

INFORMATIKA ALAPJAI MÉRNÖKÖKNEK, ALAPSZAKOS HALLGATÓK SZÁMÁRA

**Dr. Hampel György
Heves Csilla**

2019.



EFOP-3.4.3-16-2016-00014



Jelen tananyag a Szegedi Tudományegyetemen készült az Európai Unió támogatásával.

Projekt azonosító: EFOP-3.4.3-16-2016-00014.

Szerzők:

Dr. Hampel György PhD Szegedi Tudományegyetem Mérnöki Kar

Heves Csilla Szegedi Tudományegyetem Mérnöki Kar

Lektorálta:

Dr. Fabulya Zoltán PhD Szegedi Tudományegyetem Mérnöki Kar

ISBN: 978-963-306-665-2

© Dr. Hampel György PhD 2019

© Heves Csilla 2019

Szegedi Tudományegyetem

Cím: 6720 Szeged, Dugonics tér 13.

www.u-szeged.hu

www.szechenyi2020.hu



MAGYARORSZÁG
KORMÁNYA

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE

TARTALOMJEGYZÉK

ELŐSZÓ	5
1. BEVEZETÉS AZ INFORMATIKÁBA.....	7
1.1. Adat	8
1.2. Információ	9
1.3. Számítástudomány, számítástechnika, informatika	13
1.4. Számítógép-architektúra	14
1.5. A számítógép fogalma és a számítógépek csoportosítása	14
1.6. Adatmennyiség, mértékegységek	15
1.7. Ellenőrző kérdések.....	18
2. SZÁMRENDSZEREK. SZÁMOK ÁBRÁZOLÁSA. KARAKTERKÓDOLÁS. LOGIKAI MŰVELETEK ÉS KAPUK.....	19
2.1. Számrendszerek.....	20
2.2. Számok ábrázolása	22
2.3. Szöveg kódolása.....	23
2.4. Logikai értékek, logikai műveletek	24
2.5. Kapuk	26
2.6. Ellenőrző kérdések.....	27
3. HARDVER ISMERETEK	28
3.1. A hardver fogalma	29
3.2. A Neumann-elvű számítógép felépítése	29
3.3. A központi egység.....	30
3.4. Perifériák.....	33
3.5. Input/output perifériák: háttértárak	34
3.6. A memóriák hierarchiája	36
3.7. Input perifériák	38
3.8. Output perifériák	39
3.9. Ellenőrző kérdések.....	41
4. SZOFTVER ISMERETEK.....	42
4.1. A szoftver fogalma	43
4.2. A szoftverek csoportosítása szolgáltatásaik alapján	44
4.3. A szoftverek csoportosítása hozzáférhetőség szerint	45
4.4. A szoftver és a szerzői jog	46
4.5. Ellenőrző kérdések.....	48
5. SZÁMÍTÓGÉP-HÁLÓZATI ALAPISMERETEK	49
5.1. Számítógép-hálózati alapfogalmak.....	50
5.2. A hálózatok céljai	51
5.3. A hálózatok csoportosítása.....	51
5.4. A hálózat felhasználói.....	53
5.5. Hálózati topológia.....	54
5.6. A hálózati kommunikáció módjai	55
5.7. Összekapcsoló közegek	56
5.8. Hálózati hardver	57
5.9. Az internet és fő szolgáltatásai	59
5.10. Ellenőrző kérdések.....	61

6. OPERÁCIÓS RENDSZEREK ÉS ÁLLOMÁNYKEZELÉS.....	62
6.1. Az operációs rendszerek fogalma, fő feladatai	63
6.2. Az operációs rendszerek fajtái, alapvető részei, csoportjai	66
6.3. Az operációs rendszerek felhasználói felülete	68
6.4. Állományszervezés	71
6.5. Állománykezelés	72
6.6. Ellenőrző kérdések.....	74
7. MULTIMÉDIA ISMERETEK.....	75
7.1. A multimédia fogalma	76
7.2. A multimédia személyi számítógép fogalma, jellemzői	77
7.3. A multimédia személyi számítógép bemenete, kimenete és szoftverei	78
7.4. Veszteséges és veszteség nélküli tömörítés	79
7.5. Vektorgrafika, pixelgrafika.....	79
7.6. Mozgóképek	81
7.7. Hangkezelés	83
7.8. Ellenőrző kérdések.....	86
8. IRODAI ALKALMAZÁSOK	87
8.1. Az irodai alkalmazások fogalma, szolgáltatásai	88
8.2. Szövegszerkesztés.....	88
8.3. Bemutató-készítés	90
8.4. Táblázatkezelés.....	92
8.5. Adatbázis-kezelés	94
8.6. Ellenőrző kérdések.....	97
9. ADATBÁZIS-TERVEZÉS	98
9.1. Az adatbázis fogalma	99
9.2. Az adatmodell fogalma és elemei	99
9.3. Az adatbázis-tervezés lépései.....	100
9.4. Az egyed-kapcsolat modell elemkészlete, ábrázolás a modellben	103
9.5. A relációs adatmodell elemkészlete, ábrázolás a modellben	106
9.6. Egyed-kapcsolat modell átalakítása relációs modellé.....	109
9.7. Ellenőrző kérdések.....	111
10. A PROGRAMOZÁS ALAPJAI.....	112
10.1. Az algoritmus és a programozás fogalma, értelmezése	113
10.2. A programozás fogalma és lépései	113
10.3. A programozási nyelv fogalma és csoportjai	114
10.4. Forrás, értelmező, fordító	115
10.5. Programok felépítése, szerkezete	116
10.6. Az algoritmus és leíró eszközei: folyamatábra és pszeudokód	117
10.7. Ellenőrző kérdések.....	122
IRODALOMJEGYZÉK	123

ELŐSZÓ

A mérnökképzésben az informatika kurzus számos további kurzus megalapozásául szolgál. Az informatikai ismeretek azonban nem csak egy-egy konkrét kurzus teljesítéséhez szükségesek, hanem – mivel információs társadalomban élünk – a mindennapi boldoguláshoz is nélkülözhetetlenek.

A munka világában a digitális írástudás a mai kor mérnökétől elvárt nélkülözhetetlen „kapu képesség”, amely magában foglalja

- az IKT írástudást, amely lehetővé teszi az elektronikus szolgáltatások igénybevételét az információs társadalomban,
- a technikai írástudást (vagy számítógépes írástudást), amely a számítógép és a hozzá tartozó programok kezelésén felül azok mélyebb megértését is feltételezi,
- az információs írástudást, amely az információ megtalálását, feldolgozását és megosztását tartalmazza.

Az első évfolyamos hallgatók számára tartott informatika kurzusokon rendre tapasztaljuk, hogy bár a résztvevők középiskolai tanulmányaik során tanultak számítástechnikát, informatikát, még sincsenek tisztában a szakterület alapvető fogalmaival, az eszközök (hardver és szoftver) által kínált szolgáltatásokkal, így azok hatékony módon történő felhasználása is elmarad a várakozásoktól.

A tankönyv célja:

- A tankönyv tíz fejezetében szeretnénk tömören összefoglalni azokat az informatikai alapismereteket, amelyekre a mérnök hallgatók – a tanulmányaik során – ráépíthetik a további informatikai ismereteiket.
- A cél elsősorban az adott fejezet témájához tartozó – középiskolában már jórészt tanult – alapvető fogalmak felelevenítése és megértetése, a meglévő tudás kiegészítése és rendszerezése – ahol indokolt, példával, feladattal összekötve.

Nem cél a programok részletekbe menő bemutatása, illetve a kezelésük elmagyarázása!

A tankönyv által felölelt témakörök:

- 1. fejezet: Az informatika néhány fontos alapfogalma (adat, információ, informatika, számítógép stb.); a szakterületen alkalmazott számrendszerek és a számkonvertálás módszerei.
- 2. fejezet: A számok és a szöveg kódolásának módszerei; logikai műveletek elvégzése, logikai műveletek megvalósítása kapuáramkörökkel.
- 3. fejezet: A hardver fogalma és a számítógép fő alkotóelemeinek jellemzői, szolgáltatásai.
- 4. fejezet: A szoftver fogalma, csoportosítása és jellemzői; a szoftverhez kapcsolódó jogi szabályozás.
- 5. fejezet: Számítógép-hálózati alapfogalmak, hálózati hardver, felhasználók és jogosultságaik; az internet és legelterjedtebb szolgáltatásai.

- 6. fejezet: Az operációs rendszer fogalma, feladatai és jellemzői; állományszervezés és -kezelés.
- 7. fejezet: A multimédia értelmezése, részei, jellemzői; grafika-, videó- és hangkezelés.
- 8. fejezet: Az irodai alkalmazások (szövegszerkesztő, bemutató-készítő, táblázatkezelő és adatbázis-kezelő) fogalma, a fő szolgáltatások, a kezelt állományok jellemzői.
- 9. fejezet: Az adatbázis és modellezés fogalma; az egyed-kapcsolat modell és a relációs adatmodell használata.
- 10. fejezet: Az algoritmizálás és a programozás fogalmai, a programozási nyelvek; algoritmusok leírása folyamatábra és a pszeudokód segítségével.

A tananyag elsajátításához szükséges előzetes ismeretek:

- a személyi számítógép kezelésének felhasználói szintű, alapvető ismerete;
- egy operációs rendszer és a fájlkezelő műveletek felhasználói szintű, alapvető ismerete;
- irodai programok (szövegszerkesztő, bemutató-készítő, táblázatkezelő és adatbázis-kezelő) kezelésének felhasználói szintű, alapvető ismerete;
- egy böngészőprogram kezelésének felhasználói szintű, alapvető ismerete;

A tananyag elsajátításához szükséges technikai feltételek:

- a tananyag digitális változatának megjelenítéséhez PDF állomány megnyitására alkalmas eszköz;
- a tananyagban szereplő programok használatához és a feladatok megoldásához:
 - személyi számítógép operációs rendszerrel,
 - tetszőleges irodai programcsomag,
 - egy böngészőprogram internethozzáféréssel.

Bízunk abban, hogy ezzel a tankönyvvel az olvasó nagy segítséget kap informatikai tanulmányai megalapozásához és további tanulmányai sikeres teljesítéséhez.

A szerzők

*Dr. Hampel György PhD
(1-7. fejezet, 9-10. fejezet)*

*Heves Csilla
(8. fejezet)*

Jelen tananyag a Szegedi Tudományegyetemen készült az Európai Unió támogatásával. Projekt azonosító: EFOP-3.4.3-16-2016-00014.

1. BEVEZETÉS AZ INFORMATIKÁBA

Elvárt tanulási eredmények

Tudás

- Ismeri az adat és az információ fogalmát és érti a két fogalom közötti különbséget
- Ismeri a mérési skálákat
- Érti az információ osztályozási szempontjait
- Ismeri a számítástudomány, a számítástechnika, az informatika és a számítógép-architektúra fogalmát
- Felsorolja a számítógép és a számítógép-kategóriák fő jellemzőit
- Ismeri az adattárolás alapegységét, a bit és byte kiszámításának módját
- Érti a bináris és decimális prefixumok közötti különbséget

Képesség

- Egy konkrét adatot a megfelelő mérési skálába sorol
- Egy adott információt a megfelelő csoportba sorol be
- A bináris és decimális prefixumokat helyesen használja

Attitűd

- Elfogadja az információ hasznosságát, jelentőségét szakmai döntésekben
- Értékként tekint az adatra és az információra
- Nyitott az informatikai eszközök alkalmazására saját szakterületén
- Rugalmasan és kritikusan kezeli az informatika területén alkalmazott, adattároláshoz kapcsolódó mértékegységeket

Autonómia, felelősség

- Szakmai iránymutatás alapján dönt adat és információ használhatóságáról egy adott helyzetben
- Szakmai iránymutatás alapján fogalmaz meg átfogó javaslatot egy adott feladat elvégzésére alkalmas számítógépre

A hétköznapi beszédben az adat és az információ egymás szinonimái, általában nem teszünk különbséget közöttük. A különböző tudományterületek kutatói is többnyire hasonlóan járnak el, míg vannak, akik megkülönböztetik a két fogalmat.

1.1. Adat

Az adat

- egy elem (objektum) egy meghatározott tulajdonságának (jellemzőjének, attribútumának) értéke, állapota, rögzített jellemzője, amelyet megfigyeléssel, méréssel állapíthatunk meg,
- tények, fogalmak, eligazítások formai megjelenése, amely alkalmas emberi vagy automatikus gépi kommunikációra, értelmezésre és
- olyan tények, fogalmak, utasítások, amelyek alkalmasak emberi vagy gépi feldolgozásra, értelmezésre, kommunikációra,
- rögzített információ,
- információt ábrázoló, azt megtestesítő elemi jel.

Láthatjuk, hogy az adatot többféle módon lehet jellemezni. Mondhatjuk, hogy az adat egyfajta nyersanyag, amelynek feldolgozásával információ keletkezik, és amelyből az emberi tapasztalatok és képességek révén végül tudás lesz.

Attól függően, hogy az adaton milyen további feldolgozás (művelet) végezhető, négyféle mérési szintet, **mérési skálát** különböztetünk meg. Ezek egymásra épülnek, azaz a magasabb szintű skálán minden olyan művelet értelmezhető, ami a nála alacsonyabb szintű skálán is:

1. Nominális skála (névleges skála): A vizsgált objektumokról csak annyit tudunk megállapítani, hogy a tulajdonságok azonosak-e vagy sem. Ez a skálátípus csak osztályokra, kategóriákra való bontásra alkalmas. Példa: férfi – nő, beteg – egészséges.
2. Ordinális skála (sorrendi skála): Az értékek alapján az objektumok sorrendisége is megállapítható, azaz tulajdonságértékek sorba állíthatók. Példa: iskolai végzettség, fogyasztói elégedettség szintje, bőrtípusok világos bőrtől [I-es típus] a nagyon sötét bőrig [IV-es típus]).
3. Intervallum skála: A skálaértékek különbsége is értelmezhető és a skálának része valamilyen mértékegység is. A skála kezdőpontja (nullpontja) azonban vagy önkényes, vagy konvención alapszik. Példa: hőmérsékletmérés Celsius vagy Fahrenheit skálán, intelligenciaszint mérés.
4. Arány skála: Ez a legmagasabb rendű skála, ahol már a kezdőpont is egyértelműen rögzített, így bármilyen skálaérték egymáshoz viszonyított aránya is értelmezhető. Ehhez a skálához is tartozik mértékegység. Példa: testmagasság, testsúly és termelés természetes mértékegységben kifejezve, jövedelem vagy árbevétel nagysága pénzürtékben kifejezve.

1.2. Információ

Az információ fogalmát talán az ún. visszatükröződés-elmélet hívei értelmezik legtágabban: szerintük az információ nem más, mint a valóság visszatükröződése. Az információ tág értelmezését vallják a matematikai információelmélettel foglalkozók is. Az információelmélet olyan matematikai fogalmakkal dolgozik, amelyek a matematika segítségével próbálják meg leírni az információt, pontosabban az információ mennyiségét.

A szerzők az adat és az információ elkülönítésére többnyire

- az értelmezést,
- a feldolgozást,
- az újdonságot, új ismeretet (még akár egy gép számára is),
- a bizonytalanság csökkentését,
- a kommunikációt és
- a szubjektivitást, az emberi gondolkodást

használják fel.

Az előbbiek alapján az információt a következőképpen határozhatjuk meg: Az **információ**

- adatsoroknak, tényeknek valamely folyamat által használható formára alakított változata,
- tudásanyag, intelligencia, hír, illetve ezek átadása és fogadása,
- adatformában megjelenő értelmezése bizonyos dolgoknak,
- tevékenység, amely valaki tudásának a megváltoztatásához vezet,
- tényközlés formájában történő tudásanyag-átadás.
- hír, amely a címzett részére közvetlenül vagy további feldolgozás útján jelentőséggel bír,
- olyan tájékoztatás, közlés, adat, ismeret, hír, amely a címzett által értelmezhető és célja a bizonytalanság csökkentése.

Az információ-meghatározások alapvetően két nagy csoportba sorolhatók: megkülönböztetünk hagyományos (objektív) és szubjektív információ-értelmezést:

- A **„hagyományos” objektív információ-felfogás** szerint az információ abszolút és semleges, azaz nem függ az információt fogadó féltől. Az információ a továbbítása során változatlan, mindössze a megjelenési formája változik. A felhasználók az információt anélkül továbbíthatják vagy alakíthatják más formába, hogy annak tartalmát értenék. Áramlása során a változatlan információ különféle formákban jelenhet meg, amely lehet elektronikus (strukturált szám- és kódhalmaz), hangalapú (élőszó, hangfelvétel), képi (kézzel rajzolt, nyomtatott, digitalizált) vagy éppen szövegformátumú (írott, nyomtatott, digitalizált).
- A **szubjektív információ-felfogás** szerint az információ nem abszolút és nem állandó. Léteznek bizonyos adatnak nevezett elemi ismeretek, amelyek mérés vagy tapasztalat eredményeként jönnek létre. Az adatok csupán nyers tények, nem

hordoznak jelentést, csak a megfigyelő személyben alakulnak át információvá. Az információ nem azonos az adatok halmazával, hanem azokból egy aktív folyamat során keletkezik, amelyben a felhasználó fejében lévő háttérismeret (a tudás) fontos szerepet játszik. Ez a háttérismeret is egy információhalmaz, amely megadja az információ értelmezési környezetét. Ennek alapján az információ az adatok jelentése, az adatok értelmezése valamilyen (egyéni, szervezeti) tudással összefüggésben. Az információnak nincs saját fizikai megjelenése, így továbbítása csak közvetett módon, különböző adatformátumok (beszéd, írás, hang, álló- vagy mozgóképek) használatával lehetséges. A kommunikációs folyamatok során így csak adatátadás történik, és az adat, nem pedig az információ az, amely különböző formákba alakulhat át. A félreinformálás jelenségét az okozza, hogy nincs lehetőség az információ közvetlen átadására. A kommunikáló felek információikat valamilyen kommunikációs adatformára alakítják, és csak adatokat közölnek egymással, miközben arra is figyelniük kell(ene), hogy a befogadó rendelkezzen az adatok helyes értelmezéséhez szükséges háttértudással.

Míg tehát az objektív információ értelmezések azt feltételezik az információ továbbításában részt vevőkről, hogy azok nem képesek, esetleg nincs is szükségük az információ tartalmának helyes értelmezésére, addig a szubjektív értelmezések esetében az információ kialakításában a felhasználóé a főszerep.

Az adat és az információ kapcsolatát mutatja az 1. ábra.

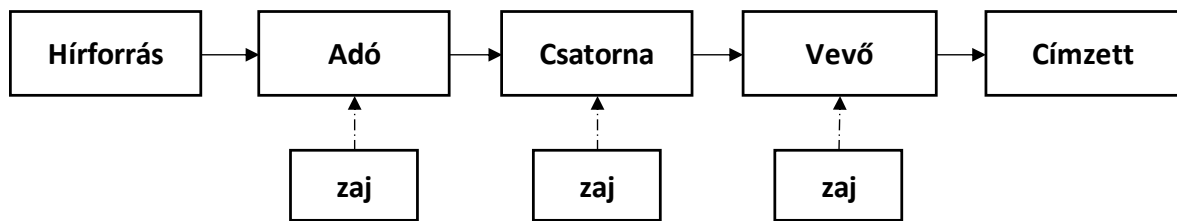


1. ábra. Adat és információ kapcsolata.

Forrás: Kacsukné–Kiss (2009)

A gyakorlatban az adat és információ fogalmának elkülönítése nem egyértelmű, hiszen nem mindenről lehet azonnal megállapítani, hogy adatról vagy információról van-e szó. Ennek oka, hogy ez függ az adott helyzettől és – ahogyan már említettük – az adatot, információt fogadó fél háttértudásától is.

Az **információ közlés** (kommunikáció) során adható át (2. ábra). A kommunikációt a felek információközlés céljából kezdik el. A közlendő információt (hírt) valamilyen módon kódolni kell, azaz át kell alakítani a közlésre alkalmas adatformára annak érdekében, hogy továbbítható legyen az átviteli csatornán. A kódolt hírt közleménynek nevezzük. A közlemény valamilyen csatornán keresztül továbbítódik. A hír a továbbítás során számos zajforrásból származó zavaró hatás miatt torzulhat. A torzított hír eljut a címzethez, aki/ami azt értelmezi.



2. ábra. A kommunikáció modellje.

Forrás: Bodnár–Magyary (2005) alapján a szerző szerkesztése.

Az és adatot és az információt sokféleképpen sorolhatjuk különböző csoportokba. A legjellemzőbb **információosztályozási szempontok** a következők:

1. A **feldolgozhatóság jellege** szerint megkülönböztethetünk kvantitatív, kvalitatív, valamint „fuzzy” információkat.
 - A **kvantitatív** (azaz mennyiségi) **információ** tartalma egyértelműen mérhető, számszerűen meghatározható. Az ilyen ismeretanyag beépíthető a feldolgozási algoritmusokba, behelyettesíthető a matematikai modellekbe és elemzésekre is alkalmas, hiszen a számok egymással könnyen összehasonlíthatók. Ez a fajta információ a számítógéppel támogatott információfeldolgozás legjellemzőbb kategóriája.

Példa:

XY részvénytársaság éves árbevétele 1 450 millió Ft.

A processzor hőmérséklete 68°C.

- A **kvalitatív** (azaz minőségi) **információk** minőségi jellemzést adnak valamilyen dologra. Ez a fajta információ is gyakori, hiszen egy objektumhoz tartozó tulajdonság nem minden esetben fejezhető ki egyszerűen, konkrét számadattal.

Példa:

Fogyasztói elégedettség mérésekor általában ordinális skálát használunk, ahol az egyes skálaértékek minőségi jellemzők (teljesen elégedetlen vagyok, kismértékben elégedett vagyok, nagymértékben elégedett vagyok, teljes mértékben elégedett vagyok).

Mérőeszköz hiányában nem állíthatjuk, hogy a processzor hőmérséklete 68°C, legfeljebb annyit mondhatunk, hogy a processzor meleg (amire a gépben lévő ventilátorok zúgásából, vagy a gépből kiáramló levegő hőjéből következtethetünk).

A kvalitatív információk feldolgozását megkönnyítheti, ha a minőségi adatokhoz számokat rendelünk hozzá. Ugyanakkor a feldolgozás során végig szem előtt kell tartanunk, hogy milyen mérési skálával dolgozunk és azon milyen műveletek elvégzésének van értelme.

Fuzzy (azaz bizonytalan, ködös) **információról** beszélünk akkor, ha egy adott dologról van valamilyen ismeretünk, de ez nagyon viszonylagos és több tényezőtől függő ismeretünket nem tudjuk sem pontosan megfogalmazni, sem minőségileg értelmezni. A fuzzy információk feldolgozása és kezelése a számítógépes rendszerek (és szakemberek) számára a legnagyobb kihívás.

Példa:

X. Y. nagyon gazdag. – De valójában nem ismert, hogy pontosan mekkora a vagyona. Ezen kívül országonként jelentős különbségek lehetnek az életszínvonalban, vagyoni és jövedelmi viszonyokban, ezért az sem mindegy, hogy hol él az illető.

Ez a számítógép nagyon gyors. – Az éppen elvégzett feladatot az adott felhasználó szerint gyorsan végzi el a gép. Lehet viszont, hogy más feladatot lassan végez el és az is előfordulhat, hogy másvalaki, aki nagyobb teljesítményű géphez van hozzászokva, lassúnak fogja találni ugyanazt a gépet.

2. A **keletkezés módja** szerint beszélünk primer és szekunder információkról.

- **Primer információknak** nevezzük azokat, amelyek a valóság objektumaihoz, tevékenységeihez eleve hozzátartozó jellemzők, amelyek az adott elemekkel együtt léteznek és azokat feldolgozás nélkül, közvetlenül jellemzik.

Példa:

Valakinek a neve, születési ideje és címe.

A vállalat árbevétele az elmúlt 10 évben.

A háttértárra írható maximális adatmennyiség.

- **Szekunder információknak** nevezzük azokat az információkat, amelyek a tiszta, ténybeli adatok – vagyis a primer információk – összegyűjtése, összesítése, csoportosítása, értékelése, értelmezése által keletkeznek.

Példa:

A születési idő és az aktuális dátum alapján az életkor meghatározása.

Trendszámítással a jövőben várható árbevétel alakulása.

Az átlagos napi írási mennyiség alapján a háttértár várható élettartama.

3. Az **információforrás típusa** szerint léteznek belső és külső információk.

Információforrásnak tekinthetünk minden olyan elemet, objektumot, amely képes arra, hogy információkat szolgáltatson (world wide web, könyv, újság, televízió, rádió, egy ember stb.).

- A **belső (intern) információ** a vizsgált rendszerből, annak belsejéből származik.

Példa:

Egy vállalat könyvelése.

Egy számítógép teljesítményfigyelő szoftvere által szolgáltatott információk a rendszer működéséről.

- A **külső (extern) információ** a vizsgált rendszer külső környezetéből származik.

Példa:

Egy vállalat számára a piacról, vagy a versenytársakról szerzett információk.

Egy adott számítógépes rendszerről külső szakemberek, tesztelők által végzett, publikált vizsgálatok eredményei.

4. Az **információtartalom teljessége** szerint: teljes, részleges információk

Ha az információt a vizsgált rendszer szempontjából elemezzük, és azt nézzük, hogy teljesen, vagy csak bizonyos mértékig jellemzi-e az adott rendszert, **teljes** vagy **részleges információról** beszélünk. Azt az ismeretet, amely bizonyos tudásanyagnak nem teljes egészét jelenti, hanem csak egy részét, vagy részeit tükrözi, részleges információnak nevezzük.

Példa:

Egy cég értékeléséhez az éves jelentés áll rendelkezésünkre, vagy csak az alkalmazottai számából, vagy a jegyzett tőkéjéből próbálunk meg következtéseket levonni.

Részletekbe menően ismerjük egy számítógép felépítését, a telepített programokat, azok teljesítményét, vagy csak a „bootolási” idő hossza alapján próbálunk következtetni a gép teljesítményére.

5. A **felhasználói igények szerint** megkülönböztethetünk releváns és irreleváns információkat.

Egy működő rendszer hatékonyságát alapvetően jellemzi, hogy a különböző feladatokat ellátók hozzájutnak-e a munkájukhoz szükséges fontos, lényeges, azaz **releváns** információkhoz. Azok az információk, amelyek egy adott helyzetben lényegtelenek, vagyis **irrelevánsak** – a szubjektív információértelmezés szerint – nem is tekinthetők információnak, csupán adatnak.

