftvercég kereskedelmi szoftvereiért kifizetett milliárdok helyett nyílt forráskódú szoftvereket használunk és a hiányzó szoftverek kifejlesztésével hazai programozókat bízunk meg. Elsőre persze az átállás költségesnek látszik, de hosszú távon bizony komoly kiadástól szabadulhatnánk meg országos szinten.

Egy tipikus gazdasági modellje a nyílt forráskódú szoftverek fejlesztésének jól megfigyelhető a Blender esetében. Ez egy nyílt forráskódú 3D animációs szoftver. A fejlesztését a Blender Founda-

tion finanszírozza, amelynek a holland Ton Rosendaal az elnöke. Az alapítvány támogatásával lehet a fejlesztéshez anyagilag hozzájárulni. De ez még nem minden! Az alapítvány honlapján (www.blender.org) webáruház is üzemeltet, ahol vásárolt termékek bevétele szintén a szoftver fejlesztésére fordítódik. Itt kaphatók ajándéktárgyak, de ami ennél sokkal fontosabb: olyan oktatóanyagok, amelyek segítségével meg lehet tanulni a program használatát. Ezen felül az alapítvány által fenntartott Blender Animation Studio időről időre olyan rövid animációs filmeket is készít, amelyeket szintén meg lehet vásárolni.

Ez nem csak újabb támogatási lehetőség, hanem a fejlesztés egy hatékony eszköze is: a fejlesztők és a felhasználók (a film készítésén dolgozó művészek) együtt dolgoznak. A felmerülő problémákon a fejlesztők azonnal elkezdnek dolgozni.

A Blender fejlesztésén több főállású programfejlesztő is dolgozik a Blender Animation Studio amszterdami központjában vagy a világ más pontján. Emellett több száz vagy akár több ezer további fejlesztője is van a szoftvernek, akik nem kapnak ezért fizetést.

További finanszírozási módja is van persze a Blender fejlesztésének. Egyrészt egy havi előfizetést alkalmazó oldalt (<http://cloud.blender.org>) hoztak létre, amelyen keresztül szintén sok hasznos tartalmat tesznek közzé. Az ebből befolyó bevételt a program fejlesztése mellett elsősorban kreatív tartalmak létrehozására fordítják. Másrészt egyre több cég száll be a fejlesztés finanszírozásába. Többek között a nagy videokártya gyártók, mint az AMD és az nVidia is köztük van.

A Blender példája jól mutatja a lehetőségeket:

- Egy-egy népszerű program fejlesztésébe sokan bekapcsolódhatnak, amely így sokkal nagyobb ütemben fejlődik és a felhasználók igényeinek jobban megfelelő lesz, mint egy kereskedelmi szoftver.
- Bár a szoftverért magáért nincs értelme pénzt kérni, hiszen a forráskód ismeretében elvileg bárki lefordíthatja azt magának, de egyéb szolgáltatásokat lehet pénzért nyújtani. Ilyen lehet a felhasználást megtanító oktatóanyag, vagy kiegészítő tartalmak árusítása.

Másik érdekes példa a szintén nyílt forráskódúvá vált népszerű és nagy tudású Unreal Engine üzleti modellje: bár nem kérnek pénzt az igencsak komoly tudást jelentő szoftverért, de a licenc előírja, hogy a vele készült bármely számítógépes játékból szerzett bevétel 5%-a az Unreal Engine fejlesztőit illeti. Tehát mindaddig ingyenes a program használata, amíg bevételhez nem jut a felhasználó. Mind a Blender, mind az Unreal Engine esetén elmondható viszont, hogy aki a programból él, annak természetes, hogy a neki bevételt hozó szoftver készítőinek juttat valamennyit a bevételéből – na jó, Magyarországon ez még sajnos nem ilyen természetes.

1.5.5. Public domain

Ezek a szoftverek nem sorolhatók az előző négy kategóriába. Ennek oka, hogy ezek nem állnak szerzői jogi védelem alatt, szerzőjük ugyanis lemondott erről és a „*közösség fennhatósága*” (angolul ez a *public domain* jelentése) alá helyezte a programot. Az ebbe a kategóriába tartozó szoftverek felhasználási feltételein a szerzője utólag már nem változtathat. Gyakorlatilag az ilyen szoftverekre semmilyen korlátozás nem vonatkozik.

1.6. Creative Commons

A GPL mintájára, amely szoftverekre ad egy nagyon szabad felhasználást, létezik általában a szellemi termékekre is egy hasonló kezdeményezés. Ez pedig nem más, mint a Creative Commons (<https://creativecommons.org/>).

Ez valójában licenceknek egy nagyon rugalmas rendszere, amelyben gyakorlatilag minden lehetőség szabályozható a teljes megszorítástól a teljesen szabad felhasználásig.

A Creative Commons változatai szabályozzák,

- hogyan lehet beszerezni,
- hogyan lehet saját célra felhasználni,
- hogyan lehet új termék előállításához felhasználni,
- milyen jogok illetik az eredeti készítőt egy adott szellemi termék esetén.

A fentiek alapján tipikus változatok a teljesség igénye nélkül⁴:

1. CC0: Globálisan szabad, mindenféle megszorítás nélkül: szabadon módosítható, felhasználható saját (akár kereskedelmi célú) művek részeként is.
2. BY: Ugyanaz, mint a CC0, de az eredeti készítőt mindig kötelező megnevezni.
3. BY-SA: Mint a BY, de az új műre is a Creative Commons BY-SA licenc érvényes (talán ez felel meg leginkább a nyílt forráskódú programok GPL licencének)
4. BY-NC: Kereskedelmi célú felhasználása nem lehetséges. Viszont módosítani lehet.
5. BY-ND: Kereskedelmi célra az eredeti változatban felhasználható, de módosítani tilos.
6. BY-NC-SA: Bár kereskedelmi célra tilos felhasználni, de módosítható. A módosított változatot ugyanezen licenc alatt lehet csak publikálni.
7. BY-NC-ND: Se módosítani, se más művek részeként felhasználni nem lehet.

A sokféleség miatt nehéz átlátni, mikor mit lehet a művel tenni. A Creative Commons az olyan műveknél hasznos, mint például ez is. De az igazi haszna ott van, amikor valaki egy olyan alkotást készít, amelyet más műben fel lehet(ne) használni. Például a fentebb említett 3D animációs szoftverhez valaki készít egy ló modellt és CC-BY-SA licenc használatával publikálja. Ezt valaki felhasználja egy animációs filmben, akkor az jogszerű, ha a film készítői között szerepelteti a ló modell készítőjét (amitől az illető a magyar törvények szerint a film szerzőjévé válik). Ha viszont ugyanaz a modell CC-BY-NC licenc alatt került publikálásra, akkor ez csak abban az esetben jogszerű, ha az elkészült filmért a készítő nem kér pénzt, azaz CC-BY-NC licenc vonatkozik rá.

Végül egy megjegyzés: a nyílt forráskódú szoftvereknél említett Blender animációs filmek mind Creative Commons BY-SA licencűek.

4 Lásd: https://en.wikipedia.org/wiki/Creative_Commons_license

2. Szoftverek funkció szerinti csoportosítása

A szoftver definíciójában mellett, hogy *szellemi termékről* van szó a szoftver céljáról eddig nem esett túl sok szó. A definíció szerint egy szoftver vagy vezérli a hardver működését vagy valamire használhatóvá teszi a számítógépet. Ez alapján az alábbi csoportosítást lehet kialakítani:

0. Mikrokód, rendszerközeli szoftverek
1. Operációs rendszer
2. Segédprogramok
3. Alkalmazások
4. Fejlesztő eszközök

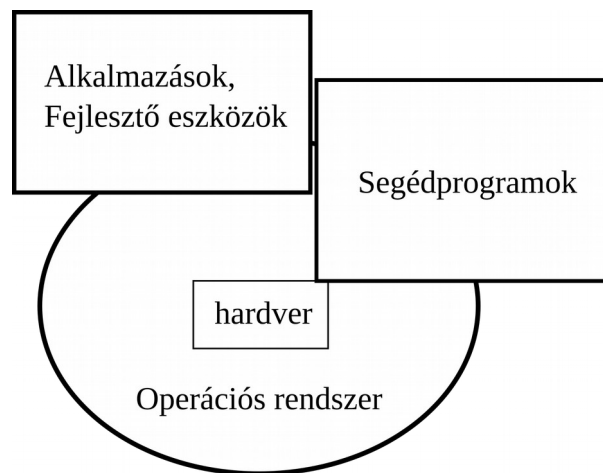
A fenti csoportosítás azért kezdődik a nulladik ponttal, mert mint már volt róla szó, a mikrokód a CISC processzorok belsejében található kód, így nézőpont kérdése, hogy a hardver részének vagy szoftvernek tekintjük-e. A rendszerközeli szoftverek szintén tekinthetők olyan kódrészleteknek, amelyek az adott hardver működésének elengedhetetlen elemei. Például ide tartozik a számítógép rendszerindítását vezérlő kód, amely általában RAM-ban található az alaplapon, például a BIOS részeként a PC-k esetén. Erről a kódról majd egy későbbi fejezetben külön lesz még szó.

A szoftverek viszonyát egymáshoz – és a hardverhez – az 3. ábra mutatja. Ezen látható, hogy a szoftverek közül hivatalosan csak az operációs rendszer az, amely közvetlenül kapcsolatban áll a hardverrel. Az operációs rendszerről majd a következő fejezetben lesz részletesebben szó. Mindenesetre valóban az operációs rendszer az, amelynek a hardverhez hozzá kell férnie, az összes többi szoftverfajta számára az operációs rendszeren keresztül vezet az út a hardver irányába.

Kivétel, mint az ábrán látható, a segédprogramok között lehet néhány, amelyeknek a működésükhöz az operációs rendszert megkerülve, szükségük lehet a hardver valamely részéhez való közvetlen hozzáférésre. Ezekről az esetekről kicsit később, de még ebben a fejezetben lesz szó. Ám azt tudni kell, hogy ezek a közvetlen hardverhez hozzáférések is az operációs rendszer felügyelete alatt zajlanak.

Az alkalmazások és a fejlesztőeszközök együtt szerepelnek az ábrán, mivel ezek csak felhasználási céljukban különböznek egymástól – valójában mondhatjuk azt is, hogy a fejlesztőeszközök az alkalmazások egy csoportját alkotják.

A szoftverek közül azok, amelyek a hardver vezérlésére készültek: az operációs rendszer, a segédprogramok és a rendszerközeli programok. A számítógép használatára szolgálnak az alkalmazások. A fejlesztő eszközök olyan speciális alkalmazások, amelyekkel új szoftvereket lehet fejleszteni.



3. ábra: A szoftverek hierarchiája

2.1. Az operációs rendszer

Az operációs rendszer feladata tehát a hardver vezérlése. Természetesen ennél többről van szó, különben nem foglalkozna vele egy egész fejezet – egészen pontosan a következő.

Mint az 3. ábra mutatja, a hardverhez közvetlenül csak az operációs rendszernek szabad hozzáférnie, minden művelet, ami a hardvert igényli, az operációs rendszeren keresztül, annak kontrollja alatt történik. Emiatt érthetően egy számítógépen egyszerre csak egy operációs rendszer futhat. Per-

sze ez alól van egy speciális kivétel, amikor virtuális számítógépeket futtat egy számítógép. Ebben az esetben viszont a virtuális gépe(ke)t futtató program, mint alkalmazás fogja szimulálni a hardvert a virtuális gép belsejében futó operációs rendszer számára. Magán a fizikai hardveren ekkor is egy operációs rendszer fut (ezt nevezi a virtualizációs technológia „host” rendszernek), a többi operációs rendszer egy alkalmazáson belüli virtuális környezetben (ezt nevezzük „guest” rendszernek) fut.

Az a követelmény, hogy egy hardveren csak egy operációs rendszer fusson és minden más szoftver csak ezen az operációs rendszeren keresztül érje el a hardvert, abból a megfontolásból fakad, hogy így egy szoftver felügyelete alatt működik a számítógép, tehát nem fordulhat elő, hogy két, a hardverhez közvetlenül hozzáférő szoftver egyszerre ugyanazzal a hardverelemmel két különböző tevékenységet akar végrehajtani. Volt idő, amikor ez lehetséges volt, az ilyen rendszereknél ebből mindig a rendszer összeomlása lett.

Az ábrán látható találkozási vonalak az operációs rendszer és a hardver, illetve az operációs rendszer és más szoftver között az a határ, ahol az egyes szoftverek és a hardver egymással kommunikálnak, jól definiált felületen keresztül. Ezt a felületet hívjuk angol kifejezéssel **interface**-nek.