Példa:

A vállalat számára a gyártandó termékek mennyiségének meghatározásához releváns információ a megrendelések jelenlegi és várható alakulása, de irreleváns információ az alkalmazottak születésnapja.

A számítógép tárolókapacitásának bővítésekor releváns információ a jelenlegi és a jövőben várhatóan tárolandó állományok mérete, de irreleváns információ a nyomtató nyomtatási sebessége.

1.3. Számítástudomány, számítástechnika, informatika

A **számítástudomány** (angolul: computer science) a számítógépek létrehozásával, működésével és információ-feldolgozó tevékenységével foglalkozó tudomány. Számos tudományterület vívmányait – így például a matematika, információelmélet, a logika, a tervezés, az elektronika, a viselkedéstudomány stb. – vívmányait ötvözi.

A **számítástechnika** a Wikipedia megfogalmazása szerint: „az automatizált adatfeldolgozás eszközeivel és azok különböző területeken való használatával foglalkozó elméleti és alkalmazott műszaki tudomány. Hozzá tartozik a hardver- és szoftverelemek tanulmányozása, a működést segítő szervezési, illetve alkalmazási, szolgáltatási összetevők rendszere” (Wikipedia 1). Tulajdonképpen az informatika egyik részterülete és alapvető alkalmazása a számítógépek építése.

Az **informatika** az információ áramlásának módjaival, az információ feldolgozásának és hasznosításának módszereivel foglalkozó tudományág, amely az információ keletkezését, továbbítását, feldolgozását és hasznosítását vizsgálja. Az informatika kifejezés a 20. század 60-as éveiben keletkezett és akkor még elsősorban a modern eszközökkel végzett könyvtári

tájékoztatást, információellátást értették alatta. Ma már inkább egy olyan több tudományt átfogó (multidiszciplináris) területet értenek rajta, amely az információval foglalkozik: az információ létrejöttétől, gyűjtésén, rendszerezésén, reprodukálásán és tárolásán át a közvetítéséig. Magában foglalja a hardvert, a szoftvert, az adattárolást, a számítógép- és telekommunikációs hálózatokat, valamint a kommunikációt is. Informatikai (információtechnológiai, IT) rendszerek fejlesztési, üzemeltetési és elemzési kérdéseivel foglalkozik és ezeket a tevékenységeket elsősorban számítógépen végzi:

- elméleti úton azáltal, hogy módszereket, modelleket, szabályokat, eljárásokat dolgoz ki számítógépek készítéséhez és működtetéséhez,
- mérnöki tevékenységgel úgy, hogy számítógépeket készít és azokhoz elektronikai eszközöket alkot (hardver),
- rendszertervezéssel és rendszerkészítéssel, azzal, hogy a számítógépek működtető eszközeit létrehozza (szoftver),
- alkalmazza a számítógépet, különböző feladatok elvégzésére (szövegszerkesztésre, raktárkészlet-nyilvántartásra, programozásra stb.)

1.4. Számítógép-architektúra

Bár az informatika egyik alapfogalmáról van szó, a **számítógép-architektúrának** még sincs általánosan elfogadott értelmezése. A kifejezést az IBM-nél vezették be: egy 1964-ben megjelent cikkben szereplő meghatározás szerint az architektúra a számítógép azon felépítését jelenti, amelyet egy alacsony szintű nyelven programokat fejlesztő szakembernek ismernie kell ahhoz, hogy megfelelő programokat tudjon írni az adott számítógépre. Az értelmezés magában foglalta a memória leírását, a gép utasításkészletét, utasításszerkezetét, címzési módjait, az utasítások aktuális kódolásának megadását, ugyanakkor nem tartalmazta a konkrét hardveres struktúrát és az áramkörökben és összekapcsolásokban használt technológiát.

A fogalom leírása az idővel finomodott, részben módosult. Ma architektúrának a rendszerek alapvető felépítését, működési elvét nagy vonalakban bemutató tervet, koncepciót nevezzük.

Az architektúra kifejezést használhatjuk az azonos alapelvek szerint működő hardverek vagy szoftverek esetében is (például „x64 architektúra”).

Az architektúra a hardver és szoftver alkotóelemek és azok kapcsolódása leírásán túl megadja, hogyan lehet az egyes rendszerfunkciókat konkrét szoftver- és hardverelemekre leképezni, illetve azt is, hogy a szoftverkomponensek hogyan rendelhetők hozzá a rendszert alkotó hardverelemekhez.

1.5. A számítógép fogalma és a számítógépek csoportosítása

A számítástechnika rohamos fejlődésen ment keresztül az elmúlt évtizedekben, azonban a számítógépek alapvető működés módja lényegében a kezdetek óta változatlan. A **számítógép** (computer) **legfontosabb jellemzői** és egyben a **meghatározása** is:

- meghatározott feladatok nagy sebességű elvégzésére szolgáló,
- széles körben alkalmazható (univerzális),

- automatikus működésű, azaz emberi beavatkozás nélkül működő,
- programozható, tehát képes egy vagy több utasítássorozat vagy program utasításait önállóan végrehajtani,
- kívülről vezérelhető, azaz a működése kívülről megváltoztatható, illetve megszakítható,
- működtető vagy ún. operációs rendszerrel rendelkező,
- gép, azaz műszakilag megvalósított rendszer, amelynek
- meghatározott feladatokat ellátó hardver egységei vannak.

A számítógép olyan gép, amely a következő **három feladatot végre tudja** hajtani:

- strukturált bemeneti adatokat fogad,
- az adatokat előírt szabályok szerint feldolgozza,
- az eredményeket visszaadja.

A számítógépeket **hardver felépítésük** alapján az alábbi fő **kategóriákba** sorolhatjuk:

- Beágyazott számítógép: okos eszközökben láthatatlan számítógépek. Manapság találhatunk ilyet hűtőgépekben, klímaberendezésekben, robotporszívókban, televíziókészülékekben, hangszórókban, sőt akár villanykörtékben is.
- Játék gép: egyszerűbb programok futtatására alkalmas, de elsősorban játékprogramokhoz használják. Ilyen gép az X-Box, Playstation, Nintendo Wii stb.
- Személyi számítógép (personal computer, PC): egy személy használatára szolgáló, hordozható vagy asztali számítógép. Ebben a kategóriában kis, közepes és nagy teljesítményű gépek egyaránt megtalálhatók. Két leggyakoribb típusa az IBM PC kompatibilis személyi számítógép és az Apple Macintosh gépcsald.
- Munkaállomás (workstation): asztali számítógép, amelyet gyakran több személy, illetve több munkahely kiszolgálására használnak.
- Nagygép (mainframe): egy vagy több egységből álló, akár egy termet betöltő berendezés, amelyet sok felhasználó kiszolgálására, továbbá nagy mennyiségű, illetve bonyolult számítások végzésére használnak.

1.6. Adatmennyiség, mértékegységek

A számítógépek az **adatokat digitális formában** (azaz számjegyként) tárolják. Az információ mértékegysége a **bit**. A bit két diszkrét érték (egyértelmű) megkülönböztetését lehetővé tevő jel-pár. Az megállapodás kérdése, hogy értékeit a kettes számrendszer jegyeivel (0, 1) jelöljük – lehetne akár a piros-kék színpár is. A számítógépben 1 bit aktuális értékéről az információt a feszültség megléte, vagy éppen hiánya hordozza.

1 bit összesen két állapot megkülönböztetését teszi lehetővé. Ha ennél több állapotunk van, akkor több bitre lesz szükségünk: n bittel pontosan 2^n különböző értéket tudunk jelölni. Célrészéégi okokból a számítógépek felépítése és működése a 8 bitet tartalmazó csoportok köré szerveződött. 8 bit együttesét **byte**-nak nevezzük (magyarul is írhatjuk: **bájt**). 1 byte-on 2^8 (azaz 256) darab különböző érték ábrázolható.

A **bináris prefixumokat** (előtagokat) a számítástechnikában használják az alapmértékegység kerek többszöröseinek rövidítésére. Általában véve a prefixum értelme, hogy a nagyobb számokat tartalmazó mértékeket a mértékegység előtt használt jelzéssel – a prefixummal – rövidebben fejezhetjük ki.

A számítástechnikában a prefixumokat többek között a memória méretének alapegységei, a bit és a byte elé illesztve használják. Ám, hogy például a „kilo” ilyen esetben pontosan mennyit is jelent, arról több évtizedes vita folyt, ill. folyik.

A Nemzetközi Elektrotechnikai Bizottság (IEC: International Electrotechnical Commission) 1998-ban véglegesítette az új jelölésrendszert. Eszerint az SI rendszerben rögzített prefixumokat ezután kizárólag a decimális alapú értelmezésben lehet használni (kilo = 10^3) – még az informatika területén is. Ugyanakkor, mivel az informatikában szükség van a bináris előtagokra is, azokra új elnevezéseket javasoltak (lásd: 1. táblázat), amit az **IEC 60027-2** számú szabványában rögzítettek. Az IEC-szabványt a Magyar Szabványügyi Testület 2007-ben honosította, és **MSZ EN 60027-2** néven kihirdette.

SI (decimális)				IEC (bináris)			
jel	név	érték		jel	név	érték	
k	kilo	10^3	1000^1	Ki	kibi	2^{10}	1024^1
M	mega	10^6	1000^2	Mi	mebi	2^{20}	1024^2
G	giga	10^9	1000^3	Gi	gibi	2^{30}	1024^3
T	tera	10^{12}	1000^4	Ti	tebi	2^{40}	1024^4
P	peta	10^{15}	1000^5	Pi	pebi	2^{50}	1024^5
E	exa	10^{18}	1000^6	Ei	exbi	2^{60}	1024^6
Z	zetta	10^{21}	1000^7	Zi	zebi	2^{70}	1024^7
Y	yotta	10^{24}	1000^8	Yi	yobi	2^{80}	1024^8

1. táblázat. SI és IEC prefixumok.

Forrás: IEC 60027-2 szabvány alapján a szerző szerkesztése.

A kilobyte és a kibibyte jelentése:

1 kilobyte (kB – kis „k”!) = 10^3 byte = 1000 byte

1 kibibyte (KiB – nagy „K”!) = 2^{10} byte = 1024 byte

A bit és többszörösei (amit elsősorban számítógép-hálózati sávszélességek, sebességek megadásánál használjuk):

1 kilobit = 10^3 bit = 1 000 bit

1 megabit = 10^6 bit = 1 000 000 bit

1 gigabit = 10^9 bit = 1 000 000 000 bit

Példa:

8 GB-os háttértár tárolókapacitása különböző prefixumokkal megadva:

Érték	Mértékegység és jele		Kiszámítása	
8 000 000 000,0	byte	B		
64 000 000 000,0	bit	b	byte · 8	
8 000 000,0	kilobyte	kB	byte / 10 ³	byte / 1000
7 812 500,0	kibibyte	KiB	byte / 2 ¹⁰	byte / 1024
8 000,0	megabyte	MB	byte / 10 ⁶	byte / 1 000 000
7 629,4	mebibyte	MiB	byte / 2 ²⁰	byte / 1 048 576
⁽¹⁾ 8,0	gigabyte	GB	byte / 10 ⁹	byte / 1 000 000 000
⁽²⁾ 7,5	gibibyte	GiB	byte / 2 ³⁰	byte / 1 073 741 824

(1): Ezt adja meg a szabványt alkalmazó háttértár-gyártó a háttértár kapacitásának. (2): Ezt írja ki a – kompatibilitási okokból – régi módszer szerint számoló operációs rendszer.

1TB-os (1000 GB-os) háttértár tárolókapacitása különböző prefixumokkal megadva.

Érték	Mértékegység és jele		Kiszámítása	
1 000 000 000 000,0	byte	B		
8 000 000 000 000,0	bit	b	byte · 8	
1 000 000 000,0	kilobyte	kB	byte / 10 ³	byte / 1000
976 562 500,0	kibibyte	KiB	byte / 2 ¹⁰	byte / 1024
1 000 000,0	megabyte	MB	byte / 10 ⁶	byte / 1 000 000
953 674,3	mebibyte	MiB	byte / 2 ²⁰	byte / 1 048 576
⁽¹⁾ 1 000,0	gigabyte	GB	byte / 10 ⁹	byte / 1 000 000 000
⁽²⁾ 9 313,2	gibibyte	GiB	byte / 2 ³⁰	byte / 1 073 741 824

(1): Ezt adja meg a szabványt alkalmazó háttértár-gyártó a háttértár kapacitásának. (2): Ezt írja ki a – kompatibilitási okokból – régi módszer szerint számoló operációs rendszer.

Amint az a példából látható, minél nagyobb a háttértár tárolókapacitása, annál nagyobb a különbség a kétfajta számítási módszerrel megadott érték között. A zavart az okozza, hogy sok esetben az operációs rendszer például gigabyte-ot ír ki ott, ahol a szabvány szerint gibibyte-ot kellene írnia: a gyártó által 8 GB-osként forgalmazott pendrive-ról az operációs rendszer azt állítja, hogy 7,5 GB, ami a felhasználóban azt az érzést keltheti, hogy becsapták. Éppen ezért egyes háttértárkezelő programokban beállítható, hogy melyik rendszer szerint szolgáltatassa az adatokat. Ha ez nem állítható be, akkor szinte biztosak lehetünk abban, hogy az adott program a régi szabály szerint számol.

1.7. Ellenőrző kérdések

1. Mi az adat? [8]
2. Milyen mérési skálák léteznek és mik a fontosabb jellemzői? [8]
3. Mi az információ? [9]
4. Mi az információ az objektív felfogás szerint? [9]
5. Mi az információ a szubjektív felfogás szerint? [9]
6. Hogyan adódhat át az információ? [10]
7. Milyen fő szempontok szerint osztályozható az információ? [11]
8. Mi jellemzi a kvantitatív információt? [11]
9. Mi jellemzi a kvalitatív információt? [11]
10. Mi jellemzi a fuzzy információt? [11]
11. Mi jellemzi a primer információt? [12]
12. Mi jellemzi a szekunder információt? [12]
13. Mi a különbség a belső és a külső információ között? [12]
14. Mitől teljes, illetve részleges az információ? [13]
15. Felhasználói igények szerint hogyan csoportosítható az információ? [13]
16. Mi a számítástudomány? [13]
17. Mi a számítástechnika? [13]
18. Mi az informatika? [13]
19. Mit jelent a számítógép-architektúra? [14]
20. Mik a számítógép jellemzői és milyen feladatokat tud a számítógép végrehajtani? [14]
21. Hardver felépítésük alapján milyen fő számítógép-kategóriák léteznek és mik a kategóriák fő jellemzői? [15]
22. Milyen formában tárolja a számítógép az adatokat, mi a bit és a byte? [15]
23. Mit jelentenek az informatikában használt prefixumok? [16]

2. SZÁMRENDSZEREK. SZÁMOK ÁBRÁZOLÁSA. KARAKTERKÓDOLÁS. LOGIKAI MŰVELETEK ÉS KAPUK.

Elvárt tanulási eredmények

Tudás

- Leír bármilyen tetszőleges q -alapú számrendszert
- Ismeri a bináris, decimális és hexadecimális számrendszer jelkészletét
- Érti a fixpontos, lebegőpontos és BCD számábrázolást
- Ismeri a szöveg kódolásának módjait
- Felsorolja a logikai műveleteket és ábrázolási módjait
- Ismeri a logikai műveletek igazságtáblázatát
- Ismeri a kapuáramkörök fogalmát és a kapuk ábrázolási módját

Képesség

- Átalakít bináris, decimális és hexadecimális számokat a számrendszerek között
- Elvégez egyszerű és összetett logikai műveleteket
- Összeállít összetett logikai műveleteket végrehajtó áramköröket

Attitűd

- Elfogadja a számrendszerek és a kódolás hasznosságát a számítógéppel való kommunikáció során
- Figyelembe veszi, hogy a matematikai logika a számítógépek működésének fontos alapját képezi

Autonómia, felelősség

- Önállóan konvertál bináris, decimális és hexadecimális számrendszerek között
- Önállóan és kreatívan alkot logikai műveleteket végző áramköröket a szabályok betartása mellett
- Felelősséget vállal az önállóan végzett munkájáért

2.1. Számrendszerek

A matematikában egy **q-alapú számrendszer** a következő: Egy x szám felírása q -alapú számrendszerben: $a_n, \dots, a_1, a_0$, ha $0 \leq a_i \leq q$ fennáll minden $i=0, 1, \dots, n$ esetén és a következő egyenlőség teljesül:

$$x = a_n \cdot q^n + \dots + a_1 \cdot q + a_0,$$

ahol

q : a számrendszer alapszáma,

$a_n, \dots, a_0$: a q -alapú számrendszerbeli számjegyek,

x : a szám értéke,

q^i : az a_i számjegy helyértéke.

Abban az esetben, ha nem egyértelmű, hogy egy számot melyik számrendszerben adtunk meg, akkor azt a szám után zárójelben a következőképpen jelöljük: $10_{(2)}$ – ez egy kettes számrendszerbeli szám, $10_{(10)}$ – ez pedig egy tízes számrendszerbeli szám.

Amikor a felhasználó különböző feladatokhoz használja a számítógépet, akkor jellemzően **tízes (decimális) számrendszert** használ ($q=10$), hiszen ezt a számrendszert használja mindenki a hétköznapi életben, ehhez szokott hozzá. A számítógépet programozók ezen kívül találkozhatnak még a **kettes (bináris)** ($q=2$) és a **tizenhatos (hexadecimális)** ($q=16$) **számrendszerrel**. A bináris számrendszerrel azért, mert ahogy az előző alfejezetben szó volt róla, a számítógép mindent digitálisan, kétállapotú jelként tárol – végső soron az adatokat és az utasításokat is; a hexadecimális számrendszerrel pedig azért, mert bináris számokat egyszerűbben, tömörebben lehet leírni hexadecimális formában.

A decimális számrendszer **jelkészlete** 0 1 2 3 4 5 6 7 8 9 (10 darab szimbólum), a bináris számrendszeré 0 1 (2 darab szimbólum), a hexadecimális számrendszer számjegyei 0 1 2 3 4 5 6 7 8 9 A B C D E F. $2^4=16$ és ebből következően egy hexadecimális számjegy helyettesíthető négy bináris számjeggyel. Az átszámítás módját a 2. táblázat mutatja.

Decimális	Bináris	Hexadecimális
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

2. táblázat. Decimális, bináris és hexadecimális számok 0-tól 15-ig.

Az **átváltás a bináris és hexadecimális számrendszerről decimális számrendszerre** egyszerű: Jobbról első helyre az egyesek, majd balra haladva a számjegyek értékei n^1 , n^2 , $n^3...$ lesznek. A szorzatokat összeadva adódnak a számjegyek valódi értékei.

Példa:

$$1010011011_{(2)} = ?_{(10)}$$

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	0	1	1	0	1	1
$2^9 \cdot 1$	$2^8 \cdot 0$	$2^7 \cdot 1$	$2^6 \cdot 0$	$2^5 \cdot 0$	$2^4 \cdot 1$	$2^3 \cdot 1$	$2^2 \cdot 0$	$2^1 \cdot 1$	$2^0 \cdot 1$
512	0	128	0	0	16	8	0	2	1

$$1010011011_{(2)} = 2^9 \cdot 1 + 2^8 \cdot 0 + 2^7 \cdot 1 + 2^6 \cdot 0 + 2^5 \cdot 0 + 2^4 \cdot 1 + 2^3 \cdot 1 + 2^2 \cdot 0 + 2^1 \cdot 1 + 2^0 \cdot 1 = \\ = 512 + 0 + 128 + 0 + 0 + 16 + 8 + 0 + 2 + 1 = 667_{(10)}$$

$$29B_{(16)} = ?_{(10)}$$

16^9	16^8	16^7	16^6	16^5	16^4	16^3	16^2	16^1	16^0
0	0	0	0	0	0	0	2	9	B
$16^9 \cdot 0$	$16^8 \cdot 0$	$16^7 \cdot 0$	$16^6 \cdot 0$	$16^5 \cdot 0$	$16^4 \cdot 0$	$16^3 \cdot 0$	$16^2 \cdot 2$	$16^1 \cdot 9$	$16^0 \cdot 11$
0	0	0	0	0	0	0	512	144	11

$$29B_{(16)} = 16^9 \cdot 0 + 16^8 \cdot 0 + 16^7 \cdot 0 + 16^6 \cdot 0 + 16^5 \cdot 0 + 16^4 \cdot 0 + 16^3 \cdot 0 + 16^2 \cdot 2 + 16^1 \cdot 9 + \\ + 16^0 \cdot 11 = 512 + 144 + 11 = 667_{(10)}$$

Decimális számot bináris számra úgy alakíthatunk át, hogy a számot kettővel osztjuk mindaddig, amíg az el nem fogy, az osztási maradékot pedig sorra feljegyezzük. A maradékok sorra a bináris számrendszerbeli a_0 , a_1 , ..., a_n számjegyeket adják.

Példa:

$$667_{(10)} = ?_{(2)}$$

Művelet	Eredmény	Maradék
667:2	333	1
333:2	166	1
166:2	83	0
83:2	41	1
41:2	20	1
20:2	10	0
10:2	5	0
5:2	2	1
2:2	1	0
1:2	0	1

A fenti táblázat maradék oszlopában található számokat alulról felfelé összeolvasva megkapjuk a decimális szám bináris megfelelőjét:

$$667_{(10)} = 1010011011_{(2)}$$

Decimális számot hexadecimális számmá az előző módszerhez hasonlóan alakíthatunk át: decimális számot tizenhatalmal osztjuk mindaddig, amíg az el nem fogy, az osztási maradékokat pedig sorra feljegyezzük.

Példa:

$$667_{(10)} = ?_{(16)}$$

Művelet	Eredmény	Maradék
667:16	41	11
41:16	2	9
2:16	0	2

A fenti számsort ismét alulról felfelé összeolvasva megkapjuk a decimális szám hexadecimális megfelelőjét. Arra ügyeljünk, hogy $10_{(10)}$ -től a számjegyeket betűk helyettesítik, ezért a $11_{(10)} = B_{(16)}$.

$$667_{(10)} = 29B_{(16)}$$

Hexadecimális számot úgy tudunk **közvetlenül bináris számra** konvertálni, hogy vesszük a hexadecimális szám számjegyeit balról jobbra és egyenként átírjuk az 2. táblázat adatainak megfelelően.

Példa:

$$29B_{(16)} = ?_{(2)}$$

$$2_{(16)} = 10_{(2)}, 9_{(16)} = 1001_{(2)}, B_{(16)} = 1011_{(2)}$$

Az egyenkénti számkonvertálás eredményét balról jobbra összeolvassuk:

$$29B_{(16)} = 1010011011_{(2)}$$

Végezetül, **binárisból** úgy állíthatunk **elő hexadecimális számot**, hogy a bináris szám számjegyeit jobbról kezdve négyes csoportokba soroljuk, a bal szélén a szükséges nullákkal kiegészítjük a számot. Majd pedig minden négyes csoportot a neki megfelelő 2. táblázatban lévő hexadecimális számmal helyettesítünk.

Példa:

$$1010011011_{(2)} = ?_{(16)}$$

$$0010 \ 1001 \ 1011_{(2)}$$

$$0010_{(2)} = 2, 1001_{(2)} = 9, 1011_{(2)} = B$$

Az egyenkénti számkonvertálás eredményeként kapott számjegyeket balról jobbra összeolvassuk:

$$1010011011_{(2)} = 29B_{(16)}$$

2.2. Számok ábrázolása

Az informatika területén **három számábrázolási módot** különböztethetünk meg, így beszélhetünk:

- fixpontos ábrázolásról,
- lebegőpontos ábrázolásról
- és BCD számábrázolásról.

Bár a számítógép végső soron bináris számokkal dolgozik, az egyszerűség kedvéért decimális számrendszerben mutatjuk be ezeket az ábrázolási módokat.

Fixpontos ábrázolás

A **fixpontos számban** a tizedesvessző (angolban tizedespont) helye rögzített. Ha a vessző a szám jobb szélén található, akkor egész számról beszélünk és a tizedesvesszőt nem kell kiírni (123,0 helyett 123). Fixpontos törtszámnál a **tizedesjegyek száma rögzített**. Ha például két tizedes pontosságig írjuk ki a számokat, akkor két tizedesjegyes fixpontos számábrázolást használunk és ebben az esetben mindig két tizedes pontosságig kell a számot kiírni (123,50, 123,00).

A számítógép a számban nem jelöli közvetlenül a tizedesvessző helyét, hanem egy külön helyen tárolja, hogy az adott szám jobb oldalán hány számjegy tekintendő tizedesjegyek. A szám előjelét előjelbit tárolja, amelynek értéke 0, ha szám pozitív, 1, ha a szám negatív.

Lebegőpontos ábrázolás

A **lebegőpontos számábrázolásnál** az elnevezés arra utal, hogy a tizedesvessző változó helyen található. Ez az ábrázolás az $m \cdot q^k$ hatványkitevős felírást jelenti, ahol q a számrendszer alapszáma, m a mantissza és k a karakterisztika.

Példa: a 667 szám lebegőpontos ábrázolása (decimális számrendszerben): $6,67 \cdot 10^2$, a 0,667 lebegőpontos ábrázolása: $6,67 \cdot 10^{-1}$.

Gyakran találkozhatunk a számok mE_k alakban történő felírásával is, ahol a mantissza és karakterisztika elválasztására egy E betűt használnak ($667=6,67E2$, $0,667=6,67E-1$).

A lebegőpontos számokat a számítógép mindig **normalizált alakban** tárolja (decimális számrendszerben ez egy 1 és 10 (illetve -1 és -10) közötti szám szorozva a 10 valamilyen hatványával). Ténylegesen csak a mantissza és a karakterisztika kerül tárolásra (667 tárolása lebegőpontos módon: 667 és 2). Akárcsak a fixpontos számábrázolásnál, itt is előjelbit jelzi a szám pozitív (0) vagy negatív (1) voltát.

A lebegőpontos számokkal végzett műveleteket a számítógépek lassabban végzik el, mint a fixpontos műveleteket. Ennek oka, hogy a számokat először közös nevezőre kell hozni, majd a művelet elvégzése után normalizálni kell tizedesjegyeket.

BCD (binárisan kódolt decimális)

BCD (binárisan kódolt decimális) ábrázolás esetén megegyezik a **szám jegyeinek** kettes számrendszerbeli alakjainak **sorozatával**.

Példa: A 99 decimális szám bináris alakja 1100011 lenne, de ebben az esetben a decimális szám minden egyes jegyét külön-külön tároljuk, azaz nem a szám értéke, hanem az „alakja” kerül letárolásra. A $9_{(10)}$ bináris alakja $1001_{(2)}$ és ezért: $99_{(10)}$ BCD formája 0000 1001 0000 1001₍₂₎.

2.3. Szöveg kódolása

A háttértáron tárolt dokumentumok betűket, számjegyeket, írásjeleket, valamint egyéb speciális jeleket tartalmaznak; ezeket összefoglalóan karaktereknek nevezzük. A **karakterek fajtái**:

- alfabetikus karakter: az ábécé kis- és nagybetűi,
- numerikus karakter: a számjegyek 0-9,

- alfanumerikus karakter: az előző kettő együtt.

A számítógép a karaktereket is bináris formában tárolja, azaz minden egyes karakternek egy bináris szám feleltethető meg. A megfeleltetés módját **kódolási szabványok** rögzítik:

- ASCII (American Standard Code for Information Interchange): Eredetileg 7 bites kódolási rendszer, a karaktereket egy kódtáblázat alapján sorszámokkal látja el (például az „A” betűnek a 65 felel meg). A 7 bites kódolásnak köszönhetően legfeljebb 128 karakter megkülönböztetését teszi lehetővé ($2^7=128$) – ami az angol nyelv karakterkészletét tekintve elegendő, de ha figyelembe vesszük a nemzeti karaktereket is (például a magyar ékezetes betűket), akkor már nem. A jelkészlet tartalmaz vezérlő jeleket is (escape, backspace stb. (lásd: 3. táblázat).

A kiterjesztett ASCII 8 bites kódolást használ olyan módon, hogy az első 128 sorszámhoz minden nyelvben ugyanazok a (7 bites ASCII szabványnak megfelelő) karakterek tartoznak (alap karakterkészlet), míg a 129-255 közötti sorszámokhoz (kiegészítő karakterkészlet) az egyes nyelvek (szabványokban rögzített) eltérő karakterei kaptak helyet – ezeket a kiegészítéseket nevezzük kódlapnak.

- Unicode: A manapság legjelentősebb és legelterjedtebb kódolási rendszer az ASCII filozófiáját követi, de a 2 byte-os kódolási rendszerének köszönhetően (2 byte = 16 bit) összesen 2^{16} , azaz körülbelül 65 ezer karakter különböztethető meg a segítségével. A hatalmas méretű táblában – többek között – a <https://unicode-table.com/hu/> weboldalon tudunk különböző szempontok szerint karaktereket keresni.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

3. táblázat. ASCII jelkészlet (részlet). Az oldalléc és a fejléc összeolvasva a karakter hexadecimális kódja, vezérlőjelek szürke háttérrel, megjeleníthető karakterek fehér háttérrel jelölve. Forrás: Wikipedia 2 (2019)

2.4. Logikai értékek, logikai műveletek

A számítógépek működésének alapjait a **matematikai logika** képezi. A számítógép kétértékű logikai rendszert használ. Ebben a rendszerben minden változó csak kétféle értéket vehet fel: igaz (1) vagy hamis (0); ebből az is következik, hogy hamis < igaz. A **Boole-művelet** egy logikai értékhez egy logikai értéket rendel hozzá. A műveletet eredményének megadása igazságtáblával történik: (Megjegyzés: A műveletek nem csak változókra, hanem állításokra is értelmezhetők.)

- **Logikai ÉS művelet**, konjukció ($A \text{ AND } B$, $A \wedge B$, $A \cdot B$): értéke akkor és csak akkor igaz, ha mindkét argumentumának értéke igaz.

A	B	$A \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0

- **Logikai (MEGENGEDŐ) VAGY művelet**, diszjunkció ($A \vee B$, $A+B$): értéke akkor igaz, ha legalább az egyik argumentum értéke igaz.

A	B	$A \vee B$
1	1	1
1	0	1
0	1	1
0	0	0

- **Logikai KIZÁRÓ VAGY művelet**, negáció, antivalencia ($A \oplus B$, $A \vee B$, $A \oplus B$): értéke akkor igaz, amikor csak az egyik argumentum értéke igaz.

A	B	$A \vee B$
1	1	0
1	0	1
0	1	1
0	0	0

- **Logikai NEM művelet**, tagadás ($\text{NOT } A$, $\neg A$, $\bar{A}$): értéke mindig az argumentum ellentettje.

A	$\neg A$
1	0
0	1

Egymás után akár több logikai műveletet is végezhetünk. Akárcsak a számtani műveleteknél, az összetett logikai műveletek végrehajtásának is van **sorrendje**:

- Először a zárójelek közé írt műveleteket kell végrehajtani.
- Ezután következnek a NEM műveletek.
- Ezt követi az ÉS műveletek végrehajtása.
- Végül a VAGY műveletek zárják a sort.

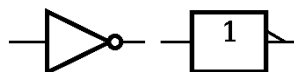
Példa: Mi lesz a következő logikai műveletek eredménye, ha $A=1$, $B=0$? $\overline{(A \cdot B)} + (A \oplus B)$

A	B	$A \cdot B$
1	0	0
A	B	$A \oplus B$
1	0	1
A	B	$\overline{(A \cdot B)}$
1	0	1
A	B	$\overline{(A \cdot B)} + (A \oplus B)$
1	0	1

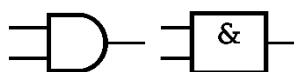
2.5. Kapuk

A számítógép digitális áramköreinek működése leírható logikai műveletek segítségével. Az alapvető **logikai műveleteket megvalósító áramköröket kapuáramköröknek** nevezzük. Az áramkörökben a bemeneti és kimeneti 0 és 1 értékeket feszültségszintek képviselik. A logikai kapuk jelölése:

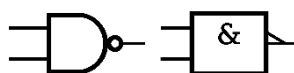
- **Negálás, NOT** kapu (inverternek is nevezik)



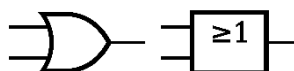
- **ÉS (AND)** kapu



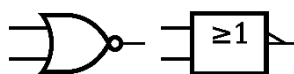
- **Negált ÉS (NAND)** kapu



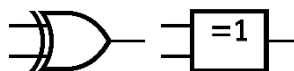
- **Megengedő VAGY (OR)** kapu



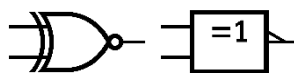
- **Negált megengedő VAGY (NOR)** kapu



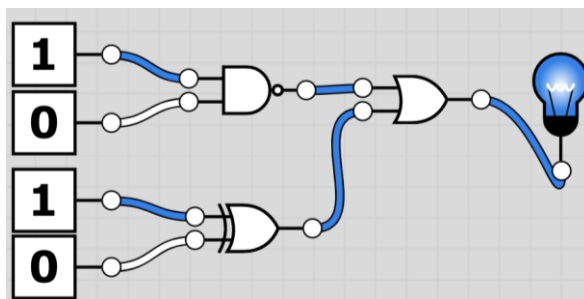
- **Kizáró VAGY (XOR)** kapu



- **Negált kizáró VAGY (XNOR)** kapu



Példa: Állítsuk össze a $\overline{(A \cdot B)} + (A \oplus B)$ logikai műveletet elvégző áramkört és nézzük meg, hogy mi lesz az eredmény, ha $A=1$, $B=0$! (Megoldás: 3. ábra)



3. ábra. Logikai áramkör.

(a rajz a <https://logic.ly/demo> weboldalon készült)

2.6. Ellenőrző kérdések

1. Hogyan határozható meg egy q-alapú számrendszer? [20]
2. Mi a decimális, bináris és hexadecimális számrendszer jelentősége? [20]
3. Sorolja fel a decimális, bináris és hexadecimális számrendszer jelkészletét! [20]
4. Hogyan alakítható át bináris vagy hexadecimális szám decimális számmá? [21]
5. Hogyan alakítható át decimális szám bináris számmá? [21]
6. Hogyan alakítható át decimális szám hexadecimális számmá? [21]
7. Hogyan alakítható át hexadecimális szám bináris számmá? [22]
8. Hogyan állíthatunk elő binárisból hexadecimális számot? [22]
9. Mit jelent a fixpontos számábrázolás? [23]
10. Mit jelent a lebegőpontos számábrázolás? [23]
11. Mit jelent a BCD? [23]
12. Hogyan történik a szöveg kódolása? [23]
13. A kétértékű logikában milyen értékeket vehet fel egy változó, vagy állítás, és mi jellemző az értékekre? [24]
14. Mit jelent a logikai ÉS művelet, hogyan néz ki az igazságtáblázata? [24]
15. Mit jelent a logikai (megengedő) VAGY művelet, hogyan néz ki az igazságtáblázata? [25]
16. Mit jelent a logikai KIZÁRÓ VAGY művelet, hogyan néz ki az igazságtáblázata? [25]
17. Mit jelent a logikai NEM művelet, hogyan néz ki az igazságtáblázata? [25]
18. Mi a logikai műveletek végrehajtásának sorrendje? [25]
19. Mi lesz a $\overline{(A \cdot B)} + (A \oplus B)$ logikai műveletek eredménye, ha $A=1$, $B=0$? [25]
20. Mik a kapuáramkörök? [26]
21. Hogyan jelöljük a negáló, az ÉS és a VAGY kaput? [26]
22. Hogyan jelöljük a megengedő VAGY és a negált megengedő VAGY kaput? [26]
23. Hogyan jelöljük a kizáró VAGY és a negált kizáró VAGY kaput? [26]
24. Állítsa össze a $\overline{(A \cdot B)} + (A \oplus B)$ logikai műveletet elvégző áramkört! Mi lesz az eredmény, ha $A=1$, $B=0$? [26]

3. HARDVER ISMERETEK

Elvárt tanulási eredmények

Tudás

- Ismeri a hardver fogalmát
- Felsorolja a Neumann-elveket és a Neumann-elvű számítógép alapvető funkcionális egységeit
- Érti a belső programvezérlés lényegét
- Átfogóan ismeri a központi feldolgozó egység és a memória jellemzőit, részeit, feladatait
- Felsorolja a betöltő-dekódoló-végrehajtó ciklus lépéseit
- Átfogóan ismeri az input, output, valamint input-output perifériák alapvető fajtáit és fő jellemzőit

Képesség

- Hierarchiába rendezi a memóriákat az elérési idő, tárolókapacitás és egységnyi ár alapján
- Összeállít egy Neumann-elvű számítógépet

Attitűd

- Érdeklődik az informatika fejlődésével megjelenő új eszközök és az azok által nyújtott szolgáltatások iránt
- Nyitott az új technológiák alkalmazására a mindennapokban és saját szakterületén
- Figyelembe veszi és elfogadja a számítógép alkotóelemeinek lehetőségeit, képességeit

Autonómia, felelősség

- Szakmai iránymutatás alapján, másokkal együttműködve javaslatokat fogalmaz meg adott feladat ellátására alkalmas számítógépre és perifériákra

3.1. A hardver fogalma

A **hardver** (hardware) a gép, a számítástechnikai berendezés. Hardvernek nevezzük a számítógépet alkotó elektronikai és mechanikai alkatrészek együttesét; ezt számítógép-konfigurációnak is szokás nevezni. A hardver a számítástechnikai termék azon része, amely kézzelfogható, nehezen reprodukálható.

A hardver biztosítja a számítógépen futó alkalmazások, programok zavartalan működését és **meghatározza**:

- az alkalmazható szoftverek körét (az adott géptípuson milyen programok futtathatók),
- a sebességet, teljesítményt (adott programot milyen gyorsan tud végrehajtani a számítógép),
- valamint a tárolókapacitást (hány [mega/giga]byte-nyi programot és adatot tud tárolni a gép).

3.2. A Neumann-elvű számítógép felépítése

A ma működő számítógépek működési elvét a magyar származású tudós, Neumann János dolgozta ki az 1950-es években, ezért az ilyen elven működő berendezéseket **Neumann-gépeknek** nevezzük (4. ábra).

A **Neumann-elvek**:

- Soros utasítás-végrehajtás (Az utasítások végrehajtása időben egymás után történik. Ennek ellentéte a párhuzamos utasítás-végrehajtás, amikor több utasítás egyidejűleg is végrehajtható.);
- Kettes (bináris) számrendszer használata;
- Belső memória (operatív tár) használata a program és az adatok tárolására;
- A számítógép meghatározott funkciókat (feladatokat) végrehajtó hardver egységekkel rendelkezik;
- Teljesen elektronikus működés (A számítógép központi egységében nincsenek mozgó alkatrészek, ellentétben a régi mechanikus, vagy elektromechanikus számológépekkel);
- Széleskörű felhasználhatóság, alkalmasság bármilyen adatfeldolgozási feladatra.

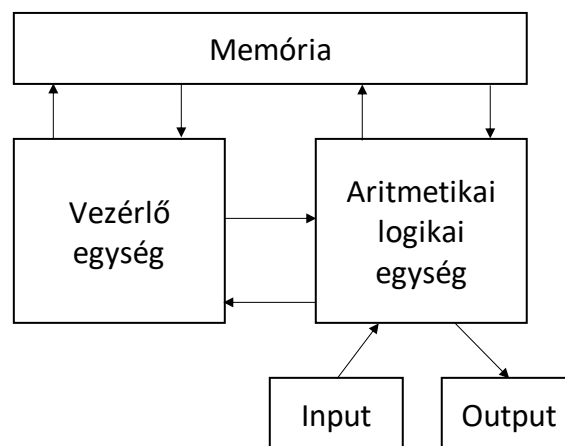
A **Neumann-elvű számítógépek alapvető funkcionális egységei, hardvere**:

1. Központi egység (Central Unit)

- Központi feldolgozó egység (Central Processing Unit; processzor vagy mikroprocesszor), amelynek részei:
 - központi vezérlő egység (Control Unit),
 - aritmetikai-logikai egység (Arithmetical-Logical Unit),
 - matematikai társprocesszor (Floating Point Unit),
 - regiszterek,

- gyorsítómemória (cache),
 - Operatív tár ([központi] memória)
2. Perifériák
- Háttértárak (merevlemez, félvezető alapú meghajtó, optikai háttértár stb.)
 - Input perifériák (billentyűzet, egér, lapolvasó stb.)
 - Output perifériák (monitor, nyomtató, hangszóró stb.)

A számítógépek **belső programvezérlésű gépek**. A belső programvezérlés lényege, hogy a számítógép memóriája nem csak az adatokat, hanem a gépet működtető programot is tárolja. A tárolt program bonyolult vezérlést tesz lehetővé, így a gép tetszőleges ideig önállóan képes működni, kezelői beavatkozás nélkül. Külső programvezérlésű gépek a számológépek, ahol az utasításokat kívülről (egy billentyűzetről) visszük be a géphez, azok nem tárolódnak.



4. ábra. A Neumann-elvű számítógép felépítése.
 Forrás: Wikipedia 3 (2018) alapján a szerző szerkesztése.

3.3. A központi egység

A számítógép központi egységéhez a központi feldolgozó egység és az operatív memória tartoznak.

A központi feldolgozó egység (processzor)

A számítógépek nagyon bonyolult feladatokat is végre tudnak hajtani, feltéve, hogy azok olyan egyszerű lépések sorozatára vannak felbontva, melyeket a számítógép értelmezni tud. Ilyen egyszerű, értelmezhető lépéseknek számítanak például az adatmozgatások, beolvasások, kiírások és az alapvető aritmetikai és logikai műveletek.

A **központi feldolgozó egység** (Central Processing Unit, processzor, CPU) a számítógép „agya”, a következő feladatokat látja el:

- a gép irányítása, illetve a feldolgozási folyamat vezérlése,
- az adatok feldolgozása,
- az adatforgalom vezérlése a külvilág felé, illetve a külvilág felől.

A CPU **több különálló részegységből áll:**

- A vezérlőegység (Control Unit) feladata az utasítások beolvasása a központi memóriából és az utasítások típusának megállapítása.
- Az aritmetikai-logikai egység (Arithmetical-Logical Unit) a program utasításainak végrehajtásához szükséges műveleteket végzi (összeadás, kivonás, logikai és, vagy stb.).
- A **matematikai társprocesszor**, vagy **lebegőpontos egység** (Floating Point Unit), a lebegőpontos számokkal való számításokat végzi el nagyon gyorsan. A „társprocesszor” elnevezés onnan ered, hogy kezdetben a fő processzortól különálló részegység volt, ma már a processzoron belül található.
- Kis tárolókapacitású memóriát is tartalmaz, amelyben részeredményeket és vezérlőinformációkat tárol. Ez a memória többfajta ún. **regiszterből** áll és mindegyiknek meghatározott mérete és funkciója van. A regisztereket nagy sebességgel lehet olvasni és írni, mivel a processzoron belül vannak. Néhány fontosabb regiszter:
 - az utasításregiszter az éppen végrehajtás alatt álló utasítást tartalmazza,
 - az utasításszámláló regiszter a sorok következő utasítás címét tárolja,
 - az akkumulátorregiszter az aritmetikai-logikai műveletek operandusait (azaz a műveletek tárgyát képező értékeket vagy azok eredményét) tárolja,
 - az állapotregiszterek, vezérlőregiszterek vezérlő- és ellenőrző jeleket tárolnak.
- A **gyorsítómemória** vagy gyorsítótár (cache) átmentet képez a processzor (nagyon gyors) regiszterei és a számítógép (viszonylag lassú) operatív memóriája között. A számítógép műveletvégző sebességét javítja azzal, hogy adatokat, utasításokat tárol.

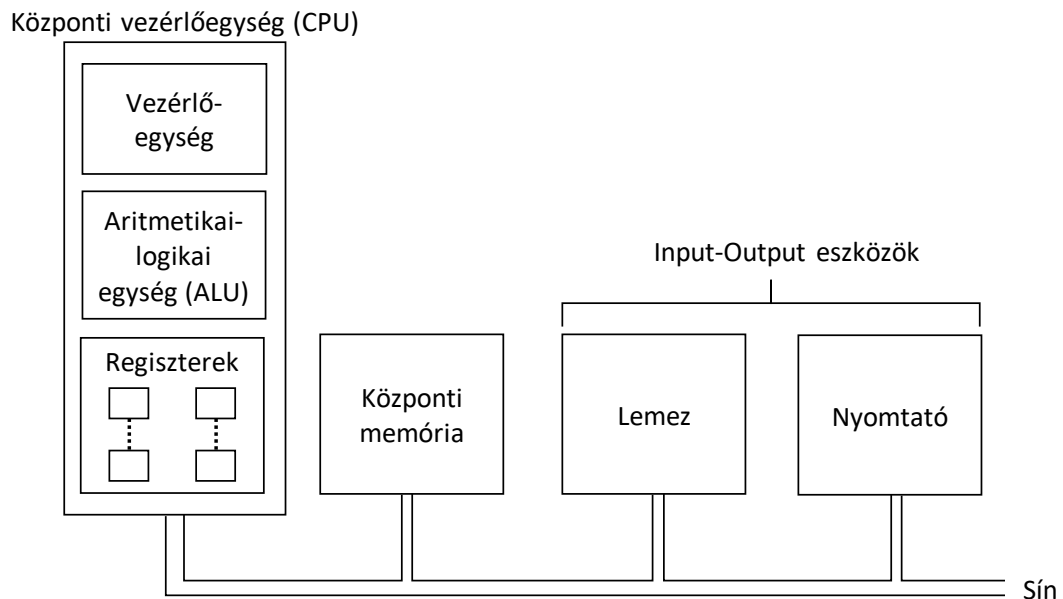
Egy számítógép típusát alapvetően a benne lévő **processzor típusa** határozza meg. Ha két gép azonos (vagy kompatibilis [egymással megegyezően működő]) processzortípust tartalmaz, akkor általában ugyanazok a programok futtathatók rajtuk, míg eltérő processzor típusok esetén ez már nem feltétlenül teljesül.

A számítógép részeinek összehangolt működését biztosítja az ún. órajelgenerátor. Az egy másodperc alatti órajelek számától függ a gép működési sebessége. A **processzor órajelét** általában olyan számokkal jellemzik, mint: 3,5 GHz (gigahertz), ami azt jelenti, hogy egy másodperc alatt $3200 \times 1\,000\,000 = 3\,500\,000\,000$ órajel ütés történik.

Szintén fontos jellemző, hogy **mekkora memóriaterületet tud megcímezni** a processzor. A 32 bites 2^{32} byte (= 4GB), míg a 64 bites elvileg maximum 2^{64} byte (≈ 18 exabyte) memóriát tud elérni, kezelni.

A processzort a számítógép többi részegységével **sínrendszer** (másképpen buszrendszer) köti össze (5. ábra). Megkülönböztetünk cím-, adat- és vezérlősíneket: a címbusz a memória megcímezéséhez, az adatbusz az adatok továbbításához, a vezérlőbusz a számítógép

összehangolásához szükséges. Ez a sínrendszer lehet a processzort tekintve külső, amely összekapcsolja azt a memóriával és az input-output perifériákkal, illetve lehet belső, amely a processzor részegységeit kapcsolja össze.



5. ábra. Egy egyszerű, egy processzorból és két input-output egységből álló számítógép felépítése. Forrás: Tanenbaum (2006) alapján a szerző szerkesztése

A processzor az utasításokat apró lépések sorozataként hajtja végre. A lépések – amelyeket **betöltő-dekódoló-végrehajtó ciklusnak** is neveznek – a következők:

1. A soron következő utasítás beolvasása a memóriából az utasításregiszterbe;
2. Az utasításszámláló beállítása a következő utasítás címére;
3. A beolvasott utasítás típusának meghatározása;
4. Ha az utasítás memóriabeli szót (a memóriában tárolt adatot) is használ, akkor e helyének megállapítása;
5. Ha szükséges, az adat (a memóriabeli szó) beolvasása a CPU egy regiszterébe;
6. Az utasítás végrehajtása;
7. Vissza az 1. pontra és a következő utasítás végrehajtásának megkezdése.

Azt a folyamatot, amikor a processzor sorra egymás után hajtja végre a memóriában tárolt gépi utasításokat, a **program futásának** nevezzük. A számítógép a bekapcsolás pillanatától kezdve egészen a kikapcsolásig folyamatosan programot futtat, gépi utasításokat hajt végre. Ez akkor is így van, amikor a gép látszólag áll, nem csinál semmit.

A memória

A processzor önmagában nem elegendő ahhoz, hogy a számítógép sikeresen végrehajthassa a feladatát. Az adatfeldolgozáshoz olyan tárolóhely is szükséges, ahol a feldolgozáshoz nélkülözhetetlen adatokat el lehet helyezni és ahol a számítógép ezekhez gyorsan hozzáférhet. A számítógép által végrehajtandó utasításokat, valamint az utasításokhoz szükséges adatokat a gép működése közben a **memória** tartalmazza (amit

elsődleges, központi vagy operatív tárnak is szoktak nevezni). A memória nem csak adatokat, hanem a gépet működtető utasítássorozatot (programot) is tárolja.

A memória a processzor munkasztala, ide teszi és innen veszi azt, amire munkája közben szüksége van. A számítógép memóriájában az információk kettes számrendszerben, 0 és 1 formában tárolódnak.

Attól függően, hogy milyen feladat megoldásához használják, **többféle memória** létezik:

- RAM (Random Access Memory – közvetlen hozzáférésű, írható és olvasható memória): ez szolgál az általános feladatok megoldásához szükséges adatok tárolására, amikor a megjegyzendő információ változhat. Tartalma a gép kikapcsolásakor elveszik.
- ROM (Read Only Memory – csak olvasható memória): a „gyárilag” beírt információt kikapcsolás után is megőrzi, átírni azt nem lehet. Nem változó programok és adatok biztonságos tárolására fejlesztették ki.
- PROM (Programmable Read Only Memory – programozható csak olvasható memória): olyan, mint a ROM, de a gyártás után még nem tartalmaz semmit. Minden egyes felhasználó saját programot és adatokat helyezhet el benne egy ún. beégető készülék segítségével. A beégetés elektromos impulzusokkal történik. A PROM tartalma később nem írható felül.
- EPROM/EEPROM (Erasable Programmable Read Only Memory – törölhető, programozható csak olvasható memória): olyan, mint a PROM, de tartalma többször módosítható. Az EPROM-nál először ultraibolya fénnel törlik a régi tartalmat, majd EPROM-égetővel viszik be az újat. Az EEPROM elektromosan törölhető, a beírás is elektromos impulzusokkal történik.

A memória legfontosabbnak tartott **jellemzője a tárolókapacitás**, amit általában gigabájtban adunk meg. Minél több van belőle annál jobb, de ügyeljünk arra, hogy mind a processzor (lásd az előző alfejezetet), mind az operációs rendszer képes legyen a teljes területét elérni. (Ha 4GB-nál több RAM memóriánk van, ahhoz 64 bites processzort és 64 bites operációs rendszert használunk!

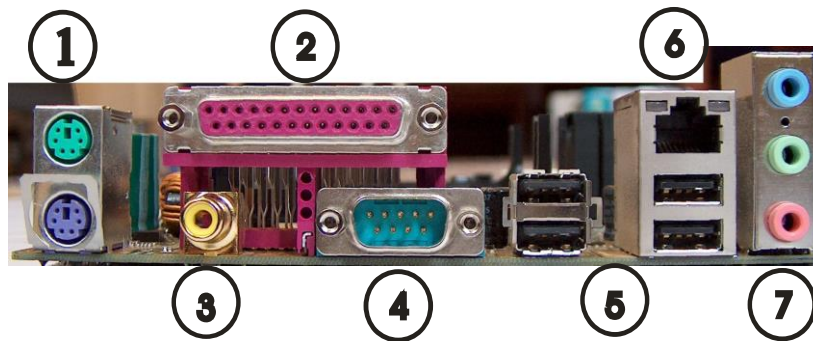
3.4. Perifériák

Perifériáknak nevezzük azokat a központi feldolgozó egységhez csatolt eszközöket, amelyek a számítógépet összekapcsolják a külvilággal, illetve amelyek lehetővé teszik programok, adatok hosszabb idejű tárolását. A perifériákat alapvetően **három csoportba** sorolhatjuk:

- Bemeneti (vagy input) perifériák: Az adat a környezetből a bemeneti periférián át a központi egység felé áramlik. Az input perifériák közé tartozik például: billentyűzet, egér, követőgolyó (trackball), digitalizáló tábla, botkormány, mikrofon, érintéses képernyő (touchscreen), kézi- és lapolvasó (szkenner).
- Kimeneti (vagy output) perifériák: Az adat a központi egységből a kimeneti periférián át a környezet felé áramlik. A számítógép ezeken keresztül közli az eredményeket, a kimenő adatokat. Az output perifériák közé tartozik például: képernyő (monitor), nyomtató (printer), rajzgép (plotter), hangszóró és fejhallgató.