Maga az *interface* minden esetben két különböző „valami” találkozását, egymással való kommunikációjának lehetőségét jelenti. Ebben az esetben az operációs rendszer a számítógépen futó egyéb programok számára kínál egy olyan felületet, ahol azok a számítógép felé továbbítandó kéréseket átadhatják az operációs rendszernek, illetve ahol a műveletek eredményeként kapott adatokat az operációs rendszer átadja az adott programnak. Ez valójában az operációs rendszerhez tartozó rendszerhívások halmaza, ami az adott operációs rendszertől függ. Az ilyen rendszerhívásokat megvalósító függvények halmazát nevezzük *Advanced Programmed Interface*-nek, azaz *API*-nak. Mivel ez operációs rendszerenként változik, így érthető, hogy egy adott szoftver mindig egy adott operációs rendszeren tud futni – azon, amelynek az API-ját használja. Ugyanaz a szoftver természetesen megírható más-más operációs rendszerre is. Ennek kitűnő példája a nyílt forráskódú program – amelynek pont ez az egyik előnye –: ugyanazt a forráskódot a megfelelő operációs rendszerre lefordítva, egy platformfüggetlenül elérhető szoftver áll a felhasználók rendelkezésére, amelynek használhatósága csak kis mértékben függ az operációs rendszertől. Ezzel ellentétben a kereskedelmi szoftverek jelentős része csak egy adott operációs rendszeren érhető el, és mivel a forráskódja nem ismert, a fejlesztő hajlandósága nélkül elképzelhetetlen, hogy más platformon is elérhető lesz valaha.

A hardver és az operációs rendszer közti interface definiálását gyakorlatilag a hardver készítője végzi. Az interface specifikációja kell ahhoz, hogy a hardver vezérléséhez szükséges kód az operációs rendszerbe belekerülhessen, így az hogy egy adott operációs rendszer egy adott hardvert képes-e kezelni, nagy mértékben függ a hardvertől és jelentős mértékben attól is, hogy az adott operációs rendszer fejlesztői megírják-e a hardver vezérléséhez szükséges kódot.

Az egyes hardverelemeket vezérlő programkódot szokás **vezérlőprogram**nak vagy **driver**-nek nevezni. Ezt maga az operációs rendszer is tartalmazhatja, de az egyes hardverek gyártói is adhatják – ez utóbbi esetben a vezérlőprogramot telepíteni kell, aminek hatására az az operációs rendszer szerves részévé válik.

Az interface szónak később még más jelentésével is megismerkedhetünk, de valójában abban az esetben is ugyanerről az esetről van szó, de esetleg az interface két oldalán levő szereplők változnak majd.

Az interface egy speciális fajtája, amikor valamelyik szoftver és a felhasználó kommunikációját kell lehetővé tenni. Ezt *felhasználói felületnek* (*User Interface*) hívjuk. Az operációs rendszertől függően ez lehet karakteres alapú vagy modernebb, grafikus (*Graphical User Interface* = **GUI**).

2.2. Segédprogramok

Vannak esetek, amikor az operációs rendszer nem elegendő a hardver biztonságos és kényelmes vezérléséhez. Ilyen esetekben szoktunk segédprogramokat használni. Elég sok fajtája lehet a segédprogramoknak.

2.2.1. Biztonsági szoftverek

A számítógépen levő adatokhoz illetéktelen személyek is szeretnének hozzáférni. Az ellenük való védekezést segítik a biztonsági szoftverek.

Ezek a víruskereső, vírusirtó programok és a tűzfalak. Másrészt ide tartoznak a különböző titkosító programok is.

2.2.2. Karbantartó szoftverek

Az ebbe a kategóriába tartozó szoftverek vagy a számítógép napi használata során szükséges tevékenységeket nyújtják, vagy az esetleg bekövetkezett hibák kijavítását, megoldását szolgálják.

Például a Windows operációs rendszer által használt állományrendszerek hajlamosak amiatt lassabban elérhetővé válni, mert a tárolt állományok egyes részei összekeverednek: töredezettség alakul ki. Ennek megoldására való az operációs rendszer részét képező töredezettségmentesítő program. Töredezettségmentesítésre a Linux által használt ext állományrendszerek esetében, a Macintoshok által használt HFS+ állományrendszeren azért nincs szükség, mert azok menet közben gondoskodnak róla, hogy úgy helyezkedjenek el az állományok, hogy mindig biztosítható legyen a folytonos elhelyezésük, illetve a háttérben elvégzik az állományok szükséges áthelyezését. Ám ezeken a rendszereken is adódhatnak olyan feladatok, amelyekre szükség van valamilyen segédprogramra.

A karbantartó programok egy részét már maga az operációs rendszer kínálja, mivel azokra elég gyakran szükség lehet. Például a lemezkezelő műveletek (particionálás, formázás) elérésére mindig tartalmaz maga az operációs rendszer is megoldást, de vannak olyan segédprogramok, amelyek ezekhez a funkciókhoz adnak kényelmesebb felhasználói felületet.

A karbantartó szoftverek másik jellegzetes típusa a helyreállító szoftver. Amennyiben valami okból – váratlan áramszünet, fizikai hiba, a felhasználó által végrehajtott művelet – például a háttértáron tárolt állományok elvesznek vagy esetleg az egész partíció eltűnik, akkor fordulunk ezekhez. A legtöbbször a törölt állományoknak jobb búcsút mondani, de vannak esetek, amikor még a megfelelő program használatával azok megmenthetők. Például az ext állományrendszer sajátossága, hogy egy törölt állomány által korábban lefoglalt területet csak akkor foglal el új állomány, ha már máshol nem elhelyezhető. Ilyenkor egy erre alkalmas program képes az üres helyen megtalálni sértetlenül a törölt állomány tartalmát. A Windows által használt állományrendszerek felépítése szintén ad olyan alkalmakat, amikor még a törölt állományok visszaállíthatók.

2.2.3. Tömörítő programok

Sokszor előfordul, hogy egy vagy több állományt kisebb helyen kell elhelyezni, mint ahol elfér. Például mert interneten el kell valakinek küldeni. Ekkor hasznos a tömörítő program, mint például a ZIP. Ez egy meghatározott algoritmust használva átvizsgálja a tömörítendő állományt és egy új kódot állít elő, amelyből később kicsomagolható az eredeti tartalom (UNZIP).

A tömörítési eljárásoknak két csoportja van:

1. veszteségmentes
2. veszteséges tömörítés.

Vannak olyan állományok, amelyeknél nem probléma, ha nem állítható helyre pontosan az eredeti tartalom, a benne hordozott információ megmarad. Ezekben az állományformátumokban eleve van valamilyen redundancia. Például egy kép esetén ha nem minden képpontot tartunk meg, attól a kép még felismerhető marad. Egy zeneállomány esetén vannak olyan hanghullámok, amelyet nem halljuk a többitől, ezek törlése nem rontja észrevehetően a zeneélményt. Egy videó esetében nem csak az előző kettő lehetőséggel lehet a tárolt adatmennyiséget csökkenteni, hanem azzal is, ha nem minden egymást követő teljes képet tárolunk. A megoldás az lehet, ha időnként – például fél másodpercenként egyszer – egy teljes képet tárolunk, míg a közbenső képekből csak az előző képhez képest változott értékű képpontok új színét. Mindezek jó példák arra az esetre, amikor a redundancia csökkentésével történik a tömörítés. Ez a veszteséges tömörítés.

Vannak azonban olyan esetek, amikor egyetlen byte elvesztése is lehetlenné teszi az eredeti információ megőrzését. Ilyen eset tipikusan a programállomány, amelyben az a byte egy végrehajtandó utasítás. Másik példa egy szöveges állomány, amelyből nem megengedett a betűk, karakterek elvesztése. Ezekben az esetekben veszteségmentes tömörítést kell alkalmazni. Ilyen esetben a tömörítés az egyes byte-ok értékeinek előfordulási gyakoriságát szokta kihasználni. Az eredetileg 1 byte méretű adat új mérete a leggyakrabban előforduló értékek esetében néhány bite, a ritkán előforduló értékek esetén akár több byte méretűre is változhat. Természetesen ebben az esetben a tömörített állományba bele kell tenni egy táblázatot is, ami alapján kitömörítéskor visszakódolható minden eredeti byte.

2.2.4. Archiváló szoftverek

Amennyiben biztonságban szeretnénk tudni az állományainkat, akkor azokról érdemes rendszeresen biztonsági másolatot készíteni. Ennek a legegyszerűbb módja, ha például optikai lemezre írjuk azokat. De lehetséges más, a gépre csak ilyenkor rákapcsolt külső merevlemezre mentés is.

Az optikai lemezekre való írást megvalósító program a tipikus példája az archiváló programoknak. És egyben az olyan segédprogramnak is, amely a hardver egy részéhez – konkrétan az optikai lemez meghajtójához – közvetlen hozzáférést igényel. Gondoljuk csak el, mi lenne, ha az egyszer írható CD írási művelete közben döntene úgy a Windows, hogy sürgősen frissíteni akar!

Az archiváló szoftverek másik változata, amely valamilyen speciális adattároló kezelését kifejezetten a biztonsági mentés céljából végzi. Ezek a programok általában különböző mentéseket tudnak végezni. Figyelni tudják például, hogy a legutóbbi biztonsági mentés óta mely állományok módosultak. Így a következő mentés során csak ezeket az új állományokat mentik. Persze időnként ilyenkor is szükséges egy teljes mentés, de sokat lehet spórolni az idővel az ilyen inkrementális mentésekkel, amelyek csak a változást mentik – cserébe ha vissza kell valamit állítani, akkor eltarthat egy darabig, mire sikerül megtalálni.

2.2.5. Állománykezelők

Sokan nem szeretik azt az állománykezelési lehetőséget, amit az operációs rendszer alaphoz kínál. Az ő kedvükért jöttek létre azok a szoftverek, amelyek más felhasználói felületet kínálnak a különböző állománykezelési feladatokra.

Tipikusan ilyen volt a helyzet a DOS idején, ahol parancsokat begépelve lehetett mindent elvégezni, így például egy állomány másolását is. Nem volt túl kényelmes, hogy az embernek fejben kellett tartani, mit honnan hova akar másolni. Erre jelentett választ PETER NORTON fejlesztése, a

Norton Utilities, amely már a nevéből következően segédprogramok gyűjteménye volt. Ennek az állománykezelő része volt a Norton Commander nevű program. Ez annyira sikeres volt, hogy bár a DOS már nem létezik, a Norton Commander tovább él különféle rein-karnációkban.

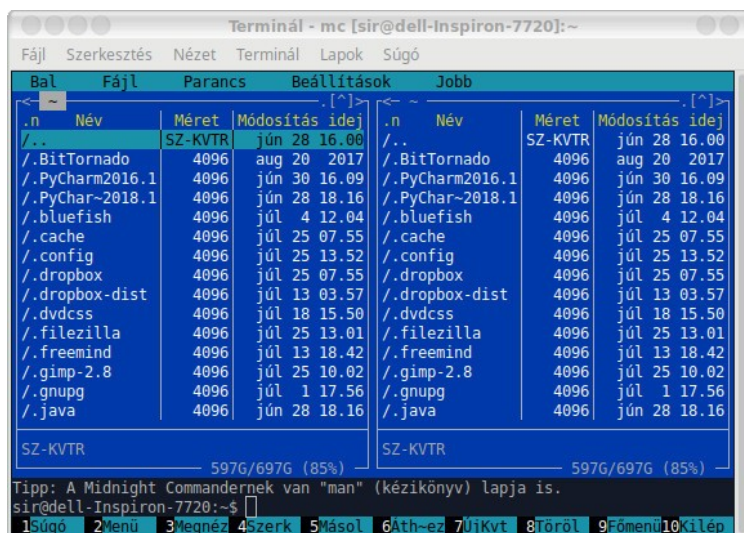
UNIX rendszereken az eredetivel szinte azonos kinézetben – továbbra is karakteres felületet használva – megjelent a Midnight Commander, amely azután Linuxon is létezik (4. ábra).

Mivel a Norton Utilities kereskedelmi szoftver volt, egy idő után DOS alatt is megjelentek a Norton Commander klónjai: Volkov Commander, FAR Commander.

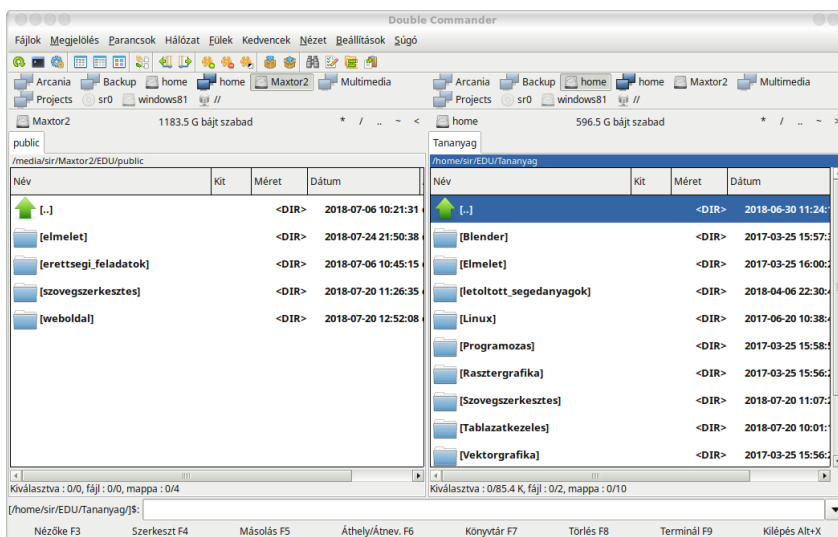
OS/2 alatt is megjelent File Commander néven a másolata, majd a Windows operációs rendszerré válása után először Windows Commander néven, majd miután ez a név védettnek bizonyult, Total Commander néven grafikus felülettel is megjelent. A Total Commander egyébként shareware.