- Be-kimeneti (vagy input/output) perifériák: Olyan egységek, amelyekről a számítógép olvasni képes (bemenet), és amelyekre írni is tud (kimenet). Ide tartoznak a különböző beépített és hordozható háttértárak, például: merevlemez meghajtó, blu ray lemez és pendrive.

A perifériák vagy be vannak építve a számítógépbe, vagy valamilyen ún. **porton** keresztül csatlakoznak a buszrendszerhez (6. ábra). (A port magyarul kapu, de általában az angol megnevezést használjuk.) A legelterjedtebb csatlakozó felület az USB (Universal Serial Bus: univerzális soros busz), amelyet ma már szinte minden periféria támogat.



6. ábra. Portok: 1: PS/2, 2: LPT, 4: COM, 5: USB, 6: UTP, 3 és 7: audio be/ki.

3.5. Input/output perifériák: háttértárak

A háttértárak tartós tárolásra szolgálnak, tartalmuk a számítógép kikapcsolása után is megmarad. **Három alapvető fajtája** az alkalmazott technológiától függően:

- a mágneses háttértárak,
- a tartós állapotú (vagy másnéven félvezető alapú) háttértárak,
- az optikai háttértárak.

Mágneses háttértárak

A **merevlemezlemez** meghajtó a leggyakrabban előfordul mágneses elven működő háttértár. Ugyanebbe a kategóriába tartozik a **szalagos meghajtó** (streamer), amellyel elsősorban vállalati környezetben találkozhatunk (adatmentésre), vagy a **hajlékonylemez meghajtó** (floppy), amely mára már a kis tárolókapacitása és sebessége miatt elavult és teljesen el is tűnt a korszerű számítógépekből. Ezek az eszközök arra az elvre épülnek, hogy a lemez (vagy szalag) mágnesezhető bevonatból készült felületét az író-olvasó fej íráskor mágnesezi, olvasáskor pedig a felület mágnesességének változását érzékeli.

A **merevlemez meghajtóban** (hard disk drive: HDD) akár több lemez is lehet egymás felett és ezek közé nyúlnak be az író-olvasó fejek. A fejek együtt mozognak (léteznek kísérletek arra, hogy a fejek akár külön-külön is mozgathatók legyenek), így az egymás alatt lévő sávok (a cilinderek) egyszerre érhetők el. A lemezek többezer fordulat/perc sebességgel forognak, a fejek nem érnek hozzájuk.

A merevlemezek fontosabb jellemzői:

- a tárolókapacitás,
- a maximális adatátviteli (írási/olvasási) sebesség,

- a lemezek fordulatszáma,
- a csatolófelület típusa, amellyel a buszrendszerhez csatlakoznak.

Tartós állapotú háttértárak, memóriakártyák és pendrive-ok

A **tartós állapotú háttértárakat** félvezető alapú, vagy szilárdtest meghajtónak is szokták nevezni (angolul: solid-state drive, SSD). Ez egy mozgó alkatrészek nélküli adattároló, amely memóriában tárolja az adatokat. A felhasznált memóriák különbözőek lehetnek, de képesek arra, hogy kikapcsolt állapotban is megőrizzék a tartalmukat.

Ezeknek a merevlemezeket egyre több területen kiváltó, helyettesítő háttértáraknak a **fő előnyei** a merevlemezekhez képest:

- A gép bekapcsolása után gyakorlatilag azonnal használhatók.
- Nincs bennük mozgó alkatrész, ami miatt kevésbé sérülékenyek.
- Az olvasási és írási várakozási idő jóval kisebb, hiszen nem kell megvárni, míg a lemez megfelelő része az író-olvasó fej alá fordul.
- Az írási és olvasási sebesség és lényegesen nagyobb és szinte állandó az írási-olvasási teljesítmény.
- Mivel nincsenek mozgó alkatrészek, teljesen zajtalan a működése.
- Szintén a mozgó alkatrészek hiányából fakadóan, jobban ellenáll a mechanikai hatásoknak.
- Alacsony áramfelvétel, kisebb hőtermelés, kis fizikai méret és tömeg.

Hátránya:

- Az memóriacellák újraírásának száma korlátozott. Ez elvileg rövidebb élettartamot jelent a merevlemezekhez képest, ugyanakkor átlagos használat mellett a mai SSD-k több mint tíz évig is hibátlanul működhetnek, így ez ma már nem igazán jelent problémát.
- Fajlagosan drágábbak, mint a merevlemez, de az árellő egyre csökken.

Legfőbb jellemzői:

- tárolókapacitás;
- a várható élettartam (függ a tárolókapacitástól, az újraírási ciklusok számától és az alkalmazott memóriacellák fajtájától);
- a maximális adatátviteli (írási/olvasási) sebesség;
- a csatolófelület típusa, amivel a buszrendszerhez csatlakozik.

A különböző **memóriakártyák**, illetve ún. **pendrive-ok** (memóriakulcsok) szintén ebbe a körbe tartoznak, tehát félvezető alapú, az EEPROM-hoz hasonló elektronikus háttértároló eszközök. Míg a memóriakártyákat elsősorban okostelefonokban, digitális kamerákban vagy egyéb hordozható eszközökben használják, addig a pendrive-ok a számítógépek közötti adathordozásra használt hajlékonylemezeket váltották le. Nagy megbízhatóságúak, jobban ellenállnak a környezeti hatásoknak, mint más adattároló eszközök.

A memóriakártyáknak többféle változata létezik (például CompactFlash, SmartMedia, Multimedia Card, Secure Digital Card, Memory Stick, xD-Picture Card) amelyek elsősorban fizikai méretükben és alakjukban különböznek egymástól. Ezeket a kártyákat kártyaolvasóba helyezve csatlakoztathatjuk a számítógépünkhöz. A pendrive-ok USB porton keresztül csatlakoztathatók a számítógéphez.

Optikai háttértárak

Az **optikai elven működő adattárolóknál** a meghajtó egység lézersugár segítségével olvassa és írja a lemezen kódolt információt. A lemezek az adatot a felületükre csigavonalban (spirálban) felírva tartalmazzák. A lemez felülete völgyek és dombok egyenletes sűrűségű sorozatából áll. Olvasáskor az olvasó a vele szemben forgó lemezsávra fókuszált lézersugarat irányít. Ezt a sugarat a sáv megvilágított völgyei szétszórják, dombjai viszont, amelyek a fémfelület síkjában vannak, visszatükrözik. A visszatükrözött jelek a lézeres olvasó fényérzékelőjébe jutnak. Egy áramkör aszerint, hogy az érzékelő a lemez egyes helyeiről visszatükröződő sugarat észlelt-e vagy nem, nullákat vagy egyeseket állít elő. A lemez a meghajtóból eltávolítható.

Íráskor előformázott spirális barázda vezeti az írólézert az írás megkönnyítésére. Az író lézer apró barázdákat éget a lemez felületére. Újraírható lemez esetén ez az állapot többször, egyszer írható lemeznél csak egyszer változtatható meg.

Fajtái – CD, DVD és blu-ray – között a legalapvetőbb különbség a tárolókapacitás nagysága.

Legfőbb jellemzői:

- a maximális tárolókapacitás (CD 700MB, DVD 9GB, Blu-ray 50GB),
- a maximális adatátviteli (írási/olvasási) sebesség,
- írható-e és hányszor,
- a csatolófelület típusa, amivel a buszrendszerhez csatlakozik.

Az optikai háttértárakat nem csak adatok, hanem zenék (CD) és filmek (DVD, blu-ray) tárolására, forgalmazására is használják. De, annak következtében, hogy a SSD-k és pendrive-ok egyre megbízhatóbbak és olcsóbbak, továbbá a szélessávú internet is mind több helyen érhető el és egyre terjednek a streaming szolgáltatások (szórakoztató tartalmak interneten keresztüli igénybevétele), az optikai tárolók ma már nem alapvető eszközei a számítógépeknek, a használatuk visszaszorulóban van.

3.6. A memóriák hierarchiája

Miután volt szó a processzorban található regiszterekről, a gyorsítótárról, az operatív memóriáról és a háttértárakról, nézzük, hogy hogyan viszonyulnak ezek egymáshoz. A viszonyt – a **memóriák hierarchiáját** – egy piramis ábrán lehet legjobban szemléltetni (7. ábra):

1. A piramis tetején találhatók a processzor teljes sebességével elérhető regiszterek. Ezek az épen végrehajtott utasítás adatait és eredményeit tárolják.
2. A regiszterek és az operatív memória között található a gyorsítótár (angolul cache), amelynek célja a számítógép gyorsítása azzal, hogy különböző technikák alkalmazásával áthidalja a kétfajta memória közötti sebességkülönbséget (például

nagy valószínűséggel sorra kerülő, vagy korábban használt utasításokat, adatokat tárol, így azok gyorsabban hozzáférhetők a processzor számára).

3. Ezt követi az operatív memória, ami azokat a programokat és adatokat tárolja, amelyeket használ a számítógép.
4. A negyedik szintet azok a számítógépbe beépített háttértárak képviselik, amelyek a gyakran használt programokat és adatokat tárolják, akár a gép kikapcsolt állapotában is. Ide tartoznak az SSD és a HDD.
5. A piramis legalsó szintjén található a hordozható, a számítógéphez csak alkalomszerűen csatlakoztatott háttértárak. Ezek a pendrive, az optikai lemezek, a memóriakártyák és a szalagok.



7. ábra. Memóriahierarchia. Forrás: Tanenbaum (2006) alapján a szerző szerkesztése.

A hierarchia szintjein lefelé haladva **három fontos paraméter értéke növekszik**:

- Az elérési idő: A regiszterekben található adatok elérési ideje néhány nanoszekundum, a gyorsítótáraknál már kicsit nagyobb ez az érték. Az operatív memória elérési ideje jellemzően néhány tíz nanoszekundum, a beépített háttértáraknál már jelentősen növekszik az elérési idő nagysága (szilárdtest meghajtóknál kevésbé, a merevlemez meghajtóknál jelentősen) stb.
- A tároló kapacitás: A processzor regiszterei néhány kilobájtosak, a gyorsítótárak mérete néhány megabájt, a központi memória mérete néhány gigabájt, a háttértárak mérete ennél nagyságrendekkel nagyobb.
- Az egységnyi pénzösszegért „megvásárolható” bitek száma: A regiszter, a gyorsítótár és az operatív memória nagyságrendekkel drágább, mint a háttértárak; ugyanazért a pénzösszegért például jóval nagyobb tárolókapacitású merevlemez tudunk vásárolni, mint operatív memóriát.

3.7. Input perifériák

Input perifériák közé soroljuk a billentyűzetet, az egeret, a trackballt, a digitalizáló táblát és a lapolvasót.

A **billentyűzet** rendszerint az írógéphez hasonló, vagyis számok és műveleti jelek mellett az ábécé betűit is tartalmazza, emellett számos vezérlő funkciót is ellát. A billentyűzet segítségével tudunk adatokat bevenni, illetve gépet vezérelni. A billentyűzettel nem közvetlenül a processzort vezéreljük, hanem a gépen futó programmal kommunikálunk. Az, hogy egy adott billentyű leütésére a gép hogyan reagál, mindig az éppen futó programtól és annak pillanatnyi állapotától függ. A fontosabb billentyűcsoportok és funkciók a következők:

- az ábécé betűi (a-z),
- számjegyek (0-9),
- pozicionáló billentyűk (mozgás valamilyen irányba, ugrás: Home, End, lapozás: Page up, Page down),
- szövegszerkesztő billentyűk (törlés: Delete, Backspace; felülíró/beszúró mód váltása: Insert),
- vezérlő billentyűk (valamilyen funkcióba való belépés, begépelte utasítás végrehajtása, adat beadása: Enter, funkcióból való kilépés, begépelte utasítás, ill. adat érvénytelenítése: Escape),
- funkció billentyűk (szerepük programonként változik; céljuk, hogy a gyakrabban használt funkciókat egyetlen billentyű leütésével érhesük el: F1-F12, például az F1 funkcióbillentyűvel rendszerint a programok súgóját lehet előhívni),
- numerikus blokk (a billentyűzet jobb oldalán található billentyűcsoport, amely számok és műveleti jelek bevitelére, vagy a kurzor pozicionálására használható),
- kapcsoló billentyűk (más billentyűvel együtt, billentyűkombinációban használjuk; kis- és nagybetű váltására: Shift, illetve programtól függő funkciója van: Alt és Ctrl),
- üzemmód váltó billentyűk (beszúrás: Insert, váltás kis- és nagybetű között: CapsLock, numerikus blokk funkcióváltása: NumLock, görgetés megváltoztatása: Scroll Lock; leütésükre egyik üzemmódból másikba vált a gép),
- egyes billentyűzeteken vannak egyéb, az operációs rendszerhez kötődő billentyűk (Windows és Menü gombok: menübe belépéshez, illetve más billentyűkkel kombinálva használható operációs rendszer szolgáltatások).

Az **egér** egy sima felületen szabadon tologatható eszköz, melynek mozgásával a képernyőn egy nyíl, vagy egyéb alakzat mozog. Ezzel célozhatunk a képernyő egyes részeire; objektumok kiválasztása pedig az egéren található gombok lenyomásával lehetséges. Az egér mozgásának érzékelése az egér aljába épített érzékelőkkel történik.

Olyan alkalmazási területeken, ahol nincs hely az egérnek, ott **trackballt** (követőgolyót) használnak, ami az egérhez hasonló elven működik. A különbség csupán annyi, hogy itt az érzékelőt (gömböt) az eszköz tetejébe építik be és kézzel kell forgatni.

A mérnöki rajzokat **digitalizáló táblákkal** lehet bejuttatni a számítógépbe. Ezek felszíne többnyire olyan fólia, amelyet, ha a táblához csatolt hegyes pálcával egy pontján megérintünk, akkor az egy vezérlőprogram által a számítógép képernyőjére is leképeződik.

A **lapolvasók** (szkennerek) szövegek, képek és rajzok digitalizálására és számítógépbe történő bevitelére szolgálnak. A lapolvasó által digitalizált képpontok képállományba kerülnek, amelyet a rajzolóprogramok kezelni tudnak, illetve beépíthetők szöveges dokumentumokba is.

A **síkágys lapolvasónál** a lapot üveglapra helyezzük. A beolvasó fejet egy mechanika állandó sebességgel mozgatja a lap alatt. A **kézi lapolvasónál** a mozgatást kézzel kell végezni. A kézi mozgatás egyenetlen sebessége miatt a kézi lapolvasók igényesebb munkához nem használhatók.

A lapolvasók fontos tulajdonsága a felbontóképesség. Ezt dpi-ben (dot per inch, 1 hüvelykre eső pontok száma, 1 hüvelyk = 2,54 cm) adják meg. Egy lapolvasóban annyi fényérzékelő elem van egy sorban elhelyezve, amennyi a készülék vízszintes felbontóképességének megfelel. (Például, ha a lapolvasó optikailag vízszintesen 300 dpi, függőlegesen 600 dpi felbontóképességű, akkor ez a készülék egy A4-es (21,0 cm × 29,7 cm) képet optikailag 2480 × 7015 képpontra tud felbontani. A beolvasott kép minőségét meghatározza a színmélység is (azaz a színek száma: fekete-fehér, szürke színárnyalatú, vagy színes).

A lapolvasóval beolvasott szöveg mindaddig csak képként kezelhető, ameddig szövegfájlá nem alakítottuk át. Erre a célra a lapolvasókhoz általában karakterfelismerő programot mellékelnek. Az optikai karakterfelismerő (OCR: Optical Character Recognition) a szabványos módon, kézzel írott, géppel írt vagy nyomtatott szöveget ismeri fel és alakítja át a gép számára érthető karakterekké.

3.8. Output perifériák

Az **output eszközök** közé sorolhatók a monitorok, a nyomtatók és a rajzgépek.

Az adatok képernyőn történő megjelenítéséhez szükséges egy buszrendszerre csatlakozó **monitorvezérlő**, továbbá egy, a monitorvezérlőhöz csatlakoztatott **monitor**. A monitorvezérlő, valamint a monitor képességeinek egymással összhangban kell lenniük!

A **monitorok fő tulajdonságai**:

- az alkalmazott technológia: régi katódsugárcsöves (CRT: Cathode Ray Tube), vagy folyadékkristályos (LCD: Liquid Christal Display) megjelenítést használ,
- a megjeleníthető színek (színárnyalatok) száma, színmélység (16 bit: 2^{16} szín, 24 bit: 2^{24} szín 32 bit: 2^{32} szín),
- a maximális (natív) felbontás (hány képpontot képes egyszerre megjeleníteni, milyen felbontásra van optimalizálva),
- a csatlakozó fajtája és típusa (analóg vagy digitális szabványt követ és azon belül melyiket).

A **nyomtatók** általában apró pontokból állítják össze a nyomtatási képet. Minél finomabb a felbontás, vagyis minél apróbb pontokból áll össze a kép, annál jobb a nyomtatás minősége. A felbontást dpi (dot per inch = pont/hüvelyk) egységben mérik.

Átlagminőségnek számít a 300 dpi felbontás, az ennél kisebb felbontás gyenge, nagyobb felbontás jó minőségnek tekinthető.

A fontosabb nyomtatótípusok:

- **Mátrixnyomtató:** A nyomtatófej soronként jobbra-balra halad a papír előtt. A fej általában 8-12 tűt tartalmaz egy oszlopban, amelyek egy festékszalagra ütve írnak a papírra. Szokásos felbontása 100-200 dpi.
- **Tintasugaras nyomtató:** Apró tintacseppeket állít elő spray formájában. A nyomtató a papír megfelelő pontjait elektrosztatikusan feltölti, és a feltöltött helyek vonzzák magukhoz a tintacseppeket. Szokásos felbontása 300-5760 dpi.
- **Lézernyomtató:** Egy forgó hengerre fényérzékeny vezető anyagot visznek fel, amelyet feltöltenek, majd lézersugárral kialakítják az exponálandó képpontokat. A lézersugarat a nyomtató memóriájában már jelenlévő kép vezérli. Egy festékszóró a hengerre festékport visz fel, amely az exponált pontokon megtapad. Ezután a magas hőmérsékletű henger a papírra nyomja, beleégeti a képet. Szokásos felbontása 300-1200 dpi.

A nyomtatás általában fekete festékkel történik, de mindhárom felsorolt nyomtatótípusnál találunk színes tintákkal nyomtató változatokat is.

A **rajzgépek** műszaki rajzok előállítására szolgáló eszközök, amelyekkel tetszőleges folytonos vonalakat lehet előállítani. Létezik több rajztollat használó, színes változatban is. Két alapvető típusa van: a síkágys és a továbbbítóhengeres rajzgép. A síkágys rajzgépnél csak a rajztoll mozog a síkágys felületére rögzített rajzpapír felett hossz- és keresztirányban. A továbbbítóhengeres rajzgépnél a papír és a toll is mozog: a papírt elgördítik a felette keresztirányban oda-vissza járó rajztoll alatt. Míg a síkágys rajzgépbe tehető rajzpapírok méretét a síkágys mindkét irányban korlátozza, addig a továbbbítóhengeres rajzgépek a papírnak csak a keresztirányú méretet szabják meg.

3.9. Ellenőrző kérdések

1. Mi a hardver és mit határoz meg? [29]
2. Sorolja fel a Neumann-elveket! [29]
3. Sorolja fel a Neumann-elvű számítógépek alapvető funkcionális egységeit! [29]
4. Mit jelent a belső programvezérlés? [30]
5. Mik a központi feldolgozó egység fő feladatai, részei és jellemzői? [30]
6. Sorolja fel a betöltő-dekódoló-végrehajtó ciklus lépéseit! [32]
7. Mit értünk a program futása alatt? [32]
8. Mi a memória, milyen fajtái vannak, mivel jellemezhető? [32]
9. Mik a perifériák és hogyan csoportosíthatók? [33]
10. Mire szolgál a port a számítógépben? [34]
11. Milyen mágneses elven működő háttértárak vannak és mivel jellemezhetők? [34]
12. Mik a szilárdtest meghajtók, mivel jellemezhetők, milyen előnnyel és hátránnyal jár a használatuk? [35]
13. Mik a memóriakártyák és pendrive-ok? [35]
14. Milyen optikai elven működő háttértárak léteznek, hogyan működnek és mivel jellemezhetők? [36]
15. Mit jelent a memóriák hierarchiája? [36]
16. A memóriák hierarchiájában milyen három paraméter értéke és hogyan változik? [37]
17. Milyen fontosabb paraméterei vannak a billentyűzetnek? [38]
18. Mi az egér, a trackball és a digitalizáló tábla és hogyan működik? [38]
19. Mi a lapolvasó feladata, hogyan működik és milyen paraméterekkel jellemezhető? [39]
20. Milyen fő paraméterekkel jellemezhető a monitor? [39]
21. Milyen fő változatai vannak a nyomtatóknak és hogyan működnek azok? [39]
22. Mi a rajzgép és hogyan működik? [40]

4. SZOFTVER ISMERETEK

Elvárt tanulási eredmények

Tudás

- Ismeri a szoftver, a szoftvercsomag és a verzió fogalmát
- Felsorolja a szolgáltatások és hozzáférhetőség szerinti szoftver kategóriákat
- Ismeri a szoftverek jogi védelmének formáit és érti azok fontosságát
- Érti a legális szoftverhasználat előnyeit

Képesség

- Elmagyarázza a telepítés, a frissítés és az eltávolítás lényegét
- Besorolja a szoftvert a megfelelő kategóriákba a tulajdonságok alapján

Attitűd

- Elfogadja a szoftver jogi védelmének fontosságát
- Magára nézve kötelezőnek tartja a legális szoftverhasználatot

Autonómia, felelősség

- Szakmai iránymutatás alapján, másokkal együttműködve javaslatokat fogalmaz meg adott feladat ellátására alkalmas szoftvertermékekre
- Szakmai útmutatással szoftvert telepít

4.1. A szoftver fogalma

A **szoftverek** elsősorban a belső programvezérlés megvalósítását teszik lehetővé. Elsődleges feladatuk, hogy a hardver által biztosított körülmények között a processzorral műveleteket végeztessenek. A szoftverek egy részével közvetlenül dolgozhatunk, más részük a felhasználó közvetlen beavatkozása nélkül (és sokszor tudta nélkül), a háttérben futva végez valamilyen feladatot.

Szűkebb értelemben a szoftver a számítógépen több, meghatározott cél érdekében együttműködő programot vagy programrendszert jelent. **Tágabb értelemben** a szoftver a hardver elemek működtetését végző programok mellett a gép használatához szükséges szellemi termékek összességét, a számítástechnikai termékek kevésbé kézzelfogható, könnyen reprodukálható, másolható csoportját (programokat, programokhoz kapcsolódó dokumentációt, valamint a különféle elektronikusan tárolt adattárakat és adatbázisokat stb.) jelenti.

A szoftver **meghatározza** az ember és a gép közötti kapcsolatot, azaz:

- a szolgáltatások körét (gondoljunk egy tervezőprogram és egy szövegszerkesztő program közötti különbségre),
- a kezelés kényelmességét (mennyire könnyű a program használatát megtanulni, a programot használni).

A továbbiakban a szoftver és (számítógépes) program fogalmakat egymás szinonimájaként használjuk.

A számítógép használatához a felhasználónak nincs feltétlenül szüksége programozói ismeretekre. Elegendő, ha ismeri a gép kezelésének alapvető lehetőségeit. Viszont azt fontos tudnia, hogy milyen feladatokra kívánja használni a számítógépet és ennek megfelelően célszerű a szoftvereket beszereznie. Amennyiben létezik már a piacon olyan szoftver, ami alkalmas mindazoknak a feladatoknak az ellátására, amelyek miatt a számítógépet használni szeretné, nem kell a programot megírnia/megíratnia, elegendő, ha beszerzi és telepíti azokat. A **szoftvercsomag** egy futtatásra kész, valamennyi szükséges összetevőt, dokumentációt is tartalmazó, megvásárolható program.

Ha beszereztünk egy szoftvert, akkor fel kell azt **telepítenünk** arra a számítógépre, amelyen használni akarjuk. A telepítést más néven installálásnak is nevezik. A telepítés folyamata általában nem csak egyszerűen a szoftver gépünkre másolását jelenti. Ahhoz, hogy programunk működni tudjon, ki kell alakítani a számára megfelelő környezetet: bizonyos, hardvertől, vagy felhasználói igénytől függő paramétereket be kell állítani, adatállományokat létre kell hozni, a tömörített állapotú programállományokat ki kell csomagolni stb. Az új szoftver telepítője hordozható adathordozón található, vagy az internet hálózatról tölthető le. A telepítés általában telepítőprogrammal indítható, vagy az adathordozó behelyezése után automatikusan elindul. A programot a sikeres telepítés után használhatjuk.

A szoftvertermékek egymást követő változatait **verziószámmal** azonosítják. Az egymást követő verziók rendszerint lefele kompatibilisek, azaz a későbbi (nagyobb verziószámú) rendszer mindent tud, amit a korábbi (kisebb verziószámú) rendszer tudott (de nem minden esetben!).

Időnként szükség lehet a programok **frissítésére**, amelyre ma már jellemzően az interneten keresztül kerül sor. A frissítés oka többféle lehet: újabb programverziók beszerzése, új funkciók beépítése, vagy hibajavítások (programhibák, biztonsági hibák javítása). A frissítés történhet

- manuálisan: a felhasználónak kell utánanéznie, hogy van-e a programhoz frissítés, és ha van, akkor azt neki kell beszereznie, letöltenie, majd végül telepítenie.
- automatikusan: ilyenkor az adott program egy része folyamatosan (a háttérben) fut, és időnként lekérdezi az interneten keresztül, hogy van-e hozzá valamilyen frissítés. Ha igen, akkor a telepítést automatikusan, vagy a felhasználó felügyelete mellett elvégzi.

Ha nincs szükségünk egy programra és a jövőben sem szeretnénk használni, célszerű azt **eltávolítani**, hogy ne foglalja a háttértáron a tárhelyet. Az eltávolítás sok esetben nem egyszerű törlést jelent, hanem egy összetett művelet, hiszen előfordulhat – különösen az összetett és sok szolgáltatást nyújtó szoftvercsomagok esetében – hogy több helyről is kell állományokat törölni, meg kell szüntetni a többi szoftverrel való kapcsolatot úgy, hogy közben azok funkcionalitása ne sérüljön. Azt is meg kell oldani, hogy ha az eltávolítás közben a felhasználó meggondolja magát, akkor visszavonhatók legyenek az addig elvégzett eltávolítási műveletek. A feladatot rendszerint az adott szoftverhez mellékelt eltávolító (uninstall) program végzi, de az operációs rendszer is tartalmazhat ilyen szolgáltatást.

A **szoftvertermékeket csoportosíthatjuk** aszerint, hogy milyen feladatot látnak el, és aszerint is, hogy milyen módon férhetők hozzá (azaz mit tehet vele a felhasználó és hogyan szerezheti be jogszerűen).

4.2. A szoftverek csoportosítása szolgáltatásaik alapján

A szoftver fogalma alatt sokféle programterméket értenek. Ezek egy része közvetlenül a hardver által nyújtott szolgáltatásokat használja, és azon kívül, hogy a gépünk működik, nemigen vesszük észre a jelenlétüket. Ezzel szemben más részükről nagyon jól tudjuk, hogy mire valók: ha megvásároljuk és elindítjuk azokat, a működésük látványos (például rajzolóprogram, vagy szövegszerkesztő). Az is előfordulhat, hogy nem találunk olyan programot a piacon, ami mindazt tudja, amire szükségünk lenne. Ebben az esetben akár mi saját magunk is elkészíthetjük a saját, személyre szabott, az igényeinknek megfelelő programot.

A **szoftvertermékeket feladatuk** szerint következőképpen csoportosíthatjuk:

- **Operációs rendszerek** (működtető rendszerek): a gép működtetéséhez nélkülözhetetlen programok (vagy inkább programok csoportjáról beszélhetünk). Alapvető hardvervezérlő feladatokat látnak el, lehetővé teszik a géppel való kommunikációt. Az operációs rendszer fő feladata biztosítani a gép hardver egységeinek kezelését, az adatállományok kezelését és egyéb programok futtatását. Legelterjedtebb változatai: Android, Linux, Mac OS, Windows.
- **Meghajtó programok** (vezérlőprogramok, driverek): adott perifériák kezelését végzik, ennek megfelelően beszélhetünk például billentyűzet driverről, egér driverről, nyomtató driverről stb. Ezeket a programokat vagy az operációs

rendszerrel együtt, vagy az adott perifériával kapjuk meg és ebben az esetben az eszköz használatának megkezdése előtt telepíteni kell.

- **Alkalmazói programok:** jól meghatározott, komplex felhasználói igényt kielégítő programrendszerek. Az alkalmazói program indítása az operációs rendszerből történik, tehát a program az operációs rendszer felügyelete alatt fut. Indításkor a program betöltődik a gép operatív memóriájába, és mindaddig ott marad, amíg a programból ki nem lépünk. Kilépéskor az alkalmazói program törlődik az operatív memóriából. Ide tartoznak például az irodai programcsomagok, fejlesztő/programozó környezetek, a játékprogramok stb.
- **Segédprogramok:** egy-egy kisebb, de gyakran előforduló feladat elvégzésére használhatók, mint amilyen például a számolás, jegyzetelés, adatállományok tömörítése, biztonsági másolatok készítése, számítógépes kártevők kiszűrése. Az operációs rendszerhez szinte minden esetben mellékelnek ilyen programokat.
- Azokat a számítógépes programokat, amelyeket azért készítenek, hogy a számítógépek működését valamilyen módon megzavarják, vagy lehetetlenné tegyék, számítógépes **kártevőknek** nevezzük. A kártevőket sokféleképpen lehet csoportosítani (lehetnek például trójai programok, férgek, vírusok stb.); a lényeg, hogy szinte mindegyikük az operációs rendszer, vagy más futtatható program programkódjába beépülve, hordozható háttértárak, vagy az internet – és sok esetben az emberi hiszékenységgel, tudatlansággal és gondatlansággal – segítségével szaporodik, terjed. Az, hogy e kártékony programok hogyan változtatják meg a környezetük viselkedését és milyen jellegű kárt okoznak, csak a kártevőíró programozó akaratától és ügyességétől függ.

4.3. A szoftverek csoportosítása hozzáférhetőség szerint

A **szoftvertermékek csoportosítása hozzáférhetőség** (jogi, kereskedelmi szempontok) szerint:

- Egyedi alkalmazások: Speciális, egyedi igényeket elégítenek ki. Ilyen szoftverek lehetnek a teljesen testre szabott ügyviteli rendszerek, vállalatirányítási információs rendszerek. Ezeket a termékeket egyedi igények szerint, megrendelésre készítik (illetve előre gyártott programmodulokat igény szerint módosítanak). Az egyedi szoftvertermékek általában nem kerülnek kereskedelmi forgalomba.
- Általános célú alkalmazások: Ezeket egy-egy adott feladat megoldására fejlesztették ki. Olyan felhasználói igényeket elégítenek ki, amelyek több helyen, általánosan felmerülnek. Ide tartoznak a szövegszerkesztők, táblázatkezelők, adatbázis-kezelők, rajzoló- és képszerkesztő programok, tervezőprogramok, játékok stb. Az általános célú felhasználói szoftverek kereskedelmi forgalomban kaphatók: „dobozos” (retail) változatban, teljes értékű szoftverként, illetve OEM (Original Equipment Manufacturer) változatban, többnyire hardverhez mellékelve (vagy hardver nélkül, de ebben az esetben telepítés után kötődik a szoftver a hardverhez). A megvásárolt szoftvereket általában lehet frissíteni: az update általában hibajavításokat tartalmaz és többnyire ingyenes, míg az upgrade új szolgáltatásokat épít be a már meglévő programba és fizetni kell érte.

- Demoware, trialware: Ingyenesen beszerezhető és szabadon terjeszthető, a teljes értékű változathoz képest funkcióiban jelentősen korlátozott, a megvásárolható változat szolgáltatásainak bemutatására szolgáló program.
- Shareware: Ingyenesen beszerezhető és szabadon terjeszthető, vagy teljesen működőképes program, vagy egyes funkciókban korlátozott program. A programot változatlan formában, ingyen kell továbbadni, módosítása tiltott. Meghatározott ideig (néhány hét, illetve hónap) használható, ez alatt a felhasználó eldöntheti, hogy megfelel-e számára. A határidő letelte után a programot meg kell vásárolni, vagy le kell törölni a számítógépről.
- Adware: Ingyenesen beszerezhető és szabadon terjeszthető program. A programot változatlan formában, ingyen kell továbbadni, módosítása tiltott. Az ingyenességet úgy tartják fenn, hogy az adware termék meghatározott időközönként az interneten keresztül reklámanyagokat tölt le és jelenít meg a felhasználó számítógépén.
- Freeware: Ingyenesen beszerezhető és szabadon terjeszthető, teljesen működőképes program. A programot változatlan formában, ingyen kell továbbadni, módosítása általában tiltott.
- Szabad felhasználású szoftver: Ingyenesen beszerezhető, szabadon módosítható és terjeszthető program.

4.4. A szoftver és a szerzői jog

A szoftvereket digitálisan rögzítik az adathordozóra. Ez az adathordozó minőségromlás nélkül másolható. Éppen ezért a **szoftverek jogi védelméről** kezdetektől fogva szigorúan próbáltak gondoskodni, hiszen egy-egy szoftvertermék kifejlesztése rengeteg munkaórát és nagy ráfordítást igényelhet, és ennek megfelelően több százezer, vagy akár több millió forintba is kerülhet.

A szerzői joggal kapcsolatos rendelkezéseket a többször módosított **1999. évi LXXVI. törvény** tartalmazza. A törvény összhangban áll számítógépes programok jogi védelméről szóló (korábban 91/250/EGK, illetve a jelenleg hatályos) **2009/24/EK irányelvvel**.

Nem árt, ha tudjuk: a szoftverek illegális felhasználását a jogszabályok szigorúan büntetni rendelik. A jogszabályok mindenkire – magánszemélyekre és gazdasági társaságokra – egyaránt vonatkoznak, tekintett nélkül a szoftver értékére és a felhasználási szándéokra. A jogszabályok ismeretének hiánya nem mentesít a büntetés alól!

A **2012. évi C. törvény a Büntető Törvénykönyvről** a következőket tartalmazza (**385.§ – 386.§**):

Szerzői vagy szerzői joghoz kapcsolódó jogok megsértése

385. § (1) Aki másnak vagy másoknak a szerzői jogról szóló törvény alapján fennálló szerzői vagy ahhoz kapcsolódó jogát vagy jogait vagyoni hátrányt okozva megsérti, vétség miatt két évig terjedő szabadságvesztéssel büntetendő.

(2) Az (1) bekezdés szerint büntetendő, aki a szerzői jogról szóló törvény szerint a magáncélú másolásra tekintettel a szerzőt, illetve a kapcsolódó jogi jogosultat megillető üreshordozó díj, illetve reprográfiai díj megfizetését elmulasztja.

(3) A büntetés büntett miatt három évig terjedő szabadságvesztés, ha a szerzői vagy szerzői joghoz kapcsolódó jogok megsértését nagyobb vagyoni hátrányt okozva követik el.

(4) Ha a szerzői vagy szerzői joghoz kapcsolódó jogok megsértését

a) jelentős vagyoni hátrányt okozva követik el, a büntetés büntett miatt egy évtől öt évig,

b) különösen nagy vagyoni hátrányt okozva követik el, a büntetés két évtől nyolc évig,

c) különösen jelentős vagyoni hátrányt okozva követik el, a büntetés öt évtől tíz évig terjedő szabadságvesztés.

(5) Nem valósítja meg az (1) bekezdés szerinti bűncselekményt, aki másnak vagy másoknak a szerzői jogról szóló törvény alapján fennálló szerzői vagy ahhoz kapcsolódó jogát vagy jogait többszörözéssel vagy lehívásra történő hozzáférhetővé tétellel sérti meg, feltéve, hogy a cselekmény jövedelemszerzés célját közvetve sem szolgálja.

Védelmet biztosító műszaki intézkedés kijátszása

386. § (1) Aki a szerzői jogról szóló törvényben meghatározott hatásos műszaki intézkedést hasznonszerzés végett megkerüli, vétség miatt két évig terjedő szabadságvesztéssel büntetendő.

(2) Az (1) bekezdés szerint büntetendő, aki a szerzői jogról szóló törvényben meghatározott hatásos műszaki intézkedés megkerülése céljából

a) az ehhez szükséges eszközt, terméket, számítástechnikai programot, berendezést vagy felszerelést készít, előállít, átad, hozzáférhetővé tesz, vagy forgalomba hoz,

b) az ehhez szükséges vagy ezt könnyítő gazdasági, műszaki vagy szervezési ismeretet másnak a rendelkezésére bocsátja.

(3) A büntetés büntett miatt három évig terjedő szabadságvesztés, ha a műszaki intézkedés kijátszását üzletszerűen követik el.

(4) Nem büntethető a szerzői jogról szóló törvényben meghatározott hatásos műszaki intézkedés megkerüléséhez szükséges eszköz, termék, berendezés, felszerelés készítése vagy előállítása miatt az, aki mielőtt tevékenysége a hatóság tudomására jutott volna, azt a hatóság előtt felfedi, és az elkészített, illetve az előállított dolgot a hatóságnak átadja, és lehetővé teszi a készítésben vagy az előállításban részt vevő más személy kilétének megállapítását.

Vásárláskor érdemes áttanulmányozni a **szoftverhez tartozó felhasználási (más néven licenz) szerződést**. A licencszerződés határozza meg a szerzői jog tulajdonosa (többnyire a szoftver készítője) által megengedett szoftverhasználat feltételeit, a felhasználó jogait és kötelezettségeit és emellett, a gyártó/forgalmazó garancia- és kötelezettségvállalását (vagy éppen annak elhárítását), a frissítés lehetőségeit és feltételeit. Kiderül belőle az is, hogy például hány gépre telepíthető az adott szoftver és szabad-e kereskedelmi, üzleti céllal használni, vagy csak nonprofit és oktatási célra alkalmazható-e.

Ha valaki szoftvert tárol, vagy használ, azt úgy kell tekinteni, mintha elolvasta és elfogadta volna a szerződést! A szerződés szoftver dokumentációjával együtt mellékelik, a program telepítésekor vagy legelső indításakor rendszerint megjelenik a képernyőn és ilyenkor a program telepítéséhez, illetve használatához el kell azt fogadni.

Számos előnye van annak, ha tiszta forrásból szerezzük be a szoftvereket és azokat a jogszabályokkal és a felhasználói szerződéssel összhangban használjuk, például:

- Nagyobb biztonságban lesznek az adataink, kisebb az esélye annak, hogy a szoftver kártevőt tartalmaz. Egy illegális forrásból szerzett programot könnyen lehet, hogy módosítottak és interneten keresztüli adatlopásra, vagy egyéb kártételre is felkészítettek.
- Minden – a jogszabályok és a felhasználási szerződés – által biztosított jogunkkal élhetünk. Megkapjuk az eredeti és teljes dokumentációt, igénybe vehetünk szükség esetén terméktámogatást, ingyen, vagy kedvezményes áron juthatunk hozzá a frissítésekhez, javításokhoz.
- Nem kell rejtőzködnünk a hatóságok előtt.

4.5. Ellenőrző kérdések

1. Mit értünk szűkebb és tágabb értelemben a szoftver fogalma alatt? [43]
2. Mit határoz meg a szoftver? [43]
3. Mit nevezünk szoftvercsomagnak? [43]
4. Mi a telepítés? [43]
5. Mit jelöl a verziószám? [43]
6. Mire szolgál a frissítés és hogyan történhet meg? [44]
7. Mit értünk a program eltávolításán? [44]
8. Hogyan csoportosíthatók a szoftverek a szolgáltatásaik, ellátott feladataik alapján és mi jellemző az egyes csoportokra? [44]
9. Hogyan csoportosíthatók a szoftverek hozzáférhetőség szerint és mi jellemző az egyes csoportokra? [45]
10. Miért kell gondoskodni a szoftverek jogi védelméről? [46]
11. Milyen fontosabb jogszabályok, előírások rendelkeznek a szoftverekhez kapcsolódó jogokkal? [46]
12. Mi a licenz szerződés és mit tartalmaz? [47]
13. Milyen előnyei vannak annak, ha tiszta forrásból szerezzük be a szoftvert? [47]

5. SZÁMÍTÓGÉP-HÁLÓZATI ALAPISMERETEK

Elvárt tanulási eredmények

Tudás

- Ismeri a számítógép-hálózat, a protokoll, a hand-shaking, a hoszt, a szerver és a kliens fogalmát
- Érti a munkacsoport és a tartomány-elvű hálózat lényegét
- Felsorolja a szerver szolgáltatásait és csoportjait, valamint a kliens feladatait
- Felsorolja a számítógép-hálózatok kialakításának céljait
- Átfogóan ismeri a hálózatok csoportjait hozzáférhetőség és területi elhelyezkedés szerint
- Ismeri a felhasználói jogosultságokat, a tipikus felhasználói csoportokat és azok fő jellemzőit
- Felismeri a hálózati topológiákat és ismeri azok alapvető jellemzőit
- Ismeri a hálózati kommunikáció módjait
- Érti a kommunikációs versenyhelyzet kialakulásának kezelési módjait
- Felsorolja az összekapcsoló közegeket és azok fő jellemzőit
- Érti az internetwork, az internetworking device és a szegmens jelentését
- Felsorolja a fontosabb hálózati hardvereket (hálózati kártya, modem, switch, repeater, bridge, router, routem, gateway) és átfogóan ismeri azok feladatait
- Átfogóan ismeri az internet kialakulásának történetét
- Érti az IP-cím és a hosztnév lényegét
- Felsorolja az internet fő szolgáltatásait és azok alapvető jellemzőit

Képesség

- Meghatározza, azonosítja a hálózati hardvert annak szolgáltatásai alapján

Attitűd

- Értékként tekint a számítógép-hálózatokra
- Elfogadja és figyelembe veszi a számítógép-hálózaton a felhasználói jogokat, illetve a jogok korlátozását
- Kész akár önálló, akár csoportos munkát végezni számítógép-hálózaton

Autonómia, felelősség

- Szakmai iránymutatás alapján, másokkal együttműködve javaslatokat fogalmaz meg topológiára, hálózati hardverre és szoftverre

5.1. Számítógép-hálózati alapfogalmak

A számítógépek kezdetben egymástól teljesen elkülönülve, önállóan dolgoztak. Később, az informatika fejlődésével megjelent az igény arra, hogy a számítógépek egymással kommunikáljanak.

A **számítógép-hálózat** egymással kapcsolatban álló önálló számítógépek rendszere. Ahogy a Microsoft fogalmaz: „Kommunikációs eszközökkel összekötött számítógépek és hozzájuk tartozó készülékek csoportja.” (Microsoft Press, 1999, 221. oldal). A hálózat tartalmazhat állandó összeköttetéseket és ideiglenes kapcsolatokat is.

Az elküldendő adatok összeállításának és a fogadott adatok értelmezésének – pontos, sok esetben szabványban rögzített – szabályait adatátviteli **protokollnak** nevezzük. A protokoll a számítógépek összekapcsolását a lehető legkevesebb hibával biztosító szabványrendszer. A protokoll határozza meg a kommunikáló eszközök adatformátumát, sorrendjét, valamint a kommunikáció során megtett lépéseket. Ha az adó és a vevő meg akarja érteni egymást, azonos protokollt kell használniuk. A hálózati protokoll feladata, hogy a számítógépek között a fizikai eszközök (például hálózati kártya, modem stb.) segítségével az adatokat elküldje, illetve az adatok átvitelét ellenőrizze. A hálózat két gépe közötti kapcsolat a protokollok felismerésével, majd az adási és vételi készség visszaigazolásával kezdődik. Ezt a folyamatot nevezzük **hand-shakingnek**. Abban az esetben, ha két számítógép nem ugyanazt a protokollt alkalmazza, nem képesek közvetlenül adatot cserélni. Ilyen esetekben egy közvetítő számítógép is szükséges a két gép közé, ami az egyik protokollról a másikra fordít.

Munkacsoport (workgroup), peer-to-peer (p2p) elrendezés esetén a hálózaton levő számítógépek egyenrangúak, nincs közös felhasználói és számítógép-adminisztráció. A **tartomány-elvű** (domain) hálózatban mindig van egy elsődleges ún. tartományvezérlő szerver, amely a felhasználók beléptetését, a tartományhoz tartozó számítógépek adminisztrációját végzi, lehetővé téve a felhasználók jogainak pontos meghatározását és a távoli gépfelügyeletet.

Azokat a számítógépeket, amelyeket egy hálózat segítségével összekötünk, **hosztoknak** nevezzük.

A hálózatban lévő gépek lehetnek kiszolgálók – idegen szóval szerverek –, továbbá ügyfélgépek, munkaállomások – azaz kliensek. A **szervergépek** általában nagy teljesítményű, nagy tárolókapacitású és folyamatos üzemű számítógépek, amelyek a hálózatba kapcsolt többi gép számára nyújtanak szolgáltatásokat. A nyújtott szolgáltatások különfélék lehetnek, sőt gyakran előfordul, hogy nem egy szerveren koncentrálódnak, hanem a hálózatban több szerver található, egy vagy több saját funkcióval. Ha egy kiszolgáló egyetlen szolgáltatást nyújt csak, akkor beszélünk **dedikált szerverről**. A **kliens** gépek, munkaállomások, valamilyen hálózati szolgáltatást vesznek igénybe.

A **szerverek csoportjai** az általuk nyújtott szolgáltatások alapján:

- Az adatbázis-szerverek, fájlserverek nagy tárolókapacitású számítógépek. Feladatuk a hálózaton közösen használt állományok, adatbázisok, alkalmazások tárolása és biztosítása.
- A nyomtatószerverek végzik a hálózaton keresztüli nyomtatással kapcsolatos feladatokat, a hozzájuk kapcsolt nagy teljesítményű nyomtatók vezérlését.

- A mail szerverek a felhasználók levelezését kezelő, az elektronikus leveleket fogadó és küldő kiszolgáló gépek.
- A web-szerverek a belső hálózat számára és a külvilág felé szolgáltatják az internetes dokumentumokat, vagy végzik például az internetről érkező megrendelések feldolgozását.
- Proxy szervereket használunk abban az esetben, ha hálózatunk nem közvetlenül kapcsolódik az internethez. Az internettel közvetlen kapcsolatban levő proxy szerveren keresztül a számítógép számára oly módon válik az internet elérhetővé, hogy a kért állományokat a proxy szerver tölti le, majd továbbítja a felhasználónak. Ha egy dokumentumra gyakran mutatkozik igény, a felhasználók azt nem az interneten lévő távoli gépről, hanem a proxy szervertől kapják, ezáltal az csökken a forgalom az internet felől. Egyes proxy szerverek képesek arra, hogy a felhasználói igényeknek elébe menjenek: a gyakran látogatott weboldalak bizonyos időközönként előre letöltik.

5.2. A hálózatok céljai

Számítógép-hálózatok kialakításának **többféle célja** lehet:

- Erőforrás megosztás: Az különböző eszközök, programok és adatok a felhasználók számára elérhető legyenek függetlenül a tárolás helyétől.
- Központi programtárolás és kiszolgálás: A programokat nem feltétlenül kell a felhasználó saját gépén tárolni. Helyi hálózatokon ennek a megoldásnak az az előnye, hogy mindig az aktuális verzió fog futni a felhasználó gépén anélkül, hogy ezzel külön kellene foglalkoznunk.
- Kommunikáció, hozzáférés távoli információkhoz: A hálózati rendszer kommunikációs közegként is használható, például elektronikus levelezéshez, hálózati telefonáláshoz (csevegéshez) és videó átvitelhez. Emellett elérhetővé válnak távoli adatbázisok is.
- Skálázhatóság: Ez annak biztosítását jelenti, hogy a rendszer teljesítményét a terhelés növekedésével fokozatosan növelni lehessen újabb számítógépek hozzáadásával (vagy éppen fordítva: terheléscsökkenés esetén csökkenteni a hálózatba bevont számítógépeket).
- Takarékoság: A személyi számítógépek sokkal jobb ár/teljesítmény aránnyal rendelkeznek, mint a nagygépek. Egy nagygép (mainframe) ugyan nagyságrendekkel gyorsabb lehet, viszont ugyanakkor akár ezerszer drágább is, mint egy nagyteljesítményű személyi számítógép.
- Nagyobb megbízhatóság: Több alternatív eszközt vonhatunk be a munkába ott, ahol egy is elegendő lenne, így – ha az egyik meghibásodik –, még mindig ott van a többi.

5.3. A hálózatok csoportosítása

Hozzáférhetőség szerint beszélünk:

- nyilvános hálózatokról, amelyekhez – megfelelő eszköz birtokában – bárki hozzáférhet (ilyen hálózat az internet) és

- magánhálózatokról, amelyhez a hozzáférés csak meghatározott személyek számára engedélyezett (ilyen lehet egy cég belső hálózata).

Területi elhelyezkedés, kiterjedés szerint a számítógép-hálózat a következő csoportokba soroljuk: helyi, városi és távolsági hálózat.

Helyi hálózat: A helyi hálózatok (LAN: Local Area Network) állhatnak néhány géptől kezdve akár több száz gépből is, tehát nem a gépek száma számít, hanem az, hogy ezek a gépek egymástól fizikailag csak néhány száz méter (esetleg néhány kilométer) távolságban helyezkednek el. A gépek valamilyen közvetítő eszköz révén kapcsolatban állnak egymással és az adatforgalmat összehangolja valamilyen programrendszer.

Hatáskörzetük nagysága alapján a helyi hálózatok a következő kategóriába sorolhatók: szobahálózatok, teremhálózatok, épülethálózatok, telephelyhálózatok, nagyvárosi hálózatok. A helyi hálózatokat más jellemzők mentén is csoportosíthatjuk; például aszerint, hogy az adott típusú hálózat milyen alkatrészekből épül fel, milyen átviteli protokollt alkalmaz, milyen operációs rendszerekkel képes együttműködni; vagy aszerint, hogy mi a szerepe a hálózatba kapcsolt számítógépeknek: egyenrangúak, vagy egyes gépek célfeladatokat látnak el (kiszolgáló), a többi gép e szolgáltatásokat veszi igénybe (ügyfél).

A helyi hálózatok rendeltetése:

- A felhasználók csoportban tudjanak dolgozni; ilyenkor fontos a gyors információcsere, az információ kölcsönös rendelkezésre bocsátása.
- Ha rengeteg adatot (dokumentumot vagy egyéb állományokat) tárolnak, és ezen valaki valamit változtat, akkor a módosított állományokhoz mindenki a lehető legrövidebb időn belül hozzáférhessen. (Ha mindenki egyénileg végezné el a módosításokat, majd utána hordozható adathordozón vagy elektronikus levélben köröztetné, akkor az adatok, állományok változása, módosítása követhetetlené válna.)
- A közös tárolás az adattárolók jobb, hatékonyabb kihasználását jelenti. Szinte bármilyen periféria megosztható a helyi hálózaton, így megoldható az erőforrások hatékony elosztása.
- Az adatok mellett a programokat is közösen vehetik igénybe a helyi hálózat használói, így azokat sem kell minden egyes gépen tárolni. A csak egy helyen tárolt programok karbantartása, frissítése is egyszerűbb.
- A helyi hálózatba kapcsolt gépek egy egységben kapcsolhatók a távolsági hálózatra. Emiatt nem kell minden egyes gépet felszerelni a távolsági hálózatba csatlakozás eszközeivel, a távolsági hálózathoz mégis mindenki hozzáférhet.