De ma már semmi szükség a shareware Total Commanderre, mivel megjelent egy platformfüggetlen változat is, a Double Commander. Ez gyakorlatilag azonos kinézettel, grafikus felületet használva érhető el Windows, OS X és Linux alatt is. Bár még bőven fejlesztésre szorul, a legalapvetőbb műveleteket, amely miatt sokan megszerették a Norton Commandert és utódait, már el lehet vele végezni.

A program nyílt forráskódú, emiatt lehetséges, hogy platformfüggetlen.



4. ábra: A Midnight Commander az XFCE parancsértelmező ablakában futtatva



5. Ábra: A család legújabb tagja: a Double Commander

2.3. Alkalmazások

A többi szoftver, amelyről eddig nem volt szó, nem a mindennapi használatra teszi alkalmassá a számítógépet, hanem valamilyen hasznos célra, amelyért a számítógépet egyébként vettük.⁵ Ezek a szoftverek az alkalmazások. Igen széles skálájuk van, így gyakorlatilag lehetetlen mindet felsorolni.

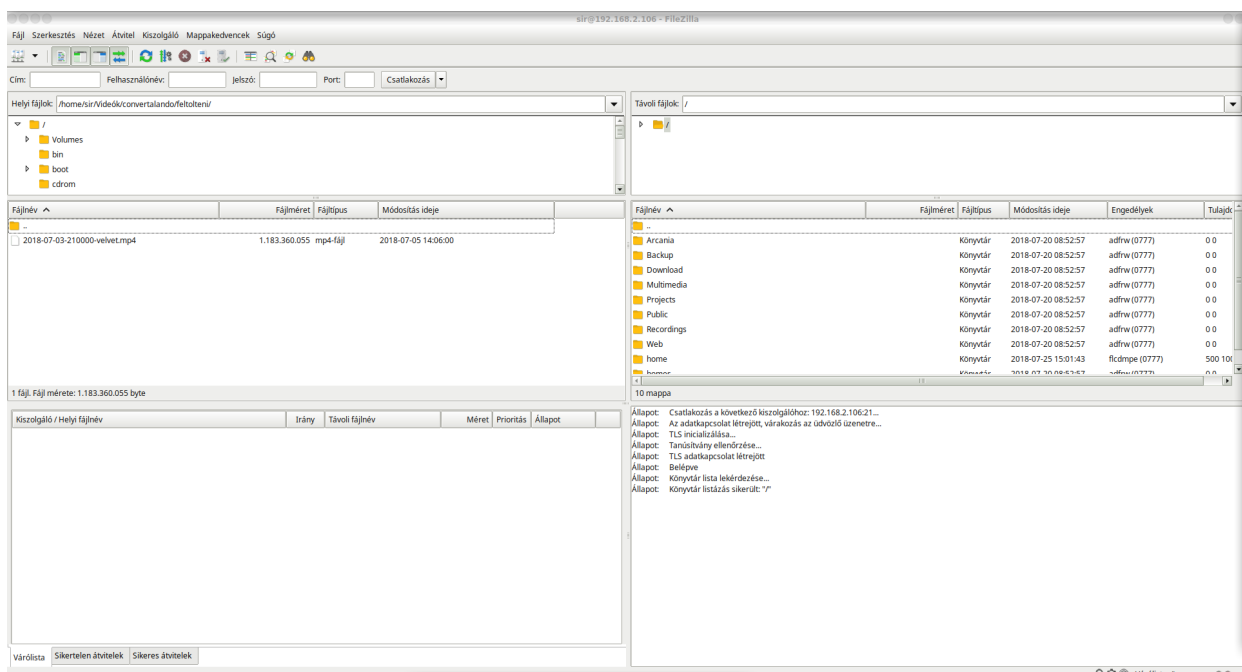
⁵ A számítógép eszköz, amelyet bizonyos feladatok elvégzésére használunk és nem pedig a számítógép irányítja az életünket. Ha valaki úgy érzi, az ő életét a számítógép irányítja és nem ő a számítógépet, az sürgősen cserélje le a Windows Linuxra vagy vegyen egy Macintosht. Ha pedig ezután is így érzi, akkor forduljon pszichológushoz, mert számítógép-függő lett!

Éppen ezért a teljesség igénye nélkül következzen egy felsorolás:

- *Irodai alkalmazások:* Manapság a számítógép egyik tipikus felhasználási helye az irodában van. De itt most az irodát nagyon széles körben kell érteni: minden olyan hely, ahol valamilyen, az irodákból megszokott tevékenység történik. Így tehát irodai alkalmazás például a szövegszerkesztő, a táblázatkezelő, de egy levelező program is ide sorolható. Ha megnézzük, hogy az MS Office vagy a LibreOffice milyen modulokból áll, akkor még megemlíthető irodai alkalmazásként a bemutatókészítő vagy az adatbázis-kezelő – bár ez utóbbit nagyon nem szabad ide sorolni.
- *Internetes programok:* böngésző, levelező, chat-program tartozhat ebbe a kategóriába. De ha alkalmazásnak tekintjük és nem segédprogramnak, akkor például az FTP-kliensek is ide tartoznak, mint például a FileZilla (6. ábra).
- *Média lejátszó:* VLC, Windows Mediaplayer, Banshee, stb. Zene vagy videó vagy mindkettő lejátszására alkalmas változatai vannak.
- *Médiaszerkesztő:* Olyan szoftverek, amelyekkel zenét, videót lehet szerkeszteni. Tipikus példa zeneszerkesztőre az Audacity. Videószerkesztőre példa az Apple Final Cut programja. De ide sorolhatjuk az animációkészítő programokat is, mint például a Blender vagy a fizetős társai.
- *Elmetérkép készítő:* Free Mind, VYM, ...
- *Képszerkesztő programok:* Gimp, Krita, MyPaint, Infranview. Na jó, a Paintet is említjük meg.
- *Játékprogramok:* ezt a kategóriát aligha kell részletezni...
- Még rengeteg olyan alkalmazás van, amelyeket meg lehetne említeni...

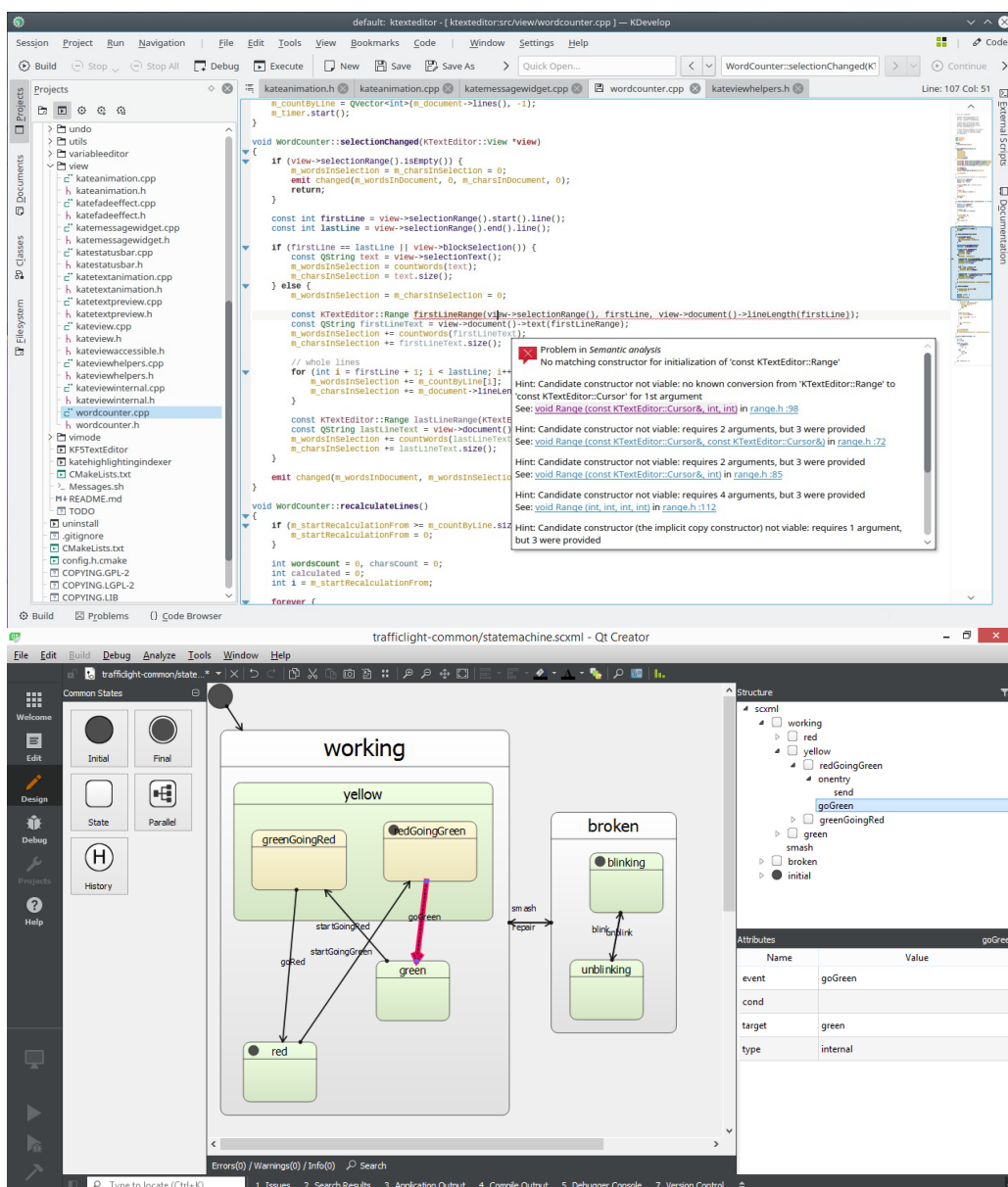
Külön érdemes egy kategóriát megemlíteni, ha már az elején is külön csoportot kapott: azok a szoftverek, amelyekkel más szoftvereket lehet készíteni. Ezek a *fejlesztő eszközök*.

A fejlesztő eszközök alatt elsősorban, de nem kizárólag a különböző programozási nyelvekhez tartozó fordítóprogramokat értjük. A programozók már nagyon kifinomult eszközökkel rendelkeznek, amelyek a szoftverfejlesztést segítik. Ezeket a fejlesztő környezeteket az *Integrated Development Environment* rövidítése alapján IDE néven szokás emlegetni. Íme néhány példa – ismételten a teljesség igénye nélkül ezekre:



6. Ábra: A Filezilla, az egyik legelterjedtebb FTP-kliens

- PyCharm a Python programozáshoz.
- Szintén a Pythonhoz tartozik egy nagyon „fapados” szerkesztő, az Python IDLE, amely a részét képezi a programnyelvnek.
- KDevelop: elsősorban a C, C++ nyelveken programozók használják, KDE grafiku környezetben (de nem csak ott érhető el)
- Qt Creator: a C++ nyelven íródott Qt-programkönyvtárhoz nyújt nagyon hasznos grafikus segítséget. Elsősorban C++ nyelvű programok fejlesztéséhez alkalmas, de mivel a Qt elérhető Pythonban is, így akár Python nyelvű programok is fejleszthetők benne.
- Eclipse: Elsősorban Java fejlesztőknek készült IDE.
- Bluefish: A webfejlesztők kedvenc fejlesztőeszköze. A webfejlesztésben használt valamennyi nyelvhez tartalmaz segítséget, így elsősorban a HTML, CSS, JavaScript, PHP nyelvekhez, de Python, C++, Java nyelv használatára is alkalmas.



7. ábra felül a KDevelop, alul a Qt Creator

3. Operációs rendszerek

Az első generációs számítógépeket még emberek működtették. Emberek – az *operátorok* – töltötték be az adatokat a memóriába, indították el a program végrehajtását, figyelték a gép működését, majd a keletkezett adatok kinyomtatását kezdeményezték. Ahogy a későbbiekben gyorsabbá és több mindenre alkalmassá váltak a gépek, az operátorok munkájának kiváltására született meg az *operációs rendszer* vagy eredeti neve alapján (*operating system*) a *működtető rendszer*. Ez az eredeti név talán jobban elmondja, mit is csinál az operációs rendszer: működteti a számítógépet.

3.1. Az operációs rendszer feladatai

Abból következően, hogy a számítógépet kell működtetnie, az operációs rendszernek számos feladata van. Ezek közül a főbb feladatok bemutatása következik.

3.1.1. A számítógép hardverelemeinek kezelése, működtetése

Ez az elsődleges feladat, hiszen erre készült. Viszont a hardver – legalábbis a PC-k esetében – nagyon sokféle lehet, amelyen ugyanannak az operációs rendszernek kellene futnia. Éppen ezért az operációs rendszertől nem várható el, hogy minden létező hardverelem vezérléséhez kódot tartalmazzon. Ehelyett inkább a korábban már említett vezérlőprogramokat használja.

A rendszer telepítésekor a telepítőprogram végigkérdezi a számítógép minden elemét, így felmérve, hogy milyen hardver van jelen. Az ennek megfelelő vezérlőprogramokat telepíti fel, ha azok rendelkezésre állnak. Ha nem állnak rendelkezésre, akkor ha van internetkapcsolat és interneten elérhető a szükséges vezérlőprogram, akkor letölti. De vannak olyan hardverek, amelyekhez kézzel kell telepíteni a hardverhez kapott lemezzről vagy az internetről letöltve azt. Utólag a géphez adott hardver esetében is általában ez a helyzet.

A lényeg, hogy az adott vezérlőprogram az, amely tudja, mikor milyen vezérlőjelet kell kapnia annak az adott hardvernek. Tulajdonképpen az operációs rendszer magja és a vezérlőprogram között is van egy interface. A hardverrel pedig maga a vezérlőprogram fog ténylegesen kommunikálni.

3.1.2. Alkalmazások futtatása

Ez egy elég összetett feladat. Az első lépés, az adott alkalmazás elindítása. Az alkalmazás programját a háttértárról be kell olvasni a memóriába. Ezután még néhány olyan tevékenységet elvégez az operációs rendszer, amire itt nincs sem hely, sem szükség, hogy kitérjünk. Végül ha az alkalmazás elindult, akkor azt az operációs rendszernek folyamatosan figyelnie kell és szükség esetén ki kell szolgálnia. Amennyiben az alkalmazás befejezi a működését, az operációs rendszernek még szintén van néhány adminisztratív teendője, amellyel most nem foglalkozunk.

Az alkalmazás kiszolgálása például azt jelenti, hogy ha az alkalmazás valamit akar a hardvertől – például egy oldalt a nyomtatón kinyomtatni –, akkor az operációs rendszert fogja a megfelelő rendszerhívással erre kérni. Vagyis a tényleges nyomtatás lebonyolítása az operációs rendszer feladata – amelyet a nyomtató vezérlőprogramján keresztül végez el. Ebből következően az alkalmazásnak semmit nem kell arról tudnia, hogy milyen nyomtató van a gépre kapcsolva.

Hasonló módon ha egy állományt akar megnyitni, akkor azt valójában az operációs rendszer fogja megnyitni. Ezért van, hogy egy adott operációs rendszerben minden program azonos párbe-

szédlablakokat használ: azok valójában nem a programhoz, hanem az operációs rendszerhez tartoznak.

Visszafelé is igaz ez: ha a felhasználó valamit begépel a billentyűzeten az alkalmazás számára, akkor azt valójában az operációs rendszer kapja meg. Szükség esetén – ha nem amerikai kiosztású billentyűzet van beállítva – kicseréli a kapott karakter kódját egy másikra. Ezután pedig átadja a begépelte szöveget az alkalmazásnak.

3.1.3. A feldolgozás ütemezése, az erőforrások megosztása

Az előző feladattal szorosan összefügg, hiszen az egyes alkalmazások között lehet szükséges az erőforrások megosztása. Egyben azt kell ütemeznie, hogy mikor melyik alkalmazás végrehajtása történjen. Hiszen általában egy számítógépen egyszerre több programot futtatunk, mint amennyit az fizikailag a jelenlevő CPU-magok alapján képes lenne. Ez akkor is igaz volt, amikor még egymagos processzorok voltak, de ugyanígy igaz jelenleg is, amikor többmagos processzorokat használunk. Nem beszélve az operációs rendszer által a különböző feladatok elvégzése céljából futtatott, a szemünk előtt általában rejtve maradó folyamatokról.

Egyszóval az operációs rendszernek kell megoldania, hogy látszólag párhuzamosan ugyanazon a processzormagon egymást váltogatva több folyamat is futhasson. Ez a váltogatás, ami olyan gyorsan történik, hogy valóban egyidejű futásnak látszik, az ütemezés. Az operációs rendszer minden folyamatnak kis időszelvényeket ad, amelyek letelte után egy másik folyamat kapja meg a végrehajtás jogát. A döntést, hogy melyik folyamat következzen sok dolog befolyásolhatja. Ezek közül csak az egyik, hogy az a folyamat, amelyik valamilyen erőforrásra vár – például arra, hogy a kezdeményezett állománymegnyitás megtörténjen –, az nyilván fölöslegesen kerülne sorra.

Az erőforrások megosztása kapcsán itt arra is gondolni kell, hogy az alkalmazások ugyanazon a háttértáron levő állományokat használhatják, egyszerre több program is kommunikálni próbál a hálózaton, esetleg több program is nyomtatást kezdeményezhet egyidejűleg. Ilyen esetekben az operációs rendszer feladata megoldani, hogy az egyes erőforrásokat valamilyen folyamat megkapja, de addig a többi folyamatnak, amely ugyanazt az erőforrást akarja éppen használni, várnia kell.

A CPU mellett a másik kiemelt erőforrás a memória, amelyből szintén nagyon sokszor kevesebb van, mint amennyi szükséges. Ilyenkor az operációs rendszernek el kell érnie, hogy az egyes alkalmazások mégis megkapják a nekik szükséges mennyiséget. Ezt a *virtuális memóriával* oldja meg. A virtuális memória egy speciális terület a háttértáron. Windows esetében ez egy állományban található, Linux rendszerekben egy külön partíció, de lehet állomány is. A virtuális memória lényege, hogy a fizikai memória egyenlő méretű lapokra van osztva és a virtuális memóriában is ennek megfelelő lapok vannak kialakítva. Amennyiben több memóriára van szükség, mint amennyi fizikailag rendelkezésre áll, akkor a legrégebben használt lapok valamelyikének tartalmát az operációs rendszer kiírja a lemezre, majd a felszabadult helyet használja a memóriában. Ha valamilyen program olyan memóriacímre hivatkozik, amelyik pillanatnyilag a lemezen félretett valamelyik lapon található, akkor az operációs rendszer ismét megkeresi a legrégebben használt lapot a memóriában és kicseréli arra a lemezen levő lapra.

A fenti eljárás alapján a virtuális memóriát tartalmazó állományt szokás *lapozómemóriának* is nevezni. Az eredeti angol neve *swap*, amely a lapoknak a memória és a háttértár közti cseréjére utal.

A virtuális memória működését mutatja a 8. ábra. Mint látható, van egy táblázata az operációs rendszernek – a memóriában – amelyben minden elvileg megcímezhető memórialap reprezentálható. Ebben a laphoz elérhető az az információ, hogy a lap fizikailag éppen hol található, ezt jelzik az ábrán a nyilak. Ha egy program olyan lapon található címre hivatkozik, amely éppen a memóriában van, akkor nincs semmi gond. Ha viszont a háttértáron, akkor a fentebb leírt eljárás keretében az operációs rendszernek azt előbb be kell juttatnia a memóriába és csak aztán folytathatja a futását a

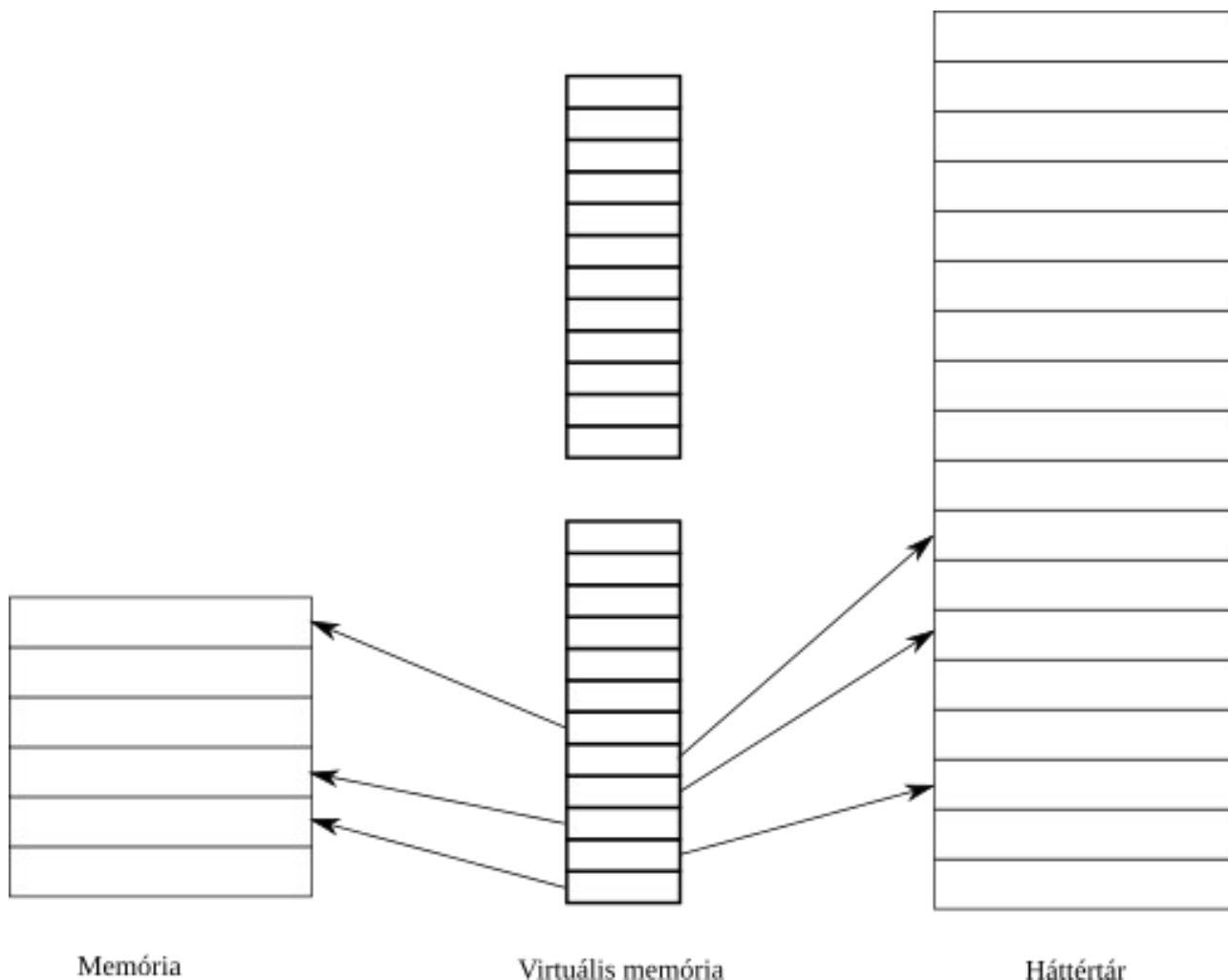
program. A virtuális memória táblázatának mérete függ a lapmérettől – amely meghatározza azt is, hogy egyetlen memóriacímért mekkora adatmennyiséget kell esetleg mozgatni – és a címezhető memória méretétől. Ez utóbbi a 32 bites processzoroknál 4 Gbyte volt, a számítógép tehát még virtuális memóriakezeléssel sem tudott (összesen) ennél nagyobb memóriát igénylő programokat futtatni. A jelenlegi korlát ennél jóval nagyobb a 64 bites processzoroknál, így jelenleg inkább az operációs rendszerek szabják meg a határt, aminél nagyobb memóriát semmiképpen sem képesek használni.

3.1.4. Adatok kezelése, átvitele

Ez is részben memóriakezelés. Másrészt a háttértárak kezelése is ide tartozik. Más néven ez a feladat a programok, adatok biztonságos tárolását jelenti. Korábban már volt szó említés szintjén az állományrendszerről. A háttértárakon az állományok tárolását az állományrendszereken keresztül végzi az operációs rendszer. Minden operációs rendszer más állományrendszereket ismer. Ezek az állományrendszerek meghatározzák, hogy fizikailag hogyan történik az állományok elrendezése, milyen állományneveket lehet használni stb.

A fontosabb állományrendszerek rövid ismertetéssel a következők:

- **FAT: File Allocation Table** – Ezt az állományrendszert a DOS-tól örököltük. Ma már jó lenne elfelejteni, de például a pendrive-ok ezt az állományrendszert használják. A neve onnan van, hogy a lemezen (merevlemez esetén partíción) egy táblázatban tárolja, hogy melyik



8. ábra: A virtuális memória

szektor foglalt és hogy az abban levő tartalom melyik szektorban folytatódik. A könyvtárbejegyzés tartalmazza az állomány nevét, méretét és a kezdő szektorát. Ezt követően ha az állomány mérete miatt abba a szektorba nem fér bele, akkor a FAT táblázatból derül ki, hogy melyik szektorban folytatódik. Így ha az állomány folyamatosan növekedett, akkor előfordulhat, hogy minden szektor után a lemez olvasása/írása során egy távoli szektorra kell átugrani. Ez a segédprogramok kapcsán már említett töredezettség, amely nagyon lassítja a FAT típusú állományrendszerek használatát. Ráadásul a FAT táblázat mérete maximalizálja a szektorok számát, vagyis egy nagyobb méretű lemezen kis méretű állományokból nem fér el annyi, amennyi a lemez kapacitása alapján várható. Ezért született meg a FAT nagyobb kapacitású változata, a FAT16 és a FAT 32. Előbbi legfeljebb 4 GB méretű lehet – az Ubuntu telepítő viszont FAT16 formátumot igényel. Így ha 4GB-nál nagyobb méretű pendrive-ot akarunk Ubuntu telepítőként használni, akkor azon egy 4 GB méretű partíciót kell használnunk. A további részét ugyan használhatjuk másik partícióként, de azt a Windows már nem fogja látni – de a Linuxok igen.