Városi hálózat: Ez a hálózat egy nagyváros integrált rendszere. A MAN (Metropolitan Area Network) helyi hálózatok (és nagyszámítógépek) összekapcsolásával jön létre. Ez az összekapcsolás valamilyen nagysebességű technológia felhasználásával történik, a kisebb hálózatok, a személyi számítógépek és egyéb eszközök pedig bérelt vonalakon tartják a kapcsolatot az alaphálózattal.

Távolsági hálózat: A nagyobb területi egységre, országra, az egész világra kiterjedő nagysebességű hálózat neve. Manapság már egyre inkább a világméretű hálózatot (GAN: Global Area Network) jelenti, hiszen az elkülönült távolsági hálózatokat folyamatosan

kapcsolják hozzá a meglévő világhálózathoz. A legismertebb formája az internet, de más, ettől elkülönült banki, üzleti, kormányzati távolsági hálózatok is léteznek.

5.4. A hálózat felhasználói

Az adatokhoz való hozzáférést előre meghatározott **hozzáférési szintek** szabályozzák. Ez azt jelenti, hogy ahhoz, hogy egy felhasználó a hálózat erőforrásait használja, egy nyilvános azonosítóval (username, user account, login) és egy titkos jelszóval (password) kell azonosítania magát.

A hálózati erőforrások, jogosultságok kiosztásának feladata a rendszergazdára hárul. A rendszergazda olyan szakember, aki az adott rendszerről mélyebb ismeretekkel rendelkezik, felelős a rendszer működéséért és biztonságáért. Ő egy megkülönböztetett jogokkal rendelkező felhasználó, számára minden elérhető a rendszerben anélkül, hogy a többi felhasználó jogait sértené.

A **felhasználói jogosultságok** általában a következők lehetnek:

- Olvasás: Ez a hálózaton elérhető állományok megnyitását és olvasását jelenti csupán.
- Létrehozás/módosítás/törlés: Új állományok létrehozását, módosítását és törlését is lehetővé teszi a felhasználó számára. Egyes hálózatokon ezek a jogosultságok elválhatnak egymástól.
- Futtatás: Programok futtatásának engedélye.
- Jogok megváltoztatása/átadása: Az állományoknak van tulajdonosa, aki általában megegyezik az állományt létrehozó felhasználóval. Ő – hacsak a rendszergazda eltérően nem rendelkezik – valamennyi jogosultsággal rendelkezik: az előbb felsorolt jogokat megváltoztathatja az adott állományra (és egyes esetekben a tárolás helyére) vonatkozóan, más felhasználóknak vagy felhasználói csoportoknak megadhat jogokat és akár a jogok megváltoztatásának jogát is átengedheti.

Egy számítógépes hálózatban a felhasználók száma igen magas lehet, ezért célszerű őket valamilyen rendszer szerint csoportosítani. A hálózatokban alkalmazott **tipikus felhasználók és csoportok**:

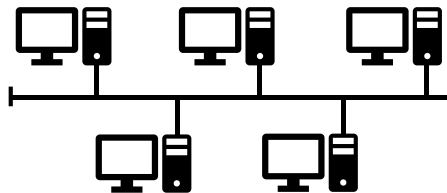
- Rendszergazda (supervisor, administrator): maximális jogosultsággal rendelkező felhasználó, aki felügyeli a teljes rendszert. (Ez nem jelenti azt, hogy bármilyen felhasználói adatot és állományt korlátozás nélkül megtekinthet, vagy megváltoztathat!)
- Operátor (operator): egy-egy részterületen rendelkezik maximális jogosultsággal, így beszélhetünk például szerver operátorról (a helyi szerver erőforrásainak kezeléséhez való joggal), fiók operátorról (a felhasználók és felhasználói csoportok kezelésének jogával), biztonsági mentés operátor (állományok és könyvtárak mentéséhez való joggal),
- Felhasználó (user): Az „átlagos” felhasználó, aki igénybe veszi a hálózati szolgáltatásokat, a hálózaton dolgozik és ennek érdekében a hozzáfér a – a feladatai ellátásához – szükséges helyi hálózaton elérhető erőforrásokhoz.

- Vendég (guest): Ha ez a felhasználói csoport engedélyezve van a hálózaton, akkor is csak nagyon korlátozottak a jogai, általában csak az adatok, állományok egy szűk körére rendelkezik olvasási jogosultsággal.

5.5. Hálózati topológia

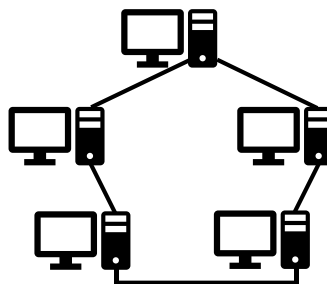
Alapvető kérdés a szakemberek számára, hogy egymáshoz viszonyítva hogyan helyezkednek el a gépek a hálózatban. Ezt nevezik a **hálózat topológiájának**. A legismertebb elrendezések:

- Sín (vagy busz) topológia: Sín elrendezésnél a hálózatba kapcsolt gépek egyetlen vezetéken osztoznak (gerincvezeték), amelyhez az összes csomópont (önálló kommunikációra képes, saját hálózati címmel rendelkező eszköz) csatlakozik (8. ábra). A gerinc mindkét vége ellenállással van lezárva. Az egyedi címekkel rendelkező csomópontok között semmiféle speciális sorrend nem adható meg, sorosan kapcsolódnak. A megoldás olcsó és kevés kábel kell hozzá. Az elrendezés hátránya, hogy vonalszakadás esetén az egész hálózat használhatatlanná válik.



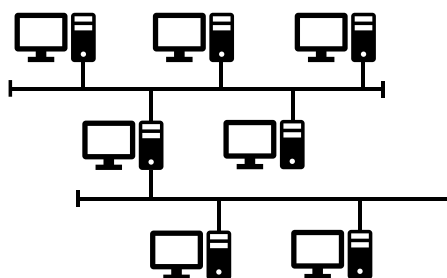
8. ábra. Sín topológia.

- Gyűrű topológia: Gyűrű elrendezés esetén a hálózatba kapcsolt gépek egymást követő zárt láncba szerveződnek, így minden kapcsolódási ponthoz rendelhető egy előző és egy következő kapcsolódási elem (9. ábra). Előnye, hogy egyszeres vonalszakadás esetén a hálózat nem válik használhatatlanná és nincs leterhelt központi csomópont. Kétszeres gyűrű alkalmazásával növelhető a hálózat biztonsága. Az adatáramlásnak meghatározott iránya van; amíg az adatot nem mentik le, az a gyűrűben kering, így nagy a kockázat, hogy az adatok megsérülnek, elvesznek.



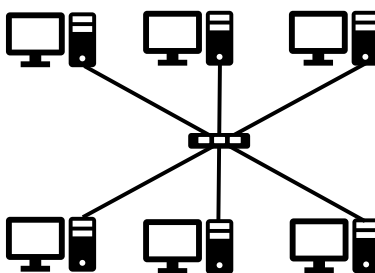
9. ábra. Gyűrű topológia.

- Fa topológia: A sín topológia fa topológiává egészíthető ki, ha a különálló buszágakat egy bizonyos ponton fastruktúrát alkotva összekapcsolják (10. ábra). Meghibásodás esetén csak a csomópont és a hozzá tartozó ágak esnek ki.



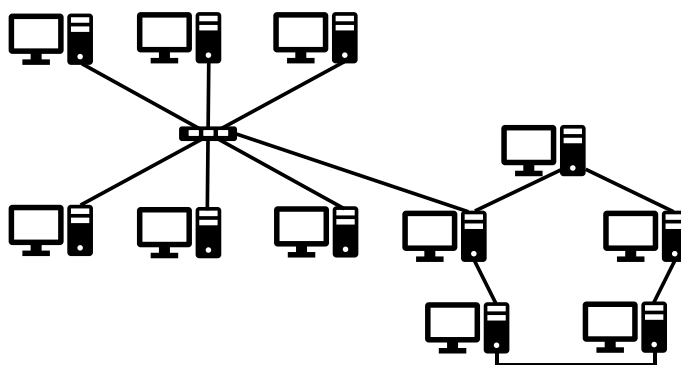
10. ábra. Fa topológia.

- Csillag topológia: Csillag elrendezésnél a hálózatba kapcsolt gépek ugyanarra a csomópontra (elosztó) csatlakoznak (11. ábra). Az elosztók gyűjtik össze egy-egy gépcsoport jeleit és továbbítják. Az elrendezés előnye, hogy újabb elosztók segítségével újabb gépcsoportokat lehet a rendszerhez csatlakoztatni, továbbá vonalszakadás esetén csak egy gép (vagy gépcsoport) válik használhatatlanná, és nem az egész hálózat; a többi gép továbbra is tud kommunikálni egymással. Ha vezetékkel valószínűleg meg, sok kábel kell hozzá.



11. ábra. Csillag topológia.

- Vegyes topológia: Az előző formák vegyes alkalmazása (12. ábra).



12. ábra. Vegyes topológia.

5.6. A hálózati kommunikáció módjai

A hálózati kommunikációnak többféle módja lehetséges, ezek a következők:

- Pont-pont összeköttetés esetén a két kommunikációs végpontot egy kábellel kötik össze. Az üzenetek – illetve szétdarabolt részei, a csomagok – a kábelen keresztül áramlanak. Ha a vevő megkapja a csomagot és az nem neki szól, azt továbbadja a

következő pont-pont összeköttetésen keresztül. Ennek a fajta összeköttetésnek az előnye, hogy a kommunikációs hiba elsősorban két pont között léphet fel, így a hibák behatárolása viszonylag egyszerű. A kommunikációban résztvevők csak egymást látják, így nem szükséges a másik fél azonosítása. A pont-pont összeköttetés hátránya, hogy több pontot tartalmazó hálózatban a pontok közötti kommunikáció csak közvetlen összeköttetések kialakításával lehetséges.

- Üzenetszórásos kommunikáció esetén a hálózatra felfűzött számítógépek minden adatot észlelnek, de csak a nekik szólókra válaszolnak. Üzenetszórásos csatornával rendelkező hálózatoknál egyetlen kommunikációs csatorna van, és ezen osztozik az összes, a hálózatba kapcsolt számítógép. Az adó által küldött üzeneteket a hálózat minden számítógépe veszi. Az, hogy az üzenet kinek szól, az üzenetben elhelyezett egyedi, gépet címező címinformáció tartalmazza. Vagyis ebben az esetben a felek egyértelmű azonosítására alkalmas címezés szükséges. A csatornán kapott üzeneteket a gépek először csak olyan mértékben dolgozzák fel, hogy eldönthessék, hogy az üzenet nekik szól-e. Az üzenet feldolgozását ezután már csak az a számítógép folytatja, amelyiknek a címe megegyezett az üzenetben lévő címmel (unicasting). Az egyedi gépcímezés mellett lehetőség van arra, hogy csoportcímezés alkalmazásával gépek egy csoportjának (multicasting), vagy a hálózatba kapcsolt összes gépnek küldjünk üzenetet (broadcasting).

Közös csatorna használata esetén előfordulhat, hogy egynél több állomás is akar adni. Ebben az esetben **versenyhelyzet** alakul ki az adók között, amit valamilyen módon fel kell oldani:

- A vezérlőjelet (token) használó hálózatokban egy állomás csak akkor küldhet adatot a hálózaton, ha a vezérlőjelet megkapta. Amint befejezte az adást, a vezérlőjelet – vagyis a használati jogot – továbbítja a következő állomásnak. A vezérlőjeles gyűrű (token ring) hálózatban minden állomás csak a két szomszédjával áll fizikai kapcsolatban. Az állomások az így kialakított gyűrűben egymásnak adják a vezérlőjelet. Az adni kívánó állomás a neki átadott vezérlőjelet kivonja a hálózathoz és elküldi az üzenetet, ami a gyűrűben haladva eléri a címzett állomást. A címzett beolvassa az üzenetet, beállítja a „rendben vett” jelzést és tovább küldi a hálózaton. Az üzenet így visszaér a feladóhoz, amelyik ellenőrzi a vételi jelzést, majd a vezérlőjelet továbbítja a szomszédjának.
- A másik lehetőség (például üzenetszórásos hálózatoknál), hogy amikor egy eszköz adatot akar küldeni, figyelni kezdi a hálózaton, hogy folyik-e éppen forgalom. Ha igen, akkor várakozik, ellenkező esetben elkezd az adást. Ez egybeeshet egy másik állomás adáskezdésével, és ekkor az üzenetek ütköznek, a célállomások nem tudják venni azokat. Az ütközést mindkét (illetve az összes) adóállomás érzékeli, és leállítja a küldést. Ezután véletlenszerű ideig várnak, majd újra figyelni kezdik a csatornát. Ha ismét ütközést észlelnek, már nagyobb tartományból választanak véletlenszerűen várakozási időt, ezzel lehetővé téve a hálózati torlódások gyors levonulását.

5.7. Összekapcsoló közegek

A számítógépeket összekapcsoló közeg igen sokféle lehet. Egy hálózaton belül rendszerint többféle közeget is alkalmazhatnak. A **kapcsolatok típusai összekapcsoló közeg szerint**:

1. Vezetékes kapcsolatok

- Koaxiális kábel: 30-300 Ohm-os vezetékek a kábel belsejében tömör rézhuzallal, amit szigetelőréteg vesz körül. Létezik vékony és vastag változata; a vastag nagyobb távolságra vihető. Egy kábelszegmensre legfeljebb 100 gép csatlakozhat, a szegmens hossza maximum 185 méter, illetve vastag kábelrel 500 méteres szegmens is kialakítható, de egy hálózat több szegmensből is állhat. Átviteli sávszélessége 100-150 Mbit/másodperc.
- Csavart érpár: Általában négy árnyékolt (STP: Shielded Twisted Pair), vagy árnyékoltatlan (UTP: Unshielded Twisted Pair) csavart rézvezeték-pár egy kábel-harisnyában. A megengedett szegmenshossz 100 méter, ami lehet hosszabb vagy rövidebb a kábel minőségétől függően. Sávszélessége 4-1000 Mbit/másodperc.
- Optikai kábel: A legkorszerűbb vezetékes átviteli módszer. Egy fényárnyékoló réteggel körülvett, vékony, hengeres üvegszál. A kábelek – a fény regenerálása nélkül akár 100 km távolság áthidalását is lehetővé teszik. Sávszélessége néhány tíz Gbit/másodperc körüli.

2. Vezeték nélküli kapcsolatok

- Rádióhullám: Nagyobb távolságok áthidalására gyakran használnak mikrohullámú átvitelt. Feltétel, hogy az adó-, és vevőtornyok „lássák” egymást. Az áthidalt távolság akár 100 km is lehet. Probléma lehet a villámlás és egyéb légköri zavarok. Többféle változata létezik, amelyek az áthidalt távolságban és a maximális sebességben (sávszélességben) is jelentősen különböznek egymástól. Ide tartozik a bluetooth (néhány méter a hatótáv és néhány Mbit/másodperc) és a wifi (néhány tíz méter, néhány száz Mbittől néhány Gbit/másodperc).
- Műhold: A rádióhullámmal történő átvitel egy változata. Az egyenlítő fölött keringő műholdakat használnak adatátvitelre. A műholdakra felküldött mikrohullámú jeleket a műhold felerősítve visszasugározza. Az adatátviteli sebesség változó, akár több száz Mbit/másodperc is lehet.
- Lézer- és infravörös sugár: A lézersugárral nagyobb távolságot, míg infravörös sugárral kisebbet szoktak áthidalni. Itt is feltétel, hogy az adó és a vevő lássa egymást. Légköri zavarok, mint az eső, köd, légköri szennyeződések zavarhatják az adatforgalmat. Az infravörös átvitel csak nagyon kis távolságra (néhány méter) és kis sebességre alkalmas átviteli forma. A lézeres átvitel a mindennapokban nem terjedt el.

5.8. Hálózati hardver

Hálózatok többféleképpen kapcsolhatók össze. Néhány évtizeddel ezelőtt erre a célra még egyedi eszközöket (szoftvert és hardvert) fejlesztettek ki. Napjainkban szabványosított eszközök széles választéka áll rendelkezésünkre. A gyártók arra törekcsenek, hogy hálózati eszközeik minél többféle hálózatot legyenek képesek összekapcsolni.

Az angol szakirodalom a több, egymással összekötött hálózatot **internetworknek**, a hálózatok összekötésére szolgáló eszközöket pedig **internetworking devicesnek** nevezi. Ez utóbbi kifejezés nem minden esetben jelent hardvert. Sok esetben az összeköttetést a fizikai összeköttetésen felül szoftveresen hozzák létre. Az olyan hálózatot, amely nem tartalmaz összekapcsoló eszközt, **szegmensnek** nevezzük. A szegmens olyan sínhálózat,

amelynek kábelhosszát az előírt technikai feltételek korlátozzák. Ebből következik, hogy az internetwork szegmensekből, és az ezek összekapcsolására alkalmas eszközökből épül fel. Előfordulhat, hogy a szegmensek összekötését biztosító szoftver vagy hardver nem szerver, hanem valamelyik munkaállomás része.

A különálló számítógépek és egyéb eszközök hálózattal való kapcsolatát valamilyen, a gépbe épített **hálózati kártya**, illetve belső vagy külső **modem** biztosítja. A hálózati kártyák sokfélék; felépítésüket és csatlakozó felületüket meghatározza a számítógép és a hálózat architektúrája is. A modem hagyományos telefonvonalon, ISDN (Integrated Services Digital Network: integrált szolgáltatású digitális hálózat), DSL (Digital Subscriber Line: Digitális Előfizetői Vonal) vonalon, vagy kábeltévés hálózaton keresztül teszi lehetővé az adatátvitelt. Létezik gépbe építhető kártya formájában és a számítógéphez kívülről csatlakoztatható változatban is.

A **switch** (adatátviteli kapcsoló) olyan szerkezeti elem, amely a hozzá csatlakoztatott eszközök közötti adatáramlást valósítja meg; útvonalszegmensek időleges egymáshoz rendelésével épít fel kommunikációs útvonalat.

A **repeater** (jelismétlő) olyan elektronikus eszköz, amely az adatátvitel során, a csillapítás következtében torzult jelek felismerését, helyreállítását és újrarendőzítését végzi. Az eszköz neve is mutatja, hogy egyszerűen a vett jeleket felerősíti, ismétli. Ebből következik, hogy a két hálózatnak, amelyeket összekötünk a repeateren keresztül, azonos típusúaknak kell lenniük, hiszen a repeater a rajta áthaladó jelek tartalmi részében semmilyen változtatást nem végez. A repeater tehát két azonos típusú szegmens összekötésére alkalmas passzív eszköz. A repeater a hálózat számára átlátszó, működése a hálózatban nem észrevehető. Használatára akkor van szükség, ha a hálózat két csomópontja igen távol van egymástól, és a továbbítandó jelek oly mértékben vesznek el, hogy a két szegmens közötti kapcsolat nem jön létre, vagy minősége nem megfelelő. Ennek a távolságnak a mértéke különböző hálózattípusoknál más és más. Elméletileg így bármilyen hosszúságú hálózatot kialakíthatnánk, hiszen egyszerűen bizonyos távolságonként jelismételőket kellene pusztán beillesztenünk jelfrissítés céljából. A gyakorlatban az egyes hálózatok előírják a két csomópont között elhelyezhető repeaterek maximális számát.

A **bridge** (hálózati híd) azonos architektúrájú, de különböző protokollok segítségével működő hálózatok összekapcsolását teszi lehetővé. A bridge bizonyos mértékig belelát az üzenetekbe: éppen annyira, hogy képes kiolvasni a csomópontok fizikai címét. Ez azt jelenti, hogy minden egyes csomagról meg tudja mondani, hogy ki a feladó és ki a címzett, vagyis ki küldi kinek. Mivel meg tudja állapítani címzett kilétét, képes eldönteni, hogy melyik szegmensbe kell továbbítani az adott csomagot. A hálózati hidak a szegmensek közötti adatforgalmat bonyolítják le.

A **router** (forgalomirányító) rendszerint egy célszámítógép, amely két, vagy több hálózatra van kapcsolva és továbbítja az adatcsomagokat a hálózatok között. A routerek elrejtik a hálózat szerkezetét a hosztok és felhasználók elől, és lehetővé teszik, hogy azok az összekapcsolt hálózatokat egyetlen nagy hálózatnak tekintsék. A routerek belenyúlhatnak az üzenetbe (pontosabban a fejrészbe), és ettől nevezhető a feladatuk igazából útvonal-kijelölésnek. A router tehát egy intelligens eszköz, amely meghatározza a hálózaton áramló adatcsomagok útvonalát. A routerek elsősorban abban különböznek egymástól, hogy az üzenet továbbküldéséhez a legrövidebb utat, vagy a legkevesebb csomóponton való

áthaladást tekintik jobb választásnak. A korszerűbb forgalomirányítóknál a rendszergazda beavatkozhat a folyamatba úgy, hogy egyes utakat súlyokkal lát el.

Mivel nincs túl nagy különbség a routerek és bridge-ek között, a gyártók elkezdtek olyan routerek előállítását, amelyek bridge-ek szerepét is képesek ellátni. Ezeket az eszközöket nevezzük **routemnek**.

A **gateway** (átjáró, hálózati zsilip) feladata igen komoly, mivel a gyökeresen eltérő hálózatok közötti tolmácsolást biztosítja. A gateway ezt úgy oldja meg, hogy a kapott csomagot kibontja, és annak tartalmát lefordítja a célhálózat nyelvére. Mivel a protollok átalakításáról minden szinten gondoskodni kell, egy általános hálózati zsilip tervezése rendkívül bonyolult, ezért a gatewayeket tipikusan úgy alakítják ki, hogy két specifikus hálózati architektúrához illeszkedjenek. A kisebb típusok általában kártya formájában vásárolhatók meg, a nagyobb kapacitásúak viszont célszámítógépek.

5.9. Az internet és fő szolgáltatásai

Az Egyesült Államokban a 60-as években merült fel egy **kevésbé sebezhető számítógép-hálózat** kiépítésének szükségessége. 1969-ben kezdett működni az **ARPANET**, amelyet a katonai bázisok és kormányzati kutatólaboratóriumok használtak elektronikus levelezésre, állományok cseréjére, valamint távoli bejelentkezésre egymás számítógépei között. 1974-ben jelent meg először az „**internet**” kifejezés. 1983-ban az ARPANET-ről leválasztották a hadászati részt, és ezzel megszületett a mai értelemben vett internet. A 80-as évek végén számos országban szerveződtek gerinchálózatok (nagysebességű kommunikációs hálózatok), amelyek egymással kiépítették közvetlen kapcsolataikat. A távközlési cégek is meglátták az üzleti lehetőséget az internet technológiájú számítógép-hálózatokban, és a hozzájuk kapcsolódó alkalmazásokban (számítógépek és adatbázisok távoli elérése, elektronikus levelezés, adatállományok átvitele, szöveg-, kép-, hanginformációk továbbítása) így megjelentek az ilyen szolgáltatásokat kínáló üzleti vállalkozások.

Az interneten minden gépnek egyedi **IP (internet protocol)** címmel kell rendelkeznie. Az IPv4 címek 32 bit, az IPv6 címek 128 bit hosszúak (például a 160.114.68.2 a Szegedi Tudományegyetem Mérnöki Karán az egyik gépnek a címe). Az egyedi címkiosztást egy erre a célra kijelölt nemzetközi szervezet és az általa megbízott országos szintű szervezetek biztosítják. Kidolgoztak egy olyan módszert is, amely segítségével a gépekre számok helyett egy, a számhoz tartozó elnevezéssel hivatkozhatunk. A **hosztnév** három (vagy több) tagból áll: a.b.c, ahol a, b, c betűket és számokat tartalmazó szöveg. Az első pontig tartó rész a gépen kínált szolgáltatás típusára (www, ftp, mail stb.), a középső része a szolgáltató, cég, illetve intézmény nevére vagy annak rövidítésére (mk, u-szeged), a jobb oldali rész arra az országra utal (hu, eu, uk, de, us stb.), ahol a szolgáltató működik, illetve amilyen tevékenységgel foglalkozik (org, gov, com stb.).

Az internet fő szolgáltatásai:

- World Wide Web: A webszerverre felmásolt weboldalakat a kliens gépen böngészőprogrammal (Chrome, Edge, Firefox, Opera, Safari stb.) tekinthetjük meg a megfelelő cím (például <https://www.mk.u-szeged.hu> beírásával. A szolgáltatás fő protokollja a HyperText Transfer Protocol (HTTP), illetve annak biztonságos változata (HTTPS).
- Elektronikus levelezés: A levelező szerveren található levélfiókhoz azonosító és jelszó megadása után férhetünk hozzá böngészőprogrammal vagy a felhasználó

számítógépre telepített levelező program (eM Client, Mailbird, Mailspring, Outlook, Pegasus Mail, Thunderbird stb.) segítségével. A hozzáféréshez szükséges egyéb beállításokat (kimenő és bejövő levelek kiszolgálója, titkosítás stb.) a szolgáltató adja meg. Az e-mail cím általános formája a@b.c. A szolgáltatás egyik fontos protokollja a Simple Mail Transfer Protocol.

- Fájlmásolás: Az ftp (file transfer protocol) szerveren tárolt állományok kezelése (feltöltés, letöltés) a felhasználó számítógépen ún. ftp kliens programok (FileZilla, Total Commander, Transmit stb.) segítségével történik. Az ftp szerverre saját azonosítóval és jelszóval (illetve nyilvános szerver esetén általában „anonymous” vagy „guest” azonosítóval és – a jelszó helyett – e-mail címmel léphetünk be. A szolgáltatás fő protokollja a File Transfer Protocol.
- Egyéb szolgáltatások: azonnali üzenetküldés csevegő rendszerekkel (Skype, Viber, Messenger stb.), internet protokoll feletti hangátvitel (Voice over IP, VoIP: Skype, Viber), internet tévé (IPTV).

5.10. Ellenőrző kérdések

1. Mit nevezünk számítógép-hálózatnak? [50]
2. Mire szolgál a protokoll? [50]
3. Mi a hand-shaking? [50]
4. Mi a munkacsoport és a tartomány-elvű hálózat? [50]
5. Mi a hoszt? [50]
6. Mi a szerver és a kliens feladata? [50]
7. Milyen fő szolgáltatásokat nyújthatnak a szerverek? [50]
8. Milyen céllal alakíthatnak ki számítógép-hálózatot? [51]
9. Hozzáférhetőség szerint milyen hálózatok léteznek? [51]
10. Területi elhelyezkedés szerint milyen hálózatokról beszélhetünk, és hogyan lehet jellemezni ezeket a hálózatokat? [52]
11. Mit jelent, hogy a hozzáférést hozzáférési szintek szabályozzák? [53]
12. Milyen felhasználói jogosultságok léteznek általában és mi jellemző azokra? [53]
13. Milyen tipikus felhasználói csoportok léteznek a hálózaton és mi jellemző a csoportokra? [53]
14. Mi a hálózati topológia és milyen ismertebb változatai léteznek, hogyan jellemezhetők? [54]
15. Milyen módjai vannak hálózati kommunikációnak? [55]
16. Mit értünk versenyhelyzet alatt és hogyan kezelhető ez a jelenség a hálózaton? [56]
17. Összekapcsoló közeg szerint milyen kapcsolattípusok vannak és mik a fő jellemzőik? [56]
18. Mi az internetwork, internetworking device és a szegmens? [57]
19. Mi a hálózati kártya és a modem feladata? [58]
20. Mi a switch feladata? [58]
21. Mi a repeater feladata? [58]
22. Mi a bridge feladata? [58]
23. Mi a router feladata és a routem? [58]
24. Mi a gateway feladata? [59]
25. Miért és hogyan alakult ki az internet? [59]
26. Mi az IP-cím és a hosztnév? [59]
27. Melyek az internet fő szolgáltatásai és hogyan jellemezhetők azok? [59]

6. OPERÁCIÓS RENDSZEREK ÉS ÁLLOMÁNYKEZELÉS

Elvárt tanulási eredmények

Tudás

- Ismerteti operációs rendszer definíciókat
- Ismeri az operációs rendszer fő feladatait
- Megérti az operációs rendszer helyét a számítógépes rendszerben
- Érti a bootolás és a kompatibilitás lényegét
- Ismeri a multiprogramozás fogalmát és két változatának jellemzőit
- Felsorolja az operációs rendszert kiegészítő segédprogramok feladatait
- Ismeri a shell és a kernel feladatait
- Megnevezi a legelterjedtebb operációs rendszereket
- Felsorolja az operációs rendszerek csoportosításának szempontjait
- Érti a karakteres és grafikus felhasználói felületek jellemzőit és használati módjait
- Ismeri az állományszervezés, az állománykezelő rendszer, valamint az állomány és a fájlrendszer fogalmát
- Felsorolja az állományok tulajdonságait
- Elmagyarázza az állományok háttértáron való tárolásának elvét
- Ismeri az adatállomány két alaptípusát és alapvető jellemzőit
- Ismeri az állományok két részének fontosabb jellemzőit és a kiterjesztés jelentőségét
- Felsorolja és csoportosítja a gyakori állománytípusokat
- Felsorolja a mappákon és állományokon végezhető műveleteket

Képesség

- Működteti a számítógépet
- Kommunikál az operációs rendszerrel annak felhasználói felületén
- Műveleteket végez mappákon és állományokon

Attitűd

- Törekszik az operációs rendszerek valamennyi szolgáltatásának megismerésére
- Figyelembe veszi a számítógépre telepített operációs rendszer szolgáltatásait és korlátait
- Nyitott az egymás mellett létező különböző operációs rendszerek használatára és az egymást követő rendszerek újdonságainak kipróbálására

Autonómia, felelősség

- Szakmai útmutatással telepít operációs rendszert
- Szakmai irányítás mellett használja az operációs rendszer szolgáltatásait
- Önállóan kezeli a háttértárakat és az állományokat

6.1. Az operációs rendszerek fogalma, fő feladatai

Egy számítógép működéséhez nem feltétlenül szükséges operációs rendszer. Az egyes programok akár közvetlenül is betölthetők és használhatóak a hardveren feltéve, hogy az adott program írója vállalja azt, hogy mindent közvetlenül, operációs rendszeri segítség nélkül fog megoldani. A kezdeti számítógépek esetén ez volt a normális működési mód: nem használtak operációs rendszereket, minden egyes új program elindításához a gépet újra kellett indítani. Az idő előre haladtával egyes apró segédprogramok állandósulni kezdtek, azokat több programhoz is használták, és kialakultak azok a szokásos programrészek, melyeket újraindítás után újra és újra használni szerettek volna. Ezekből alakultak ki a kezdeti operációs rendszerek.

Az **operációs rendszerre számos definíció** született. A definíciók a rendszer által nyújtott szolgáltatásokon keresztül értelmezik a fogalmat. Néhány példa:

- Az operációs rendszer a számítógép bekapcsolásától egészen annak kikapcsolásáig folyamatosan működik. Felügyeli minden más program működését.
- Az operációs rendszerek interfészként szolgálnak a felhasználó és a számítógép között.
- Kezeli a számítógép erőforrásait, továbbá alapot biztosít a többi program írásához. Feladta, hogy megvédje a programozókat a hardver bonyolultságától úgy, hogy a hardver fölé egy szoftverréteget képez a felhasználó számára, amely a kezeli a rendszert és kapcsolódási felületet (virtuális gépet) alkot a felhasználó számára, amelyet könnyebb programozni és kezelni.
- Program, amely lehetővé teszi, hogy a használni tudjuk a számítógép által biztosított lehetőségeket. Operációs rendszer nélkül nem tudnánk használni a gépet, ez a számítógép „alapprogramja”. A számítógépre akár több operációs rendszer is telepíthető, de egy számítógépen egyszerre mindig csak egy futhat.
- A számítógépen futtatott legalapvetőbb program, amelynek elsődleges feladata összehangolni a hardverelemek működését, valamint felületet biztosítani a felhasználó számára, hogy az tudjon a géppel kommunikálni. Emellett gondoskodik a rendelkezésre álló erőforrások szétosztásáról, ütemezi a gépen futó egyéb programok működését és egységes futtatási környezetet biztosít a programok számára.
- Különböző hardver erőforrások kiosztását és használatát irányító program. Olyan keretet, szolgáltatásokat biztosít, amelyre a többi szoftver épít.
- Sok esetben nem egyetlen program, hanem programok együttese, amelyek közvetítő szerepet töltenek be a felhasználó és a számítógép erőforrásai között. Feladatai közé tartozik a memóriakezelés, fájlkezelés, az operatív memória és a háttértár közötti adatforgalom (betöltés, mentés) lebonyolítása, a felhasználói programok felügyelete, illetve ezzel összefüggésben a hatékony végrehajtás elősegítése. Támogatja, ellátja a rendszerfelügyeletet, a karbantartási és védelmi feladatokat.
- Az operációs rendszer rendszerműködtető program(ok) és számos, különböző feladatokat ellátó segédprogram integrált rendszere. Az operációs rendszer (vagy annak egy része) folyamatosan az operatív memóriában van a gép bekapcsolt

állapotában. A különféle operációs rendszerek szolgáltatásai között lehetnek eltérések, de mindegyiknek el kell látnia következő feladatokat:

- a számítógép bekapcsolásakor a rendszer indítása (bootolás),
 - programok futtatása,
 - a számítógép erőforrásainak kezelése.
- Az **ISO** nemzetközi szabványosítási szervezet **definíciója** szerint: „Olyan programrendszer, amely a számítógépes rendszerben a programok végrehajtását vezérli: így például ütemezi a programok végrehajtását, elosztja az erőforrásokat, biztosítja a felhasználó és a számítógépes rendszer közötti kommunikációt.”

Létezik **technológiai, funkcionális, folyamatcentrikus és felhasználói szemléletű megközelítése** is a fogalomnak, eszerint az operációs rendszer:

- a számítógépet alkotó hardver eszközök működését felügyelő és vezérlő program (technológiai megközelítés),
- számítógép tevékenységét meghatározó programokat felügyelő és vezérlő szoftver (funkcionális megközelítés),
- a számítógépes rendszerben rendelkezésre álló) erőforrásokat elosztó szuperfolyamat (folyamat-centrikus szemlélet),
- olyan program, amely kapcsolatot teremt (és tart fent) a számítógépet alkotó technikai-technológiai (hardver) elemek és a (számítógéppel tevékenységet végző) felhasználó között (felhasználói szemléletű definíció).

Nézzük ezek után tehát az operációs rendszerek **fő feladatait**:

- Kapcsolattartás a felhasználóval: A felhasználó elsősorban az operációs rendszer által biztosított felhasználói felületen kommunikál a számítógéppel.
- Az erőforrások kezelése: Minden számítógépben található és ahhoz csatlakoztatott hardvert az operációs rendszer felügyel és kezel. A felhasználók és a programok is az operációs rendszeren keresztül érhetik el a hardver eszközöket. Ha egy perifériát nem ismer fel az operációs rendszer, azt nem tudja kezelni, így az nem elérhető sem a felhasználók sem a többi program számára. Előfordul, hogy több program versenyez ugyanannak az erőforrásnak a használatáért és ilyenkor néha (és általában a nem megfelelően megírt programnak köszönhetően) „lefagy” a gép, nem reagál sem egérre, sem billentyűzetre.
- Processzorütemezés: Az operációs rendszer határozza meg, hogy mikor és melyik program használhatja a processzort (továbbá többprocesszoros gépben azt is, hogy melyik processzort).
- Memóriakezelés: Az operációs rendszernek kell minden periféria és program számára elegendő memóriaterületet biztosítania a megfelelő működéshez és arról is gondoskodnia kell, hogy a párhuzamosan futó programok ne tudjanak beleírni egymás memóriaterületébe. Ha már nem fut a program, akkor gondoskodni kell a rendszernek a memória felszabadításáról is.
- Háttértárkezelés: Ez a háttértárakon tárolt adatok tárolását és mozgatását, ellenőrzését jelenti.

- Felhasználói utasítások értelmezése és végrehajtása: Az operációs rendszer biztosít egy karakteres vagy grafikus felületet, ahol a felhasználó rendszerszintű feladatokat (például rendszerbeállítások módosítását, biztonsági mentéseket, állománykezelést) végezhet el.
- Programok futtatása: Az operációs rendszernek biztosítani kell, hogy a futó programok a működésükhöz szükséges eszközökhöz hozzáférjenek, azokat használni tudják.
- Hibajelenségek kezelése: A műveletek végrehajtása során felléphetnek problémák, amelyeket az operációs rendszernek vagy jeleznie kell a felhasználó felé, vagy meg kell próbálnia kezelni – akár a felhasználó tudta nélkül.
- Hálózati szolgáltatások: Bár sokáig nem tartozott az operációs rendszerek feladatkörébe, de ma már nem nagyon találkozhatunk olyan rendszerrel, amely ne támogatná a helyi és távolsági hálózatok elérését, illetve a számítógép-hálózaton folytatott munkát. A mai számítógépek általában nem önállóan üzemelnek, hanem valamilyen módon össze vannak kapcsolva. A felhasználó az operációs rendszer segítségével ezen a kapcsolaton keresztül adatokat vihet át az egyik gépről a másikra; az egyik gépről bejelentkezhet egy távoli számítógépre, és azon úgy dolgozhat, mintha közvetlenül a távoli számítógép előtt ülve végezne munkát. Az operációs rendszer feladata ilyenkor az, hogy a megbízhatatlan, rossz minőségű csatornán megbízható kommunikációs csatornát biztosítson a felhasználóknak, amiben az egyik gépről a másikra küldött adatok nem vesznek el, és az adatokat a fogadó állomás az elküldés sorrendjében kapja meg.

Az **operációs rendszer helyét a számítógépes rendszerben** mutatja a 13. ábra:

- Legalul, az 1. szinten helyezkedik el a több rétegből álló hardver. A hardver legalsó rétege tartalmazza a fizikai eszközöket (az integrált áramköri lapkákat, a huzalozást, áramellátást stb.)
A következő réteg a mikroarchitektúra, ahol a fizikai eszközöket működési egységek szerint csoportosítják (tartalmazhat processzort, regisztereket, buszrendszert stb.). A mikroarchitektúrát valamilyen módon vezérelni, programozni szükséges. A programozók számára elérhető hardver és utasítások együttese az utasításkészlet-architektúra, vagy gépi nyelv.
- A hardverre épülnek rá a 2. szinten a rendszerprogramok, amelynek első rétege az operációs rendszer, amely elrejti az alatta lévő rétegek bonyolultságát a programozó és a felhasználó elől, valamint egy kényelmesebb, egyszerűbben használható utasításkészletet biztosít. Az operációs rendszer felett található a rendszerprogramok maradék része: a programozói nyelveken írt utasításokat, programokat gépi nyelvre fordító program, az operációs rendszer parancsainak fordítását végző parancsértelmező és egy egyszerű szövegszerkesztő.
- Végül a harmadik, utolsó szinten találjuk a különféle felhasználói programok.

A számítógép bekapcsolása után az operációs rendszer (szükséges része) automatikusan betöltődik a háttértárból a memóriába; ezt a folyamatot **„boot”-olásnak** nevezzük. A számítógéppel a bootolás befejezése után hajthatunk végre feladatokat.

Banki rendszer, repülőjegy foglalási rendszer, webböngésző, elektronikus levelező stb.	Felhasználói programok
Fordító programozási nyelvhez, parancsértelmező, szövegszerkesztő	Rendszerprogramok
Operációs rendszer	
Gépi nyelv	Hardver
Mikroarchitektúra	
Fizikai eszközök	

13. ábra. A számítógépes rendszer felépítése.

Forrás: Tanenbaum–Woodhull (2007) alapján a szerző szerkesztése

A felhasználók által használt programok az operációs rendszeren keresztül érik el a számítógép erőforrásait. Ebből következően a programok egy-egy adott operációs rendszerrel kell, hogy együttműködjenek, azzal, kell, hogy **kompatibilisek** legyenek.

A bonyolultabb hardver rendszerekre egyre bonyolultabb operációs rendszereket építettek. Megjelent a **multiprogramozás**, a mai operációs rendszerek egy alapvető fontosságú tulajdonsága. A multiprogramozásnak két változata van: a **többfeladatos** (multitasking) illetve a **többfelhasználós** (multiuser) rendszer (és ez a két forma kombinálható is).

- A többfelhasználós rendszerekben egy központi egységen (processzoron) osztozik több felhasználó, de a központi egység nagy sebessége miatt minden felhasználó úgy érzi, hogy egy saját gépe van, amin csak ő dolgozik.
- A többfeladatos rendszer annyit tud, hogy ott egy felhasználó egyszerre több feladatot indíthat el, és az elindított feladatok egyszerre (látszólag), párhuzamosan hajtódnak végre

Az operációs rendszer általában nem egyetlen program, hanem sok kisebb programból álló rendszer. Az előzőek alapján mondhatjuk, hogy egy nagyon **összetett folyamatokból álló programrendszer**ről van szó.

A mai operációs rendszerekre jellemző, hogy a fenti feladatokat ellátó alaprendszert – szinte kötelező jelleggel – kiegészítik néhány, általános kisebb, de annál fontosabb feladatot ellátó **segédprogrammal** a következő feladatok ellátásához:

- Állománykezelő: a háttértárakon található programok és adatok kezeléséhez;
- Adminisztrációs programok: a felhasználók és jogaik beállításához, a teljesítmény figyeléséhez, naplózáshoz stb.;
- Víruskereső és tűzfal: a számítógépes kártevők elleni védelem biztosításához.

6.2. Az operációs rendszerek fajtái, alapvető részei, csoportjai

A ma **legelterjedtebb operációs rendszerek** a Windows, a Linux és a MacOS valamelyik változata, illetve verziója (személyi számítógépeken és szervereken), míg hordozható eszközökön a Google Android és az Apple iOS rendszerek fordulnak elő leggyakrabban.

Az operációs rendszert **shellre** és **kernelre** oszthatjuk. Vannak olyan operációs rendszerek, melyekben a kernel és a shell különválnak, így lehetőség nyílik ezek kombinációjára (például

Linuxban), más rendszereknél, a különválasztás mind elméleti, mind gyakorlati szempontból nehezebb (Windowsban):

- A shell (felhasználói felület) az, amelyen keresztül a felhasználó és a számítógép kommunikál egymással. Feladatai közé tartozik a kapcsolattartás a felhasználóval karakteres vagy grafikus felhasználói felületen, valamint az alkalmazások futásának kezelése (indítás, futási feltételek biztosítása, leállítás).
- A kernel (mag) az rész, amely közvetlenül a hardverrel áll kapcsolatban. Fontosabb feladatai a perifériák kezelése, a memória-hozzáférés biztosítása, a processzor idejének elosztása, háttértárolók és fájlrendszerek kezelése.

Az **operációs rendszerek csoportosítására** többféle szempont szerint lehetséges:

- Felhasználói felület szerint:
 - Karakteres felület (CLI: command line interface): a legkisebb megjeleníthető és kezelhető egység a karakter, kezelése elsősorban billentyűzettel történik.
 - Grafikus felület (GUI: graphical user interface): a legkisebb megjeleníthető és kezelhető egység a pixel (képpont), a kezelése elsősorban egérrel (vagy más pozicionáló eszközzel) történik.
- Felhasználók száma szerint:
 - Egyfelhasználós rendszer: a számítógépet és az operációs rendszerét egyetlen felhasználó kiszolgálására tervezték.
 - Többfelhasználós rendszer: a rendszer több felhasználó – nem egyidőben történő – kiszolgálására képes. Ilyenkor minden felhasználó saját azonosítóval és jelszóval rendelkezik, amelyhez különböző jogosultságok kapcsolódhatnak. A felhasználók saját felhasználói felülettel rendelkeznek. (Kliensgépen futó operációs rendszer.)
 - Hálózati operációs rendszer: Olyan rendszer, amely akár több felhasználó egyidejű kiszolgálására is képes. (Szervergépen futó operációs rendszer.)
- Folyamatkezelés szerint:
 - Köteget rendszer: A programok előre összeállított sorrendben és módon futnak le, amelynek menetébe a felhasználók interaktív módon nem avatkozhatnak be.
 - Multiprogramozott rendszer: Egyszerre több program futtatására képesek. Ezek a rendszerek tovább bonthatók:
 - Valós idejű rendszerekben a keletkező adatokat nem gyűjtik, hanem azonnal feldolgozzák. Olyan területeken alkalmazzák, ahol a környezetben bekövetkező változásokra nagyon gyorsan kell reagálni (például ipari folyamatok irányításában).
 - Az időosztásos rendszer több felhasználó számára áll egyidejűleg rendelkezésére, akik interaktívan adhatják ki a rendszer működtetéséhez szükséges parancsokat, futtathatják programjaikat. A felhasználók látszólag egyidejű kiszolgálása úgy történik, hogy az operációs rendszer periodikusan a felhasználók mindegyikének ad egy-egy időszeletet, amelyikben a rendszer csak az adott

felhasználónak dolgozik. A váltások olyan gyorsak, hogy a felhasználók úgy érzik, mintha egyedül használnák a rendszert.

- Elosztott rendszerekben a programrészek, számítások több központi egység (processzor) között oszlik meg, így az erőforrások jobban kihasználhatók és nagyobb a rendszerek megbízhatósága is.
- Hardver-architektúrák szerint:
 - Számítógép-kategóriák: lásd a hardver alapismereteket tartalmazó fejezetben!
 - A processzor-architektúra alapján: a bonyolult utasításkészlettel dolgozó CISC (Complex Instruction Set Computer) és a csökkentett utasításkészlettel dolgozó RISC (Reduced Instruction Set Computer) processzorra készülő operációs rendszerek.
 - Memóriakezelés alapján: aszerint, hogy mekkora memóriaterületet képes az operációs rendszer elérni, kezelni, beszélhetünk 16 bites, 32 bites, 64 bites rendszerekről.
- Jogállás szerint:
 - Szerzői jogvédelem alá tartozó operációs rendszerek: Ezek a rendszerek általában csak díj ellenében használhatók, általában a módosításuk és továbbadásuk nem megengedett. A felhasználó csak a lefordított, a gép processzora által értelmezhető gépi kódú programot kapja meg használatra.
 - Nyílt forráskódú rendszerek: Az ilyen rendszereknek az eredeti, magas szintű programozási nyelven írt változata is rendelkezésre áll a felhasználó számára (sőt az is előfordul, hogy csak az). A felhasználó igényei és tudása szerint módosíthatja, átírhatja az operációs rendszert és a módosításokat (illetve azok forráskódját) is terjesztheti.
- Felhasználói preferenciák szerinti csoportosítás: Ide sorolhatók azok a szempontok, amelyek szerint a felhasználók általában döntenek az általuk használt operációs rendszer mellett, például felhasználói felület, kezelhetőség, megbízhatóság, gazdaságosság, támogatás, a rendszer készítője stb.

6.3. Az operációs rendszerek felhasználói felülete

Az operációs rendszer és a felhasználó közötti **kommunikáció kétirányú**: egyrészt a felhasználó jelez a rendszernek, ha valamilyen feladatot végre akar hajtatni, másrészt a rendszer küld üzenetet a kért feladat végrehajtásáról.

A **kapcsolattartás módja** is kétféle lehet:

- **közvetlen** kapcsolattartásról akkor beszélünk, ha a felhasználó egy operációs rendszer által biztosított interaktív felületen adja ki a parancsot, míg
- **közvetett** kapcsolattartásról akkor, ha a felhasználó egy programon keresztül kiadott utasítással jelzi, hogy milyen feladat elvégzését várja.

Az operációs rendszernek azt a részét, amely a kommunikációért felel, **felhasználói felületnek** (interface, interfész) nevezzük. A felhasználói felület lehet karakteres vagy grafikus.

Az **operációs rendszer információt szolgáltat** a felhasználó számára. Az információk lehetnek:

- **Prompt üzenetek:** Ezzel jelzi az operációs rendszer a felhasználó felé, hogy utasításra, vagy egyéb információra vár. (Ez a fajta üzenet elsősorban karakteres felhasználói felületen fordul elő, grafikus felületen nem jellemző.)
- **Kisegítő üzenetek:** Segítő, magyarázó üzenet a felhasználó számára arról, hogy az adott helyzetben mit kell tennie.
- **Előrehaladási üzenetek:** Hosszantartó folyamat esetén tájékoztatja a felhasználót arról, hogy hol tart éppen a folyamat. Ez az információ a hátralévő idő, illetve a folyamat befejezéséhez szükséges idő megbecslésében is segíti a felhasználót.
- **Befejezést jelző üzenetek:** Egy folyamat végét, befejezését jelzik a felhasználónak. Amennyiben a folyamat hibaüzenettel áll le, akkor erről ad tájékoztatást az operációs rendszer.

A karakteres felhasználói felület

A **karakteres felhasználói felület** használatakor a shell legfontosabb feladata a felhasználó által begépelte utasítás értelmezése (a parancsértelmező által). Az operációs rendszer által ismert (értelmezhető) parancsok együttese az **utasításkészlet**.

Az **utasításszerkezet** határozza meg, hogy az utasításkészlet egyes elemeihez hány és milyen paraméter és kapcsoló tartozhat (meghatározza az adott utasítás viselkedését), milyen módon kell az egyes részeket elhatárolni, mekkora lehet egy utasítássor maximális hossza stb. Az utasításszerkezet általában egy utasításon belül a következő három funkcionális egység megkülönböztetését teszi lehetővé:

kulcsszó paraméterek kapcsolók

Az utasításszerkezetnek minden esetben kulcsszóval kell kezdődnie. Az utasítás végét (lezárását) az operációs rendszer számára egy kitüntetett szimbólum (például enter) jelzi. A paraméterek/kapcsolók száma utasításonként eltérő lehet.

Példa: Egy adott háttértár teljes tartalmának kilistázása a képernyőre:

Windows 10 parancssorban: `dir /s`

PowerShellben: `Get-ChildItem -recurse`

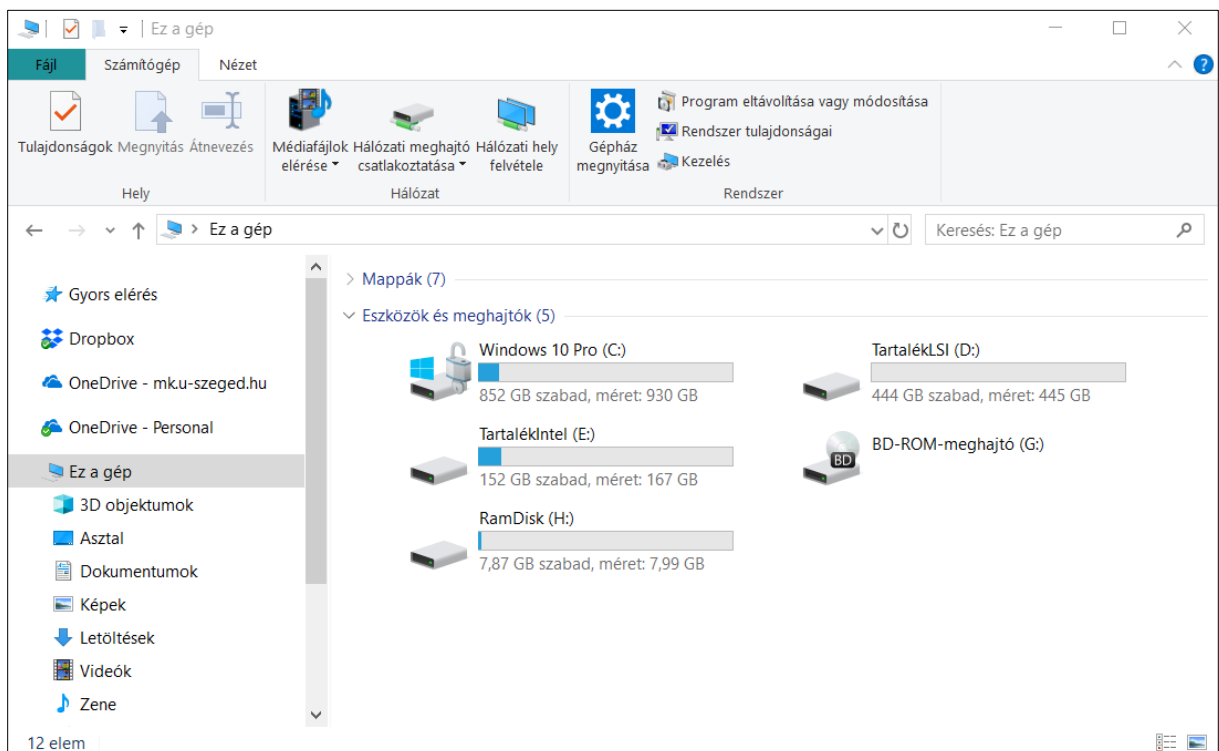
Unix, Linux parancssorban: `ls -R`

Az operációs rendszer az utasítás lezárása után szintaktikailag **ellenőrzi** a kapott parancsot, majd megpróbálja a megadott paraméterek és kapcsolók értékei alapján **végrehajtani**. A végrehajtás eredményével kapcsolatos információk kezelése szempontjából alapvetően két megközelítés fordul elő: a pozitív és a negatív nyugtázás. **Pozitív nyugtázás** alatt azt értjük, hogy az operációs rendszer a sikeresen elvégzett parancs végrehajtásáról ad valamiféle visszajelzést a felhasználónak, a **negatív nyugtázásos** rendszerben ezzel szemben csak a hibás parancsok váltanak ki reakciót az operációs rendszer részéről.