A FAT állományrendszeren tárolt állományok csak 8 karakteres névvel és még egy 3 karakteres kiterjesztéssel rendelkezhetnek. Innen ered, hogy a legtöbb állománykiterjesztés három karakteres. Később a Microsoft azt a megoldást találta ki, hogy ha ennél hosszabb egy állomány neve, akkor az két könyvtárbejegyzést foglal el. Ebből az első az állomány nevének 8+3 karakterre rövidített változatával a szokásos adatokat tartalmazza, míg a második a hosszú nevet.

- **NTFS:** A Windows NT-hez kifejlesztett állományrendszer, amelyet a FAT kiváltására terveztek. Itt már akár 250 karakteres, ékezetes állománynevek is tárolhatóak. Ráadásul a töredezettségmentesítés problémáját is megoldották azzal, hogy az állományokat nem feltétlenül egymás után helyezik el, hanem hagynak távolságot azok esetleges növekedéséhez. A nagy kapacitású külső merevlemezeken erre az állományrendszerre formázva kaphatóak.
- **HPFS:** Az IBM fejlesztette ki az OS/2 számára *High Performance File System* néven. Érdekessége, hogy az állományokat az üres terület közepére helyezte, így nagyon kis valószínűsége volt annak, hogy egy állomány feldarabolódott azért, mert nem volt neki elég hely.
- **HFS és HFS+:** Az Apple által kifejlesztett állományrendszer. A Macintoshok használják.
- **ext, ext2, ext3, ext4:** Eredetileg az ext állományrendszert a UNIX rendszerek használták, majd a Linux átvette a továbbfejlesztett ext2 változatot. Jelenleg a legfejlettebb az ext4, amely már naplózást is ismer (*Journal*), amely bár több helyet foglal adminisztrációra, de nagy előnye, hogy ha szabálytalanul áll le valamiért a gép, akkor nem kell esetleg egy órát is várni a lemez ellenőrzésére, mert a naplóból a hibákat azonnal javítani tudja a rendszer.
- **ISO 9660:** Az optikai lemezeken használt állományrendszer.
- **UDF:** Az ISO 9660 állományrendszer továbbfejlesztése, amely már képes 2 GB-nál nagyobb méretű állományokat is kezelni.

Az állományrendszerek mellett még a partíció fogalmát is érdemes ismerni. Egy hajlékony mágneslemez még nem volt értelme több logikai lemezre osztani, de mint fentebb a FAT16 kapcsán már szerepelt, egy pendrive esetében néha szükséges lehet. Egy néhány TB kapacitású merevlemeznel pedig még valószínűbb, hogy több partícióra van osztva. Minden partíció saját állományrendszerrel rendelkezik. Hogy a különálló partíciókat hogyan lehet elérni, az az adott operációs rendszertől függ. Például a DOS és a Windows minden partíciót önálló betűjellel lát el és önálló könyvtárszerkezetként jeleníti meg. Ezzel szemben a Unix alapú rendszerek, mint a Linux vagy az OS X egyetlen egy könyvtárfába integrálja az összes partíciót. Linuxon például szokás a /home könyvtár tartalmát külön partícióra tenni, hogy ha a rendszert újra is kell telepíteni, a felhasználói állományokat ez ne érintse.

3.1.5. Párbeszédes kapcsolattartás a felhasználóval.

A modern operációs rendszerek esetében a felhasználó folyamatosan tud kommunikálni a géppel: utasításokat adni, a programok futási eredményét megnézni. Ezeket a rendszereket *interaktív* felhasználói felület jellemzi. Az interaktív azt jelenti, hogy a felhasználó cselekedetére az operációs rendszer – vagy a program – azonnal reagál.⁶

A felhasználói felület kétféle lehet: karakteres és grafikus.

A karakteres felület esetén a képernyőn csak karakterek jelennek meg. Minden utasítást parancsok formájában be kell gépelni. Ez bonyolultnak hangzik, de egy rendszeradminisztrátor nagyon szereti a karakteres felületet, mert a megfelelő parancsok kiadásával bármit pontosan meg lehet adni és gyors is.

A grafikus felület a mai kényelmes felhasználóknak készült. Az első működő grafikus felületű operációs rendszer az Apple Macintosh gépén futott 1986-ban. A géphez egért kell ehhez csatlakoztatni, hiszen a legtöbb művelet az egerrel kezdeményezhető. A grafikus felület a karakteresnél jelentősen nagyobb memóriát és processzoridőt igényel. Ezért is van az, hogy sok szervergépen egyáltalán nincs grafikus felület – inkább az alkalmazások kapják a memóriát és a processzoridőt. A grafikus felületet használó operációs rendszerek is biztosítanak általában valamilyen módon a karakteres felület elérésére, hiszen a grafikus felület valójában csak egy segédprogram az operációs rendszeren belül. Ilyen karakteres felület érhető el a Parancssor vagy a Terminálablak formájában, de Linuxon például teljesen karakteres módba is átválthatunk a CTRL+ALT+Fn (pl. F1) billentyűkkel, majd a terminálmódok között az ALT+Fn billentyűkkel válthatunk. A grafikus felület is vagy a 7-es vagy a 8-as terminálon található, így az ALT+F7 vagy az ALT+F8 vált rá vissza.

3.1.6. Működési zavarok kezelése, jelzése

Más néven: hibakezelés, hibajelzés.

A számítógép használata során előfordulhatnak különféle problémák. Például az egyik ilyen, amikor egy program olyan memóriacímre hivatkozik, amelynek a lapja éppen nincs a memóriában. Ezt észre sem vesszük, mert ezt a hibát az operációs rendszer kezelni tudja – ekkor áll neki a lapcsere a virtuális memória kezeléseként.

Van más olyan hiba is még, amelyet az operációs rendszer anélkül old meg, hogy azt a felhasználó tudomására hozná. Ám van rengeteg olyan hiba is, amelyet nem tud megoldani, vagy nem szabad megoldania, mert a felhasználó döntése szükséges. Ilyen például, amikor egy adathordozót nem tud olvasni. Ilyen például, amikor nem érzékeli az egeret. Lehet még hosszan sorolni...

3.2. Elterjedtebb operációs rendszerek

Az idők során elég sok operációs rendszert fejlesztettek ki. Az alábbiakban ezek közül a jelentősebbekről lesz szó. Lesz köztük olyan is, amelyet ma már nem használunk. De mindenekelőtt érdemes néhány szempontot sorra venni, amely alapján különbséget lehet tenni az operációs rendszerek jellemzésében.

⁶ *Elvileg! Kivétel ez alól a Windows 7 – Windows 10, amely képes teljesen figyelmen kívül hagyni a felhasználót, amikor éppen arra támadt kedve, hogy frissítéseket keressen a CPU idő 100%-ának felhasználásával, majd telepítse azokat, akár órákra használhatatlanná téve a számítógépet. De mint már volt róla szó, a Microsoft fejlesztői úgy tűnik fordítva gondolkodnak: Nem a számítógép van a felhasználóért, hanem a felhasználó a Windows-ért...*

- **Felhasználói felület** alapján megkülönböztetünk *karakteres és grafikus felületű* operációs rendszereket. Erről korábban már volt szó, mint ahogy arról is, hogy a grafikus felületű operációs rendszerek – még a Macintosh is – valójában egy olyan karakteres felületű operációs rendszer, amelynek van egy grafikus felhasználói felülete, amely ráépül az alaprendszerre. Így lehetőség van valamilyen módon a karakteres felülethez is hozzáférni és ott parancsokat kiadni. Ez a legtöbb esetben egy ablakon belül futó *parancsértelmező* – más terminológia szerint *terminálablak* – formájában szokott megvalósulni.
- **Megkülönböztetett felhasználók** alapján beszélhetünk egyfelhasználós és többfelhasználós operációs rendszerekről. A ma használatos operációs rendszerek majdnem mindegyike a többfelhasználós csoportba sorolható. A megkülönböztetés az alapján történik, hogy az operációs rendszer tárol-e a felhasználóról olyan adatokat, amelyek alapján lehetőség van arra, hogy a számítógépet különböző időben ugyan, de több felhasználó egymástól függetlenül használja. Ez minimum a felhasználói azonosító és jelszó használatát igényli. Az egyfelhasználós rendszereken nincs lehetőség felhasználói azonosító és jelszó használatára. Manapság ilyen rendszerek szinte már csak a célszámítógépekben fordulnak elő, mint például a digitális fényképezőgép szoftvere esetében.
- Az alapján is lehet különbséget tenni, hogy **egyszerre mennyi felhasználó** tudja a számítógépet használni. Elsőre szokatlannak tűnik a gondolat, hogy egy időben ugyanazt a számítógépet többen használják. De ez azért van, mert a személyi számítógépekben gondolkodunk. Am ha a nagyobb gépekre – például a szerverekre vagy a szuperszámítógépekre – gondolunk, akkor már nem is olyan furcsa gondolat, hogy a gépre egyszerű terminálokról vagy személyi számítógépekről egyszerre több felhasználó is be tud jelentkezni.
- **Egyszerre futtatható programok száma** alapján is két csoportot lehet megkülönböztetni. Az egyik csoport ismét elhanyagolható kisebbséggé vált és inkább csak a beágyazott rendszereknél maradt meg. Ezeknél a hardveren egyszerre csak egy program futhat. Ennek oka lehet maga az operációs rendszer, de lehet a hardver is olyan, amely képtelen biztosítani egynél több program futását. Ezt az esetet *monoprogramozásnak* is szokták nevezni. A másik eset, amely a ma használt operációs rendszerek zömére jellemző, hogy képes egyszerre több programot futtatni, tehát az ehhez szükséges feladatok elvégzésére alkalmas. Amikor egyszerre több program – látszólag – párhuzamosan fut, *multitask* néven ismerjük.
- **A feladatok szervezése** alapján megkülönböztetünk három operációs rendszer fajtát:
 - *Köteget rendszereknek* a végrehajtandó programokat egymás után át kell adni az adatokkal együtt, majd azt a számítógép egymás után végrehajtja, a végén szolgáltatva az eredményt. A felhasználó menet közben nem igazán tud közbeavatkozni, legfeljebb az egész végrehajtást megszakítva, ami után mindent előlről kell kezdeni. A kezdeti rendszerek így működtek. Manapság már kevesebb példa van ilyen megoldásra, de a szuperszámítógépeken azért, hogy a teljesítményüket teljesen ki lehessen használni, van értelme ilyen megoldásnak: az operációs rendszer folyamatosan fogad végrehajtandó programokat, amelyeket optimalizáltan eloszt a rendelkezésre álló CPU-k között. Természetesen köteget rendszerek csak az interaktivitást nem igénylő, elsősorban számítások elvégzését igénylő feladatokra alkalmasak.
 - *Valós idejű (realtime)* rendszereket olyan helyen használnak, ahol a beérkező jelekre azonnal kell reagálni. Ilyenek lehetnek például a repülőgépek irányítását végző fedélzeti számítógépen, egy atomerőmű vezérlő rendszerén. De ilyen egy automata vezérlésű autó vagy egy automata vezérlésű metróvonal is. Sőt, még a nem automata vezérlésű vasútvonalakon is a biztosítóberendezéseket lehet ilyen valós idejű rendszernek tekinteni, ahogy egy mesterséges intelligenciát használó forgalomirányító rendszert is.

- *Interaktív* rendszerek találhatók a személyi számítógépeken. Ezekről is elvárjuk, hogy amikor valamilyen utasítást adunk a gépnek, akkor az arra azonnal reagáljon. Viszont a valós idejű rendszerekkel ellentétben itt nem létfontosságú az azonnali válasz. Az interaktív rendszereket leginkább azzal jellemezhetjük, hogy folyamatos párbeszédet tesz lehetővé a felhasználó és a számítógép között.

Az alábbiakban nagyjából időrendi sorrendben a fenti szempontokra figyelve következnek a jelentősebbnek bizonyult operációs rendszerek bemutatása, részben egyfajta operációs rendszer történelemként is.

3.2.1. UNIX

Az 1960-as években már nagyon sok kutatóintézetben, egyetemen működött számítógép. Ám ezek sokfélék voltak, mindnek saját operációs rendszere volt, amelyek mind más felhasználói felülettel rendelkeztek. Amennyiben valaki az egyik intézetből a másikba került át, gyakorlatilag újra kellett a számítógép használatát tanulnia.

Ezen a helyzeten változtatott DENNIS RICHIE (1941–2011), aki munkatársaival korábbi nyelvek továbbfejlesztésével ekkoriban hozta létre az azóta hatalmas sikert befutó C programozási nyelvet. Amellett, hogy ez a programozási nyelv mára egy egész nyelvcsalád megalapozójává vált, a sikert az jelentette igazán, hogy a programozási nyelvhez gyakorlatilag az összes forgalomban levő számítógépre készült fordítóprogram. Ez tette először lehetővé, hogy az egyik számítógépen kifejlesztett programot bármelyik gépen használni lehessen.⁷ A programnyelv megalkotása után vagy vele egy időben Richie és Ken Thompson egy operációs rendszer fejlesztésével is foglalkozott, amelyet C nyelven írtak. Mivel a kód gyakorlatilag bármelyik számítógépen lefordítható volt, így keletkezett egy olyan operációs rendszer, amely gyakorlatilag bármely akkor használatban levő számítógépen működőképpé vált. Egységes felhasználói felületével megszüntette az addig, fentebb leírt problémát. Ez az operációs rendszer lett ezt követően a nagygépek egységes operációs rendszere, a UNIX.

A 9. ábrán felsorolt nevek még elő fognak kerülni, mivel mindegyiküknek köze van valamilyen operációs rendszer létrejöttéhez. Ráadásul több esetben olyan operációs rendszeréhez, amely nagyon sokat köszönhet a UNIX-nak.

Ennek pedig az az oka, hogy a UNIX egy idő után szabványosodott. Létrejött a POSIX szabvány, amely leírja azokat a követelményeket, amelynek az operációs rendszer meg kell, hogy feleljen: a felhasználói felület mellett az egyes programok által használható interface, az API elemeinek definícióját is. Amit manapság UNIX-alapú operációs rendszernek nevezünk, az valójában ennek a POSIX szabványnak megfelelő operációs rendszer.

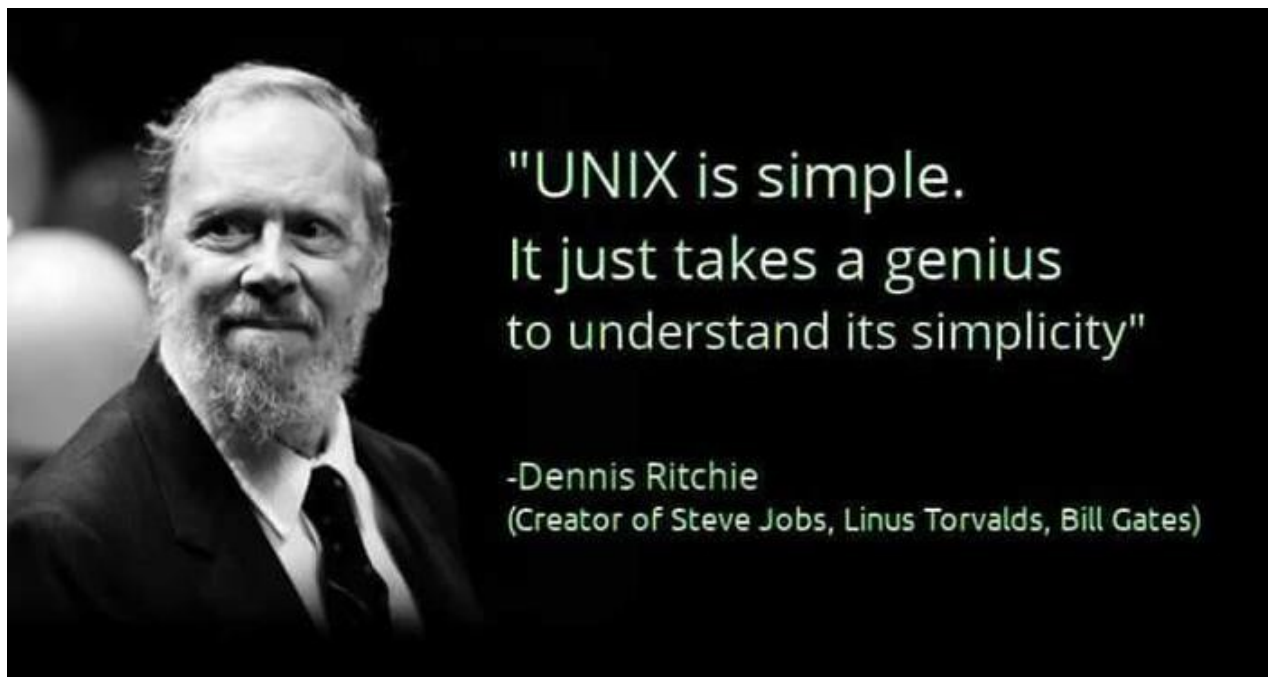
Hogy mennyire nagy hatással volt az operációs rendszerek fejlődésére a UNIX, azt a 10. ábra nagyon jól mutatja. Az ábra eredetije – jóval nagyobb méretben, olvashatóbban – a <https://en.wikipedia.org/wiki/Unix> címen megtalálható.

Persze a UNIX nem csak az operációs rendszerek fejlődését alapozta meg. Az USA védelmi minisztériuma létrehozta az ARPA (*Advanced Research Projects Agency*) intézményeiben működő számítógépek közötti hálózatot, ez lett az ARPANET. Ezek a számítógépek mind UNIX-ot használtak, így a hálózat működéséhez szükséges szolgáltatásokat UNIX alatt kellett kifejleszteni, így a TCP és az UDP protokollok is a hálózati kommunikációra. Itt jelent meg először például az email és az FTP szolgáltatás is.

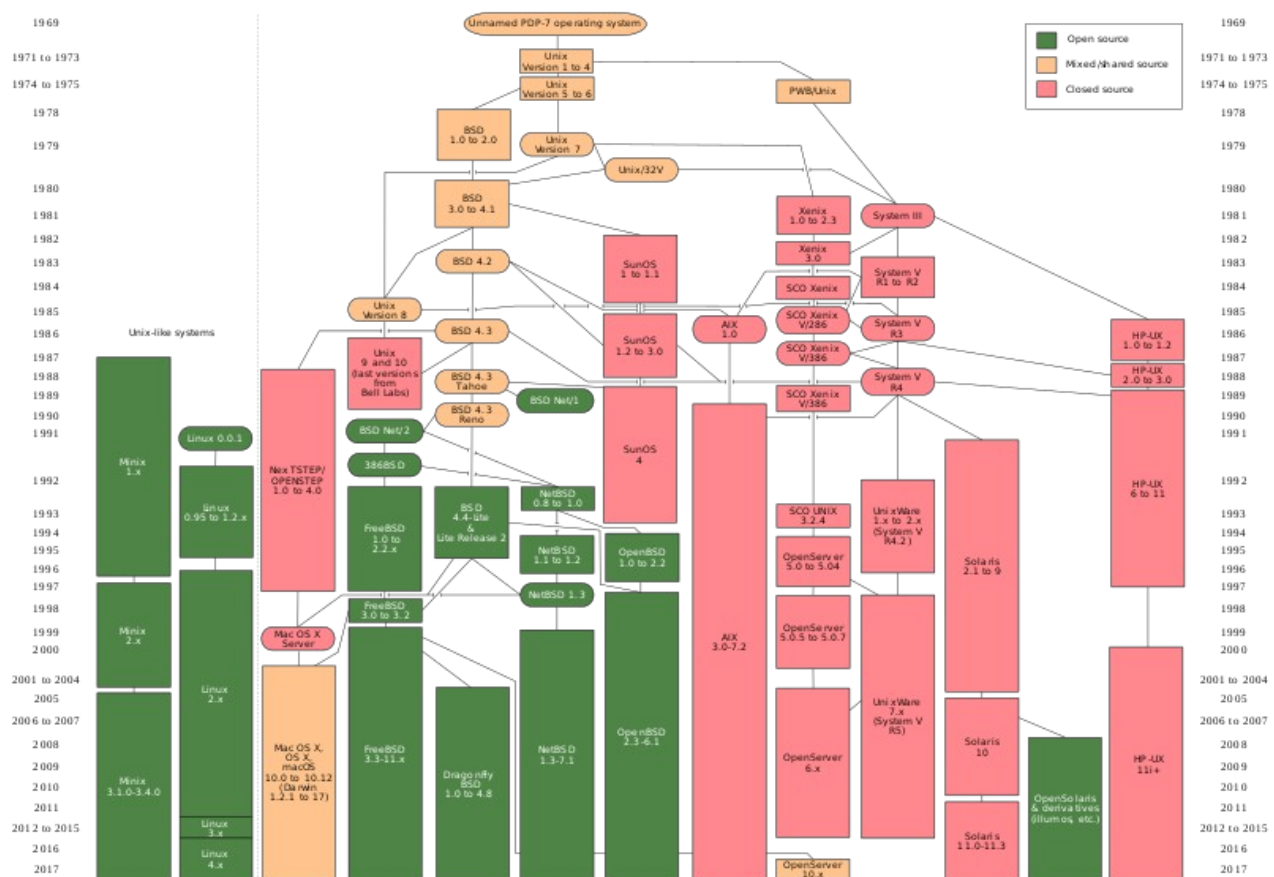
Majd az ARPANET előbb az Egyesült Államokban vált bárki számára elérhetővé, aztán az egész világon. Ezzel megszületett az Internet, amelynek tehát a legtöbb alapvető funkcióját UNIX környezetben fejlesztették ki.

⁷ A nyílt forráskódú programok a mai napig szinte kivétel nélkül a C nyelven vagy valamelyik leszármazottján íródnak, ezért lehet ezeket bármelyik ma használt operációs rendszerre lefordítani.

Maga az operációs rendszer – mivel nagygépen fut – egyszerre sok felhasználót tud kiszolgálni. Ebből következően több program futtatására is alkalmas (multitask). Az eredeti UNIX (és a PO-



9. ábra: Dennis Ritchie mondása: A UNIX egyszerű, csak egy zseninek kell lenni az egyszerűsége megértéséhez.



10. ábra: A Unix és utódai története

SIX szabvány is) karakteres felületet adott, de a manapság használatos POSIX szabványnak megfelelő operációs rendszerek már grafikus felülettel is használhatóak.

3.2.2. Disc Operating System = DOS

Az IBM 1980-ban dobta piacra a személyi számítógépét, amit egyszerűen csak Personal Computer, azaz PC névvel látott el (IBM PC). A gépet úgy tervezték, hogy minél olcsóbb legyen. Ezért a kereskedelmi forgalomban elérhető alkatrészekből rakták össze. Ezzel az volt a céljuk, hogy a többi számítógépgyártó cég tudjon hozzá kiegészítőket gyártani. Emiatt a gép specifikációját is nyilvánosságra hozták. Ennek viszont az lett a következménye, hogy elég gyorsan jelentek meg klónok, amelyek ugyanazt a gépet még olcsóbban kínálták. Emiatt egy idő múlva az IBM fel is hagyott a személyi számítógépek gyártásával, de ott még nem tartunk időben.

A géphez operációs rendszer is kellett. Ezügyben az IBM Bill Gates frissen alakuló cégét, a Microsoftot kereste meg. A Bill Gates nem sokkal korábban vásárolt meg egy operációs rendszert, amit addig még nem igazán használt senki arra alkalmas hardver hiányában – tehát inkább csak egy szoftvertervezetnek tekinthető. Ez az operációs rendszer a UNIX mintáját követte, de az akkor még jóval kisebb tudású személyi számítógépek számára. Bill Gates cége ebből a programból némi módosítással alkotta meg az IBM PC operációs rendszerét, amelynek a DOS nevet adták, mert elsősorban a mágneslemez kezelésére és az azon található programok memóriába töltésére és futtatására volt tervezve (DOS = Disc Operating System).

A DOS célja az volt, hogy a UNIX-szal azonos felhasználói felületet nyújtson az IBM PC-n. Tudni kell, hogy az akkoriban elérhető személyi számítógépekre többen próbálkoztak a UNIX-hoz hasonló operációs rendszer megalkotásával. A DOS ebben odáig eljutott, hogy a karakteres felületén a UNIX alatt elérhetőhöz eléggé hasonló parancsok voltak elérhetőek. Érdekes módon azért több olyan módosítással, amely inkább távolította a UNIX-tól (mint az állományok útvonalában a könyvtárak elválasztására szolgáló / jel megfordítása \ jelre).



11. ábra: Bill Gates

Ám a hardver túl gyenge volt ahhoz, hogy egy nagygépekre tervezett operációs rendszer megvalósítható legyen rajta. Így nem volt képes több program egyidejű futtatására sem. Emiatt a DOS egyfelhasználós, monoprogramozott rendszer volt.

Azután az Intel újabb processzorai, amelyek a PC-be kerültek már alkalmassá váltak a multitask megvalósítására, de a DOS ezt nem követte. Furcsa módon a multitaskot elengedhetetlenül igénylő grafikus felülettel is egy ilyen egyprogramos rendszeren próbálkozott először a Microsoft, amiről még lesz szó.

Miközben az IBM számára összességében nagy buktának bizonyult a PC, a Microsoft nagyot nyert anyagilag rajta, mivel a két cég közötti megállapodás megengedte, hogy a DOS más számítógépeken is használható legyen, így miközben az IBM gépeit alig vásárolták, a klónokra vásárolt DOS nagyon jól fogyott.

3.2.3. Apple Macintosh

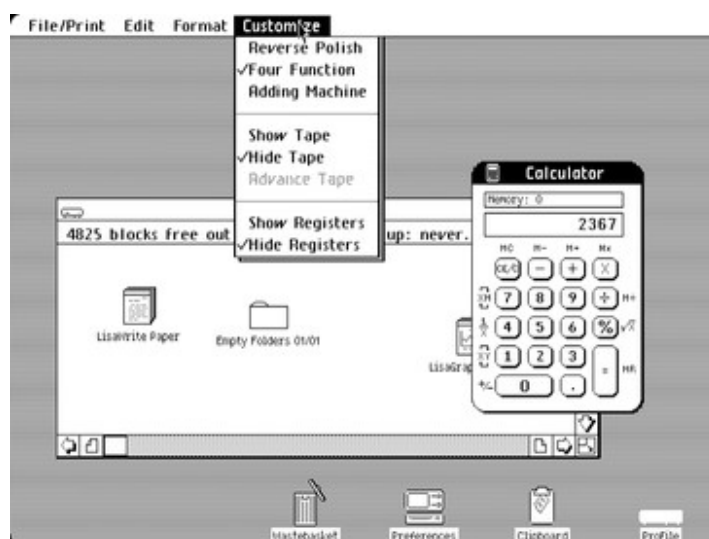
Steve Jobs cége, az Apple közben szintén piacra dobott egy személyi számítógépet, amelynek nem volt akkora sikere, mint az IBM PC-nek – amelynek utódai mára szinte egyeduralkodóvá váltak. Ez a gép is egy karakteres felületű operációs rendszert használt. Ez volt az Apple első számítógépe.

Ezt követően a Xerox grafikus felületével való kísérleteit látva Jobs úgy döntött, az Apple-nek készítenie kell egy olyan számítógépet, amely ilyen grafikus felületet használ. Ez az 1983. január 19-én megjelent gép volt a Jobs lányáról elnevezett Lisa, amelynek operációs rendszere volt az első grafikus felületű operációs rendszer.

Ennek a gépnek ugyan nem volt sikere, de néhány továbbfejlesztés révén 1986-ban megjelent az azóta nagy sikert elért Macintosh gépek első néhány verziója. Ennek operációs rendszerét egyszerűen „Rendszer” névvel látták el: System. Később Jobs távozni kényszerült a saját cégétől, amely hamarosan a csőd szélére került, de Jobs közben a Next Computer nevű cégnél egy UNIX alapokra épített grafikus felületű rendszert fejlesztett, amely a Macintosh gépek operációs rendszere lett, miután visszatért az Apple élére. Ez volt a System 8, majd mára elérkezett a System 10-hez, amit először Mac OS X, majd rövidebben OS X néven ismerünk.



12. Ábra: Steve Jobs



13. Ábra: A Lisa operációs rendszere a Lisa OS

Az OS X egy POSIX szabványnak megfelelő (tehát UNIX alapú) operációs rendszer, kezdettől grafikus felülettel, ebből következően több program egyidejű futtatására képes. Mostanra nagyon fejlett rendszer alakult ki, amely még mindig mintának számít a legtöbb szoftvercég számára.

3.2.4. IBM: OS/2

Ahogy az Intel egyre többet tudó processzorokat készített, az IBM PC-n a DOS képességei egyre nagyobb hátrányt jelentettek. Ezért az IBM új operációs rendszert akar a gépekre – ezúttal is a Microsoft közreműködésével. Egyben hogy visszaszerezze a piacot, amit a PC klónok miatt egyre jobban kezdett elveszíteni, elkészítette 1987-ben a PC továbbfejlesztett változatát az IBM PS/2 (Personal System 2) gépet. A gép számos technikai újdonságot hozott ugyan, de a PC klónokra ráált hardvergyártók nem igazán támogatták az új, sokkal megkötöttebben elérhető rendszert, így a PS/2 gép nem volt túl hosszú életű. De megmaradt utána néhány dolog. Ilyen a csak mostanában USB-portra cserélődő, de a PS/2 gépeknél megjelent billentyűzet és egér csatlakozó, amelyeket szintén PS/2-portként ismerünk.

A PS/2-höz készült operációs rendszer az OS/2 (Operating System 2) nevet kapta. Az első változat is már grafikus felületű volt az Apple gépeinek mintájára. Azonban mivel a gép, amelyre készült, nem lett túl népszerű, nem terjedt el a szoftver sem túlzottan.

Az operációs rendszer további változatai éppen ezért – már a Microsoft nélkül – az eredeti IBM PC újabb változatain voltak elérhetőek. Saját tapasztalat: nagyon megbízható, stabil operációs rendszer volt, ami igazán megérdemelte volna, hogy szélesebb körben elterjedjen.

Az operációs rendszer eleinte egyfelhasználós volt, később megjelent többfelhasználós változata is.

Grafikus felületet használt, az első olyan operációs rendszer volt a PC-re, amely valóban képes volt egyszerre több program futtatására is. Sőt, igazából a virtuális memóriakezelés is itt jelent meg először a PC-n.

Az OS/2 fejlesztése közben összekülönbözött egymással az IBM és a Microsoft, így az OS/2 3.0 Warpot az IBM már egyedül fejlesztette és jelentette meg 1995-ben. A Microsoft a DOS-t fejlesztette tovább MS DOS néven egészen a 6.22 verzióig, amely után szintén 1995-ben új operációs rendszert mutatott be. Az OS/2 ezután sem vált túl népszerűvé, pedig kifejezetten jó minőségű operációs rendszer volt, ráadásul a Warp volt az első 32 bites operációs rendszer PC-re.

3.2.5. Microsoft: Windows

Az Apple grafikus operációs rendszere mintájára a Microsoft elkezdett egy grafikus felületet fejleszteni még az OS/2 előtt – valószínűleg ennek volt a legnagyobb szerepe az IBM és a Microsoft közötti vitának –, amely a DOS-ra épült egy külön programként. Ezt úgy kell elképzelni, hogy a `win` DOS parancsot kellett kiadni az elindításához. Ekkor elindult a `win.exe` nevű program, amely révén egy grafikus felület jelent meg. Ez volt a Windows 1.0 1985-ban. Ám mivel ez DOS-ra épült és a DOS monoprogramozású volt – annyira, hogy még az operációs rendszer és a program sem tudott egyszerre futni rajta, így a grafikus felület csak látszólag volt képes több programot futtatni egyszerre. Így aztán ez a program nem is aratott nagy sikert.

Ám a Microsoft nem adta fel, hanem továbbfejlesztgette, egészen a Windows 3.1 verzióig, amely már-már működőképes kezdett lenni. Ám ekkorra be kellett látnia a Microsoftnak, hogy a működőképesség legfőbb akadálya maga a DOS. Igaz, hogy adódtak módok, amelyek meg lehetett kerülni a DOS-t és elérhetővé vált a hardver által már biztosított multitask lehetősége – ahogyan az OS/2 már 1987-től azt kihasználhatta –, de elég körülményesen.

Ezért pár hónappal az OS/2 3.0 Warp megjelenése után, szintén 1995-ben megjelent a Windows 4. Ez már nem a DOS-ra épült, hanem önálló operációs rendszerként működött. Ám nem ezen a néven ismertük meg, hanem Windows 95 néven. Ám amennyiben valaki egy parancsértelmezőt indított rajta, akkor ott a DOS-ból ismert parancsok voltak elérhetőek. De a `ver` parancs a korábbiakkal ellentétben, amikor például az `MS DOS 3.5` jelent meg – vagy az utolsó verziójánál az `MS DOS 6.22` –, az operációs rendszer belső nevét jelenítette meg, amely `Windows 4` volt. Ezt követte három évvel később a Windows 98. Ezután a fejlesztés kétfelé bomlott és a Microsoft egyszerűre kezdett szerverek felé és személyi számítógépek felé nyitni. A nagy teljesítményű felhasználásra szánták a Network Technologies rövidítéséből Windows NT-nek nevezett változatot, míg személyi felhasználásra 2000-ben megjelent a nem túl nagy sikert megért Windows Millennium, majd a még sok helyen ma is használt Windows XP, amely sokak szerint máig a legjobban megalkotott Windows.

Az eddig felsorolt Windows operációs rendszerek mind 32 bitesek voltak. Ezután a Microsoft elkezdte a 64 bites rendszerre átállást, amelyről az XP már sajnos lemaradt. Sajnos, mert ha lenne az XP-nek 64 bites változata, alighanem sokan még ma sem lennének hajlandóak újabb verzióra váltani.

Szintén nem túl sikeresnek bizonyult a Windows Vista, amely az eredeti verziószámozás szerint a 6. verzió egy alverziója lenne. Ezután elég hamar próbálta a Microsoft javítani a Vista hibáit és piacra dobta a Windows 7-et, amely az XP után a második legjobban megalkotott Windows verzióknak bizonyult. Ezután sokak szerint hanyatlás indult: a Windows 8 ismét nem aratott sikert, így elég gyorsan javítani kellett, amit a Windows 8.1 megjelenése jelentett.

Az amerikaiakra jellemző babonásságnak lehet nagy szerepe abban, hogy látva a tendenciát, hogy minden második verzió sikerül csak működőképesre, a Microsoft úgy döntött, hogy átugorja az „átok” miatt várhatóan ismét sikertelen 9. verziót és rögtön a Windows 10-re lép tovább. Ám az átok úgy tűnik, így is hatott, mert a Windows 10 elég problémásra sikerült, így egyre többen sirják vissza a Windows 7-et vagy akár a Windows XP-t is.

Amióta a Windows nem segédprogram a DOS felett, hanem önálló operációs rendszer, azóta elmondható róla valóban a multitask-képesség, a többfelhasználós mód.

3.2.6. Bezárult a kör: UNIX a PC-n

1990-ben egy svéd nemzetiségű finn egyetemista, név szerint LINUS TORVALDS az egyik egyetemi feladata során – amit történetesen Andrew Tanenbaum professzortól kapott – kicsit többet tett annál, mint amit kellett volna. Az operációs rendszerek elméletét tanulta Tanenbaum professzornál. A program, amit alkotott egy POSIX szabványú operációs rendszer alapját jelentette. A művét – amely még távolról sem nevezhető operációs rendszernek, csupán annak alapvető elemeit valósította meg – 1990. augusztusában publikálta a neten.

Ebből a programból azután a nyílt forráskód lehetőségét kihasználva továbbfejlődött és létrejött belőle, először elég kezdetleges képességekkel a PC-n elérhető első teljesen POSIX szabványnak megfelelő operációs rendszer, a Linux.

Linus Torvalds egyébkén a Linux alapját képező rendszermag, a kernel fejlesztésének koordinálásában mind a mai napig részt vesz, de maga nem ragaszkodik mindenáron nyílt forráskódú szoftverek használatához. Ha a számára szükséges célra kereskedelmi szoftver a legmegfelelőbb, akkor azt használja – vallja.

A Szabad Szoftver Alapítvány által kezdeményezett GNU projektnek normálisan használható operációs rendszere nem volt, így inkább befogadták a Linuxot és annak fejlesztését segítik. Így az operációs rendszer hivatalos neve: GNU/Linux.

Az operációs rendszer gyakorlatilag UNIX rendszernek tekinthető, annak minden következményével: eredeti változatában karakteres felületű volt, multitask és egyszerre több felhasználó munkáját lehetővé tevő.

Mostanra több változata terjedt el, amelyeket disztribúció néven emlegetünk. Ilyen disztribúciók közül a legismertebbek a Red Hat, a Debian vagy a Debianra épülő Ubuntu.

A Linuxhoz – és rajta keresztül a többi UNIX-rendszerhez – mára rengeteg grafikus felület született, amelyek közül a fejlettebbekkel a Linux mára méltó versenytársává vált nem csak a Windowsnak, de a Macintoshoknak is.

3.2.7. Mobil operációs rendszerek

Ahogy az okostelefonok, tabletek terjednek, úgy növekszik a fontossága az ezeken futó operációs rendszereknek.

Korábban többféle rendszer létezett, de ezek mára egymás után kikoptak a használatból. Ami mára megmaradt, az a személyi számítógépeknél használt operációs rendszerek mobil eszközökre átalakított változata – persze erős túlzással. Három ilyen operációs rendszer használatos manapság:

- Windows phone, később Windows 8, 8.1, 10 mobil változata. Ez utóbbiak valójában a Microsoft állítása szerint teljesen azonos a személyi számítógépeken elérhető változattal. Ebből persze adódnak problémák mind a személyi számítógépen, mind a mobil eszközökön, hiszen utóbbin az érintőképernyő a fő beviteli eszköz, míg az előbbieknél ez általában nem is elérhető.
- A Linux kernelét felhasználva a Google létrehozta a saját mobil operációs rendszerét, az Androidot, amely manapság a mobil eszköz gyártók zömének készülékein megtalálható – gyakorlatilag az Apple eszközeit és a Windows-t futtató gépeket kivéve mindegyiken.
- Az Apple is a személyi számítógépén használt operációs rendszerből indult ki: a Mac OS X alapjára, de a felhasználói felületet az eltérő hardvernek megfelelőre lecserélve hozta létre az iOS-t. Egy időben az Apple is azt tervezte, hogy teljesen egyesíti az OS X és iOS rend-



14. Ábra: Linus Torvalds és az 1990. augusztusában publikált levele egy (nyilván későbbi) Linux rendszerben

szereket egyetlen operációs rendszerré – így csak egy fejlesztőcsapatra lenne szüksége –, ám ez éppen a kétféle hardver és a teljesítmény igény közti rendkívül nagy eltérés miatt nem történt meg.

4. Számítógépes kártevők

A számítógépen futó programok nem feltétlenül hasznos célra vannak. Sajnos az élet többi területéhez hasonlóan a számítógépeken is vannak, akik gonosz szándékkal rendelkeznek. Többféle módon, többféle céllal próbálják megkéséríteni a felhasználók életét. Néha merő rosszakaratból, néha gazdasági bűncselekmények céljából, néha pedig kifejezetten káros politikai célzattal.

Így a számítógépes kártevőknek is elég széles skálája alakult ki mára. Legendák szerint az első számítógépes vírust egy a munkahelyéről kirúgott programozó készítette azzal a szándékkal, hogy volt munkaadóján bosszút álljon, de a bosszú túl jól sikerült. Más legenda szerint ellenben a kereskedelmi szoftverek felelősek a vírusok megjelenéséért, mert olyan másolásvédelmi módszerekkel próbálták védeni termékeiket, amelyek káros tevékenységet – elsősorban az állományok törlését – végeztek el az eredeti telepítő lemez másolatán. Aztán a legenda szerint ezek a másolásvédelmi programok szabadultak el.

Ma már egészen biztosan szándékosan készülnek a kártevők, amelyek általában a számítógépes rendszereink hiányos védelmi pontjait támadják – ezért vannak a szoftvereken, főleg operációs rendszereken a rendszeres frissítések.

4.1. A kártevők fajtái

4.1.1. Önszaporító vírusok

A klasszikusnak nevezhető vírusfajta a nevét onnan kapta, hogy a biológiai vírusokhoz hasonlóan valamilyen gazdaszervezetet megfertőzve, azt készíti saját kódja sokszorosítására. Ezúttal a gazdaszervezet maga a programkód, amelyhez hozzáírja magát a vírus. A sokszorosítandó kód pedig nem RNS vagy DNS, hanem maga a vírus.

Amikor egy fertőzött programot elindítunk, a vírus kódja is azonnal a memóriába kerül. Itt két dolgot fog elkövetni: egyrészt keres a háttértárakon még megfertőzhető programokat. Ha talál, akkor annak kódját is módosítja. Másrészt pedig valamilyen tevékenységet is kifejt, amivel a gép használatát akadályozza. Például töröl adatállományokat. Vagy a memória egy részét lefoglalja magának, így lassítva a gép működését.

4.1.2. Programférgek

A programféreg vagy *worm* általában a nem megfelelően védett internetkapcsolaton keresztül jut be a számítógépre, ahol azután a háttérben tevékenykedni kezd. Általában ha az első komponens megérkezett, akkor az a háttérben futva további állományokat, más kártevőket is letölt a gépre.

Általában a férgek nem ártanak közvetlenül a számítógépnek, helyette bekapcsolják azt egy úgynevezett *botnet*-be, amelyet valaki a távolról tud irányítani. Ezzel az akár több százezer gépre nőtt, a világ minden részén jelenlevő hálózattal azután a gépek tulajdonosának tudta nélkül képes a botnet gazdája spam-eket elindítani, szerverek ellen túlterheléses támadást indítani, vagy feltörni úgy, hogy a személye rejtve marad, hiszen nem a saját gépe, hanem gyanútlan emberek tulajdonában levő gépek végzik el a piszkos munkát.

4.1.3. Kémprogramok

A kémprogramok (*spyware*) olyan programok, amelyek a háttérben figyelik, hogy mit csinál a felhasználó a számítógépen. Különösen az internetes bejelentkezéseket figyelik, hogy megszerezzék a felhasználó belépési adatait. Ha ez sikerül, akkor elküldik a kémprogram készítőjének, aki ezáltal illetéktelenül fel tudja azokat használni.

Egyszerű felhasználóként talán nem látszik ez túl veszélyesnek. De valójában itt nem csak arról van szó, hogy valakinek a facebook jelszavával vissza lehet élni. Az internetes bankok világában ennél jövedelmezőbb helyekre is meg lehet szerezni a belépési adatokat. De ennél még veszélyesebb, ha valaki egy fertőzött gépről olyan helyre jelentkezik be, ahol titkos információk találhatók.

4.1.4. Trójai programok

Ezek úgy kerülnek a gépünkre, hogy elővigyázatlanul mi magunk töltjük le. Valami hasznos programnak hirdeti egy weboldal, pedig valójában nem is az. Vagy lehet, hogy az, de a kódjára rátelepült egy féreg vagy egy kémprogram. Pont ezért hívják trójai programnak.

4.1.5. E-mail vírusok

A levelezőprogramon keresztül terjedő vírusfajta is veszélyes tud lenni. Nem másról van szó, mint hogy valakitől kapunk egy levelet, amelynek csatolmánya egy vírust tartalmaz. A levél nem csak idegentől érkezhet, hanem ismerőstől is, ugyanis a vírus a levelezőprogramban tárolt valamennyi levélben szereplő e-mail címet felkutatja és a tudtunk nélkül elküldi magát azokra.

4.1.6. Halászó oldalak

Tipikus módja mások belépési adatai megszerzésének, ha egy webes szolgáltatáshoz nagyon hasonló oldalt készítenek, majd szétküldenek hamis e-maileket, amellyel ráveszik az embereket, hogy ezen az oldalon jelentkezzenek be. Például egy bank oldalát másoló oldalra irányítja a levél a címezett azzal a szöveggel, hogy valaki megkísérelte feltörni a fiókját, legyen szíves bejelentkezni és sürgősen megváltoztatni a jelszavát. Valójában senki nem törte fel a fiókot – még. Ám ha a felhasználó ténylegesen megpróbál bejelentkezni, akkor maga adja meg a bűnözőknek a bankszámlája elektronikus hozzáféréséhez szükséges adatokat.

4.1.7. Hoax, álhír

Manapság egyre terjed a hamis híreket közlő internetes oldal. Ezek olyan oldalak, amelyek címe szenzációra utal. Ezeknek a facebook kitűnő terep, hiszen a megosztásokból csak egy cím, egy kép és az oldalt tartalmazó szerver címe látszik. Sokan ez utóbbit meg sem nézik, pedig nagyon sok ilyen oldalról már tudható, hogy ilyen álhír-oldal.

Az oldal tele van hirdetésekkel és jó esetben van benne valahol valami rövid szöveg is az ígért témában. De régebben volt olyan is, hogy egy külön gombra kellett a tartalom megtekintéséhez kattintani. Amely gombra kattintva automatikusan megosztotta a felhasználó a tartalmat, majd ezután láthatta esetleg a nyúl farknyi vagy hosszabb írást.

Kevésbé kifinomult, de szintén működő megoldás az álhír terjesztésére valóban létező hírportálhoz hasonló nevű portált létrehozni és ott forrásmegjelölés nélkül cikkeket megosztani.

Az ilyen álhírek ellen lenne a legegyszerűbb a védekezés, de sokan mégsem teszik. Egyszerűen ellenőrizni kell a leírtak forrását, van-e szerzője az írásnak, az illetőnek lehet-e más írását megtalálni. Valamint meg kell nézni, hogy van-e más, megbízható oldal is, amely megosztja a hírt.

4.2. A kártevők elkerülése

Elsősorban egy megfelelő víruskereső/vírusirtó programmal, illetve tűzfallal tudunk védekezni a gépünkre bejutni próbáló kártevőkkel szemben. Ezen felül fontos, hogy ezt a programot használjuk és rendszeresen frissítsük, hogy az újonnan keletkezett kártevők is belekerüljenek az adatbázisába.

Amennyiben az internetről akarunk letölteni bármit, ellenőrizzük, hogy az oldal megbízható-e. Ha igen, akkor is a letöltött állományt megnyitás előtt vírusellenőrzésnek kell alávetni. Emiatt célszerű minden letöltött állományt mindig ugyanabba a könyvtárba (Letöltések) letölteni, mert így a könyvtár lehet ellenőrizni.

Amennyiben olyan adathordozót csatlakoztatunk a számítógéphez, amit előzőleg egy másik számítógépen használtak, akkor mielőtt arról bármit megnyitnánk, futtassuk le rajta a vírusellenőrzést. Az optikai lemezek, pendrive-ok esetében előfordulhat, hogy van egy autorun.exe nevű állomány, amelyet az operációs rendszer azonnal futtat. Ezzel máris vírus kerülhet a rendszerünkbe, főleg mert több olyan kártevő is van, amely ezt a lehetőséget használja ki. Éppen ezért bármilyen kényelmesnek tűnik is ez a funkció, kapcsoljuk ki!

Előfordulhat, hogy a gép váratlanul – például egy internetes kép megnyitásakor – kéri a rendszergazda jelszavát. Ennek az az oka, hogy például a képbe egy vírus kódja van elrejtve, amely szeretne hozzáférni a rendszerhez. Emiatt csak abban az esetben adjuk meg a rendszeradminisztrátori jelszót, ha tudjuk, miért kéri a gép. Persze ha eleve rendszeradminisztrátori joggal vagyunk bejelentkezve, akkor az ilyen támadás nem fog feltűnni. Ezért fontos, hogy a rendszeradminisztrátori fiókot csak akkor használjuk, amikor feltétlenül szükséges. Szerencsére a Unix alapú rendszerek, mint az OS X és a Linux nem használhatóak állandóan rendszergazdaként...

Az e-mail vírusok ellen úgy lehet védekezni, ha figyelünk arra, milyen levelet nyitunk meg. Ha ismeretlen személy küld csatolt állományt, a levelet azonnal célszerű törölni. Ha ismerősünk küld tőle szokatlan tartalmú levelet – más nyelven ír, vagy más stílusban, mint tőle várnánk –, akkor az is lehet vírus!

A halászó oldalak ellen talán a legkönnyebb védekezni: oda kell figyelni, hogy mindenhova csak a hivatalos bejelentkező oldalon lépünk be. Fontos tudni, hogy a bankok soha nem küldenek e-mailben felszólítást a számlánk ellenőrzésére! Ha küldenek ilyet, akkor azt postai úton teszik. Ha pedig mégis elektronikusan keresnek meg, akkor sincs a levélben link a bejelentkezéshez, nekünk magunknak kell a hivatalos címet felkeresnünk.

4.3. Miről ismerjük fel a bekövetkezett fertőzést?

Nyilván ez nagyon függ a kártevőtől, de van néhány tipikus tünete annak, hogy baj van:

- A gép szokatlanul lassú.
- A memória vagy a háttértár szabad területe indokolatlanul lecsökkent vagy megnőtt.
- Ismeretlen nevű állományok lelhetők fel a háttértáron.
- A program bizonyos funkciói elérhetetlenné válnak.
- Egyes perifériák hibásan működnek vagy elérhetetlenné válnak.
- A gép időnként ok nélkül újraindul.

4.4. Mit tegyünk fertőzés esetén?

Először is azonnal indítsuk újra a számítógépet, hogy minden fölösleges tartalom eltűnjön a memóriából. Sőt, mivel már az operációs rendszer elindulásakor bekerülhet a kártevő a memóriába, ezért a hagyományos indítás helyett más megoldást kell keresnünk. Ez lehet egy másik indítólemezről indulás – ami írásvédett – vagy a Windows Csökkentett módja, amelyben csak a minimális szolgáltatások indulnak el. A biztonság kedvéért kapcsoljuk le a gépet a hálózatról is!

Ezután pedig következhet a teljes és tüzetes átvizsgálása a gépnek. Minden adattárolót, ami a gépre volt csatlakoztatva a fertőzés feltételezett időpontja óta a víruskeresővel végig kell nézni. Jobb víruskeresők a tömörített állományokba is bele tudnak nézni – ami persze növeli a művelet időigényét, de alaposabb vizsgálatot eredményez. Ha erre nem képes, akkor minden azóta keletkezett tömörített állományt, amikor a fertőzést gyanítjuk, bontsunk ki és ellenőrizzük le.

Felhasznált irodalom

1: Gross Balázs, A szoftver-licenszerződések típusai, 2001., <http://www.jogiforum.hu/publikaciok/>
19