A grafikus felhasználói felület

A **grafikus felületű** rendszerek működésének hátterében ugyanazok a tevékenységek zajlanak, mint a karakteres felület esetén, csak éppen az eszközrendszer különbözik. A felhasználói felület fontosabb **elemei** (14. ábra):

- Ablak: minden folyamathoz hozzárendelhető a képernyő egy (alapértelmezés szerint téglalap alakú) területe: ez a terület a program futási környezete.
- Ikon: kisméretű grafikus ábra, amely egy objektum vagy tevékenység reprezentálására szolgál.
- Menü: tevékenységek szöveges megjelenésű, strukturált rendszere.



14. ábra. Windows 10 ablak ikonokkal és menüvel.

A billentyűzet mellett (vagy inkább helyett) pozicionáló eszközt használhatunk a különböző szolgáltatások eléréshez és kiválasztásához. A pozicionáló eszközök sokfélék lehetnek: egér, botkormány, gördítő-csúszó kijelölő (touch-ball, touch-pad), érintőképernyő stb.

Attól függően, hogy milyen módon érhetjük el az operációs rendszer szolgáltatásait, **háromféle vezérlést** különböztethetünk meg:

- A grafikus felület ablakos rendszerére jellemző az eseményvezérlés. Ennek alapelve, hogy az operációs rendszer a felhasználó tevékenységét, valamint a rendszer működésében megjelenő állapotváltozásokat is folyamatosan figyeli. Ha a rendszer környezetében valamilyen változás áll be, akkor erről a grafikus felület üzenetet küld az operációs rendszernek, az pedig az üzenet információi alapján meghatározza az elvégzendő műveletet.
- Menüvezérlésről beszélünk akkor, ha az operációs rendszer szolgáltatásait és az utasításokat menükre, menüpontokra kattintva érhetjük el.

- Parancsvezérlést leginkább karakteres felületen alkalmazunk, de grafikus felületen is találkozhatunk vele. Itt az utasítás begépelésével adunk parancsot a gép számára.

6.4. Állományszervezés

Az **állományszervezés** az operációs rendszer mindazon tevékenységeinek összessége, amely a felhasználói információk háttértárolón történő elhelyezésével, azonosításával és visszakeresésével kapcsolatos.

Az **állománykezelő rendszer** logikai alapegysége az állomány. Az **állomány** (file, fájl): a háttértáron elhelyezkedő adatok egyedi azonosítóval rendelkező együttese; a felhasználó szempontjából az adattárolás egysége.

A **fájlrendszer** az operációs rendszerekben a fájlok elnevezését, tárolását és szervezését meghatározó általános rendszer. Windowsban a **fájlrendszer típus** lehet például FAT (File Allocation Table, kisebb háttértárakhoz), a FAT32 (nagyobb méretű háttértárakhoz), exFAT (Extended File Allocation Table, amelyet pendrive-okhoz, memóriakártyákhoz fejlesztettek ki) valamint NTFS (New Technology File System, a nagyobb méretű háttértárakhoz, biztonsági, helyreállítási szolgáltatásokkal kiegészítve). Más operációs rendszerekben más fájlrendszerekkel találkozhatunk (Linuxban például ext2, ext3, xfs stb.).

A felhasználó által megadott azonosító alapján az állománykezelő **meghatározza a keresett információ fizikai helyét**, és **utasítja** a háttértár író/olvasó mechanizmusát a megfelelő **művelet** elvégzésére.

A felhasználó valamilyen alfanumerikus karaktersorozattal hivatkozhat a tárolt adatokra – ez az állomány neve. Az adott **operációs rendszer névhasználati szabályai** határozzák meg, hogy az állományok neve milyen hosszú lehet (hány karakterből állhat) és milyen szimbólumokat tartalmazhat. Az operációs rendszertől függ az is, hogy milyen egyéb tulajdonságokat tart nyilván az egyes állományokról. A legáltalánosabban használt **tulajdonságok** a következők:

- Típus: Az állomány fajtája (lehet például szöveg, táblázat, adatbázis, hang, kép, videó, futtatható program stb.).
- Időbélyegek: A létrehozás, módosítás, megnyitás időpontját tartalmazhatja.
- Méret: Az állományban tárolt információ mennyisége, általában byte-ban, illetve többszöröseiben kifejezve.
- Jellemzők (attribútumok): Operációs rendszerenként változó kiegészítő jellemző (például írásvédett, rejtett, archiválandó, rendszerfájl, futtatható fájl stb.).
- Jogosultságok: Amennyiben az operációs rendszer támogatja (jellemzően többfelhasználós rendszerekben), akkor beállítható, hogy adott állományra vonatkozóan adott felhasználó milyen jogokkal rendelkezik (például olvasás, módosítás, törlés, jogosultságok megváltoztatása stb.).

A **mappa** (folder) (vagy másnéven **könyvtár**, directory) egyrészt az operációs rendszer által használt adminisztratív célokat szolgáló állomány, amely az állományok logikai csoportosítását a fizikai elhelyezkedésük tárolásával teszi lehetővé, másrészt a (felhasználói) állomány elhelyezkedésének logikai struktúráját tükröző bejegyzés. A lehetséges tárolási szerkezetek közül az operációs rendszerek a **hierarchikus fa struktúrát** használják legelterjedtebben. Ez a struktúra egy körmentes, irányított gráffal írható le,

amelynek pontosan egy kiinduló csúcspontja (gyökere) van és minden más pontjába pontosan egy út vezet a gyökértől. Az egyes csúcsokból tetszés szerinti számú él vezet más csúcsokhoz. Azt a csúcsot, ahonnan egy él kiindul, **gyökérnek** (szülőnek), ahová vezet, **almappának** (**alkönyvtárnak**, gyereknek) nevezzük. A gyökérnek nincs szülője és minden gyereknek pontosan egy szülője van. A könyvtárak névhasználatára általában az állományokkal azonos szabályok vonatkoznak (kivéve a gyökérkönyvtárat, amelynek a neve – operációs rendszertől függően – általában \ vagy /).

A logikai tárolási rendszer legmagasabb szintjén a háttértár egészének vagy részének megkülönböztetését lehetővé tevő azonosítók állnak.

- Ilyen a **partíció**, amely a háttértár fizikailag összefüggő önálló része, amely az operációs rendszer számára önálló adattároló egységként kezelhető.
- És ilyen a **kötet** is, amely a háttértároló egyedi azonosítóval rendelkező területe. A kötetek azonosítására az angol ábécé 26 betűjét (ABCDEFGHIJKLMNOPQRSTUVWXYZ) használjuk.

6.5. Állománykezelés

A háttértáron való adattárolás alapegysége az előzőek alapján tehát az **adatállomány** (fájl), amelynek **két alaptípusa** létezik:

- A szövegfájl karakterek (számok, betűk) sorozata, amely esetleg még szövegformázó információt is tartalmazhat. Ez a fajta állomány szövegszerkesztő programmal olvasható.
- A bináris fájl csak a számítógép által értelmezhető információt tartalmaz (lehet például gépi kódú program, tömörített fájl, kép, zene stb.). Ha ezt szövegszerkesztővel próbálnánk meg olvasni, értelmetlen karakterhalmazt kapnánk.

Az **állományok (és mappák) neve két részből áll:**

- Az első rész tulajdonképpen bármi lehet, van azonban néhány karakter, amit fájlnevében nem lehet alkalmazni (például + = > jelek – ha mégis ilyen karaktert tartalmazó fájlnevet próbálunk megadni, az operációs rendszer figyelmeztet és nem fogadja el a fájlnevet).
- A második rész a fájl típusát határozza meg (kiterjesztésnek is nevezik). A kiterjesztés fájlnevhez adása általában az állomány létrehozásakor, mentésekor automatikus, illetve, ha mégsem, akkor a fájlnévtől ponttal elválasztva kell beírni (például: szöveg.txt – szöveges állomány neve).

Gyakori fájltypusok:

- Futtatható programok: com, exe (gépi kódú program), bat, cmd (szövegszerkesztővel szerkeszthető, utasításokat és programneveket tartalmazó parancsfájl); (Érdemes megjegyezni, hogy Unixban és Linuxban nem a fájl kiterjesztése, hanem az egyik jellemzője (attribútuma) határozza meg, hogy futtatható-e az adott állomány, vagy sem.)
- Irodai programok: docx, odt, txt, rtf (szövegfájl), ods, xlsx (táblázatkezelő munkafüzet), odp, pptx (bemutató), accdb, dbf, odb (adatbázis);

- Hang állományok: mid, mp3, ogg, wav;
- Kép állományok: bmp, jpg, gif, png;
- Videó állományok: avi, mp4, mpg, mov, wmv;
- Weblapok: htm, html;
- Tömörített állományok: 7z, cab, rar, zip.

A mappákon és fájlokon (objektumon) **végezhető műveletek** a következők:

- **Létrehozás:** a háttértáron új objektum létrehozása. Létrehozáskor kötelező nevet adni. A névnek abban a mappában, ahol az új objektumot létrehozzuk, egyedinek kell lennie.
- **Átnevezés:** Egy már létező objektumnak új nevet adhatunk. Az új névnek az adott mappában egyedinek kell lennie.
- **Megnyitás:** Ez az objektum tartalmának megjelenítését, vagy program esetén futtatását jelenti.
- **Másolás, áthelyezés:** Másolás esetén az eredeti objektum nem törlődik, arról másolat készül – beillesztés során – egy másik háttértáron, vagy ugyanazon a háttértáron egy másik mappában, vagy akár az eredeti mappába is vissza lehet másolni egy objektumot, de ez utóbbi esetben az egyedi elnevezésről gondoskodni kell. Áthelyezés esetén az eredeti tartalom – a művelet sikeres lebonyolításakor – törlődik.
- **Törlés:** Az objektum eltávolítása a háttértárról.
- **Visszaállítás:** Az objektum törlése – az operációs rendszer beállításaitól és a törlés módjától függően – sok esetben nem végleges törlést jelent. A rendszer biztosíthat egy ún. lomtárat (ami tulajdonképpen egy mappa a háttértáron), amely a törölt állományokat tárolja egy meghatározott ideig, vagy addig, amíg a lomtár meg nem telik. Innen a „törölt” állomány egy ideig még visszaállítható.
- **Tömörítés, kicsomagolás:** Az eredeti objektumot helytakarékosági megfontolásokból összetömöríthetjük a háttértáron. Fájltömörítésről beszélünk akkor, amikor az állományokat (vagy állományok egy csoportját) tömörítjük össze. Az összetömörített fájlt a használat előtt ki kell csomagolni. Meghajtó tömörítésről beszélünk, ha nem csak egyes állományokat tömörítünk a háttértáron, hanem a háttértár teljes tartalmát. Ebben az esetben – miután megadtuk, hogy a háttértár tömörítést alkalmazunk az adott tárhelyre – a tömörítés és a kicsomagolás automatikusan, a háttérben megtörténik, amikor a háttértárra állományt másolunk, vagy onnan átmásolunk, illetve megnyitunk valamit.
- **Tulajdonságok megtekintése, megváltoztatása:** Az objektumok tulajdonságait megtekinthetjük, egyes jellemzőket (például tulajdonos, rejtett, írásvédett stb.) megváltoztathatunk.
- **Keresés:** Nem szigorúan tartozik a mappa- és fájlműveletek közé, de ma már alapszolgáltatásnak számít, hogy különböző paraméterek (például név, típus, dátum) megadásával állományokat kereshetünk.

6.6. Ellenőrző kérdések

1. Ismertesse az operációs rendszer néhány definícióját! [63]
2. Az ISO hogyan definiálja az operációs rendszert? [64]
3. Technológiai, funkcionális, folyamatcentrikus és felhasználói szemléletű megközelítésben mi az operációs rendszer? [64]
4. Mik az operációs rendszer fő feladatai? [64]
5. Hol helyezkedik el az operációs rendszer a számítógépes rendszerben? Mi jellemzi az egyes szinteket? [65]
6. Mit jelent a „boot”-olás? [65]
7. Mit értünk kompatibilitás alatt? [66]
8. Mit értünk multiprogramozás alatt és milyen változatai léteznek? [66]
9. Milyen segédprogramokkal egészíthetik ki az operációs rendszereket? [66]
10. Melyek a legelterjedtebb operációs rendszerek? [66]
11. Mi a shell és a kernel feladata? [66]
12. Milyen szempontok szerint csoportosíthatók az operációs rendszerek? [67]
13. Hogyan, milyen módon folyik a kommunikáció a felhasználó és az operációs rendszer között? Milyen kapcsolattartási módokat ismer és milyen információt szolgáltat a rendszer? [68]
14. Mit nevezünk felhasználói felületnek? [68]
15. Hogyan használható a karakteres felhasználói felület (utasításkészlet, utasításszerkezet, utasítás végrehajtása, nyugtázás)? [69]
16. Hogyan használható a grafikus felhasználói felület (elemei, használata, vezérlési módok)? [70]
17. Mit értünk állományszervezés, állománykezelő rendszer és állomány alatt? [71]
18. Mi a fájlrendszer és milyen típusai lehetnek? [71]
19. A néven kívül milyen tulajdonságai lehetnek egy állománynak (fájlnak)? [71]
20. Mi a mappa (könyvtár) és hogyan épül fel a mappák struktúrája? [71]
21. Mi a partíció és a kötet? [72]
22. Melyik az adatállomány két alaptípusa és mi jellemzi ezeket? [72]
23. Mik az állományok két részének fontosabb jellemzői? [72]
24. Milyen gyakori fájltypusokkal találkozhatunk? [72]
25. Milyen mappákon és fájlkon végezhető műveleteket támogat az operációs rendszer? [73]

7. MULTIMÉDIA ISMERETEK

Elvárt tanulási eredmények

Tudás

- Átfogóan ismeri a multimédiát támogató számítógépek megjelenésének körülményeit
- Ismeri a multimédia fogalmát, értelmezéseit
- Érti az MPC specifikációk jelentőségét
- Felsorolja egy modern multimédia személyi számítógép fő alkotórészeit
- Felsorolja a multimédia személyi számítógép bemeneteit, megjelenítő és feldolgozó eszközeit, valamint kimeneteit
- Érti veszteséges és veszteség nélküli tömörítés célját és lényegét
- Érti a pixelgrafika és vektorgrafika közötti különbséget
- Felsorolja a képszerkesztőket és azok szolgáltatásait
- Felsorolja az elterjedtebb képállományokat és azok fő jellemzőit
- Átfogóan ismeri a mozgókép állományokat és az elterjedtebb formátumokat
- Felsorol videószerkesztőket, konvertálókat, médialejátszókat, azok fontosabb szolgáltatásait
- Felsorol mozgóképállományokat és azok fő jellemzőit
- Érti a mozgóképeket tömörítő eljárások lényegét
- Ismeri hanghullámok analóg-digitális és digitális-analóg konverziójának lényegét
- Ismerteti a hangtömörítésnél alkalmazott elveket
- Ismeri a hangállományok fő jellemzőit
- Felsorol hangszerkesztőket és azok szolgáltatásait
- Felsorol hangállományokat és azok fő jellemzőit

Képesség

- Megnyit álló- és mozgóképet, hang- és videóállományt
- Megnyit, lejátszik multimédia tartalmat

Attitűd

- Értékként tekint az információ többcsatornás közvetítésére
- Kreatívan alkalmazza a multimédia eszközeit saját szakterületén tanulási, illetve feladatmegoldási célra

Autonómia, felelősség

- Önállóan fogalmaz meg ajánlást álló- és mozgóképek, valamint hangok létrehozására, szerkesztésére, továbbítására és lejátszására alkalmas hardver- és szoftverkörnyezetre
- Az önállóan végzett munkájáért felelősséget vállal

7.1. A multimédia fogalma

Számítógépek kezdetben meglehetősen korlátozott adatmegjelenítési képességekkel rendelkeztek. A személyi számítógépek 1980-as évekbeli megjelenésekor kezdetben csak szöveges megjelenítésre voltak képesek és a „hanghatásokat” egy egyszerű, csipogásra alkalmas hangszóró biztosította (aminek leginkább hibajelzés volt a feladata). A felhasználói igények és a számítógépek teljesítményének növekedésével, valamint a perifériák fejlődésével lehetővé vált a grafikus megjelenítés – nagyjából az 1990-es évek elejétől –, először 2-3, majd 4, 16, 256 színárnyalat felhasználásával. Megjelentek a csipogásnál többre képes hangeszközök (hangkártyák) is, amelyek már alkalmasak voltak hangeffektus, vagy akár zene lejátszására elfogadható minőségben (sőt egyes hangkártyák hangrögzítésre is képessé váltak). A nagyfelbontású grafika és a hangeffektusok alkalmazása először a számítógépes játékokban és néhány speciális területen (például számítógépes grafika, tervezés, zeneszerkesztés) terjedt el. Az 1990-es években egy nagyfelbontású grafikával és hangképzéssel felruházott (személyi) számítógép még ritkaságszámba ment. A **teljesítmény rohamos növekedésével** és az **eszközök egyre olcsóbbá válásával** a grafika és a hang kezdett szerepet kapni még az irodai alkalmazásokban is.

Ma már természetesnek vesszük, hogy akár szövegszerkesztés közben (is) nagyfelbontású grafikus felületen dolgozunk, a szöveg közé képeket, ábrákat szerkeszthetünk be, sőt hangeffektusokat, zenét, vagy akár animációt és videót is beszúrhatunk. Eljutottunk oda, hogy a tucattermékeknek számító, nem túl nagy teljesítményű számítógépek képesek szövegszerű adatok feldolgozása mellett hang, álló és mozgókép kezelésére is. Az elmúlt évtizedek fejlődésének következtében az adatok szövegszerű kezelése mellett a **grafika** és a **hang** alkalmazása is **alapszolgáltatássá vált**.

A **multimédia** szó két latin eredetű részből áll. A multi jelentése: több; a médium (egyik) jelentése: **közvetítő**. Azt a csatornát, amely közvetíti az információt, médiumnak nevezzük. A médium többszáma média. Ebből következik, hogy multimédia = több + média = több közvetítő(k!). A gyakorlatban akkor beszélünk multimédiáról, ha az **információ többféle adatformátumban** jelenik meg különböző csatornákon keresztül és ezeket **együttesen**, egymással **összehangolva** használjuk. Fontos, hogy különböző formában közölt információnak egymással összefüggésben kell lennie! Az információelemeknek egymással szinkronban kell hatniuk, annak érdekében, hogy kellő hatást érjenek el, ne zavarják, hanem erősítsék egymást.

Multimédia alatt a különböző információhordozó objektumok kombinációjával létrehozott **audiovizuális terméket** érthetjük; a **fő alkalmazási területei** az információközlés, ismeretterjesztés, oktatás és szórakozás.

Feldolgozási rendszer, amely többféle csatornát használ (szöveget, hangot, képet, animációt, videót és mindezt interaktívan) a felhasználók tájékoztatására vagy szórakoztatására.

A multimédia olyan **kreatív alkotó közeg**, amely biztosítja egyének és csoportok számára a digitális formában rögzített információ (kép, grafika, mozgókép, hang, írott szöveg stb.) interaktív kezelését (rögzítést, tárolást és módosítást). **Célja** az információigények magas színvonalú kiszolgálása – a hatékonyság érdekében – minél több érzékszervre egyidejűleg ható ingerekkel.

„A **számítógépes multimédia alkalmazások** különböző információhordozók együttes, összehangolt megjelenítése számítógép segítségével.” (Debreceni, 2005, 9. oldal).

Vagy, ahogy a Microsoft fogalmaz: A multimédia hang, grafika, animáció és mozgókép kombinációját jelenti és a **hipermédia része**. A hiperszövegben a szöveg egyes részei kapcsolatban állnak egymással, amelyek között a felhasználó információkinyerés céljából tetszőlegesen mozoghat. Ha ezt a hiperszöveget tetszőlegesen (de a szövegben lévő információtartalomhoz kapcsolódóan) kombináljuk grafikával, hanggal és videóval és ezek is szerepet kapnak az információközvetítésben, akkor beszélünk hipermediáról.

7.2. A multimédia személyi számítógép fogalma, jellemzői

Az 1990-es évekig már eljutott oda a technológia, hogy képessé váltak a számítógépek a grafikus megjelenítésre és a (csipogásnál) jobb minőségű hang kibocsátására, de többféle és különböző jellemzőkkel bíró megoldás született, ami megnehezítette az eligazodást mind a felhasználók, mind a programozók számára.

A **Multimédia PC specifikációját** a Multimedia PC Marketing Council adta ki az 1990-es évek elején annak érdekében, hogy segítse a standardizált multimédia megoldások elterjedését a személyi számítógépek piacán. Később ezt további specifikációk követték, végül az MPC 3 szintjét határozták meg (4. táblázat).

MPC Level 1 (1991)	MPC Level 2 (1993)	MPC Level 3 (1996)
16 MHz-es 386SX CPU; 2 MB RAM; 30 MB-os merevlemez; 640×480 VGA 8 bit (256) színnel; 1x-es sebességű CD-ROM meghajtó, olvasásnál CPU leterheltség 40% alatt, keresési idő 1 mp alatt; Hangkártya kimenet 22 kHz, 8 bit, bemenet 11 kHz, 8-bit; Windows 3.0 multimédia kiterjesztéssel;	25 MHz-es 486SX CPU; 4 MB RAM; 160 MB-os merevlemez, 640×480 VGA, 16 bit (65 536) szín; 2x-es sebességű CD-ROM meghajtó olvasásnál CPU leterheltség 40% alatt 1x-es sebességnél, keresési idő 400 ms alatt; Hangkártya kimenet 44 kHz, 16 bit CD minőségű hang; Windows 3.0 multimédia kiterjesztéssel vagy Windows 3.1;	75 MHz-es Pentium CPU; 8 MB RAM; 540 MB-os merevlemez; Videó rendszer, ami képes 352×240 felbontású videó lejátszására 30 képkocka/mp sebességgel 16 bit színmélységben (65 536 szín) MPEG-1 lejátszás hardver- vagy szoftvertámogatással; 4x-es sebességű CD-ROM meghajtó olvasásnál CPU leterheltség 40% alatt, keresési idő 250 ms alatt; Hangkártya kimenet 44 kHz, 16 bit CD minőségű hang; Windows 3.11 vagy Windows 95;

4. táblázat. Multimédia PC specifikációk (1., 2. és 3. szint)

Forrás: MPC Standards (1997) alapján a szerző szerkesztése.

A 90-es évek végére a személyi számítógépek alapvető tartozékává vált az MPC specifikációban felsorolt hardver és szoftver, még akár irodai környezetben történő használat esetében is. Azok a PC-k, amelyeket kifejezetten videó- és hangfeldolgozásra, ill. ezek lejátszására, vagy játékokra használnak, jellemzően nagyobb teljesítményű processzorral, háttértárral, valamint grafika, videó és hang kezelésére alkalmas perifériákkal rendelkeznek. Ma minden olyan személyi számítógép **multimédia PC**-nek tekinthető, amely rendelkezik a következő **paraméterekkel**:

- Egy vagy több nagy teljesítményű processzort tartalmaz;
- Sok (több gigabájtnyi) operatív memóriával rendelkezik;

- Nagyméretű és gyors háttértár (merevlemez-meghajtó vagy szilárdtest-meghajtó) van beépítve;
- Egy vagy több nagy teljesítményű, hardveres 3d és videógyorsító funkcióval rendelkező grafikus kártyával rendelkezik, ami azzal kompatibilis képmegjelenítőre (monitorra, projektorra) csatlakozik;
- Hangkártyát használ (cd vagy hifi minőségű többcsatornás hang be- és kimenetekkel, esetleg szintetizátorkapcsolódási lehetőséggel), amire fejhallgató vagy hangszórók csatlakoznak;
- Optikai háttértár (DVD olvasó/író, ill. blu-ray olvasó/író) van beépítve; (Megjegyzés: elavulófélben lévő technológia, a legkorszerűbb számítógépek sokszor már nem tartalmaznak optikai háttértárat.)
- Operációs rendszere támogatja a multimédiát, vagyis egységes programozási felületet biztosít, eljárásokat tartalmaz a multimédia eszközök programozásának megkönnyítéséhez.

7.3. A multimédia személyi számítógép bemenete, kimenete és szoftverei

Inputként, vagyis a multimédia forrásaként a következő eszközök jöhetnek szóba:

- Szöveg: szöveges állomány, lapolvasó, optikai karakterfelismerő szoftver;
- Hangbemenet: mikrofon, rádió, magnetofon, CD lejátszó, szintetizátor stb.
- Képbemenet: lapolvasó, digitális fényképezőgép stb.;
- Videó bemenet: analóg és digitális videólejátszó, analóg és digitális videokamera, webkamera, analóg és digitális televízióadás;
- A fentiek valamelyiket tartalmazó bemenetek (például háttértár, internet);

Megjelenítő és feldolgozó szoftver és hardver a multimédia állományok készítéséhez:

- Szövegszerkesztő: szöveges állományok készítéséhez, szerkesztéséhez;
- Képszerkesztő és képmegjelenítő: képek szerkesztéséhez, megjelenítéséhez, archiválásához;
- Hangszerkesztő, dalszerkesztő és lejátszó: hanghatások, zenék, szintetizátorzenék szerkesztéséhez és visszajátszásához;
- Videószerkesztő és lejátszó: animációk és videók készítéséhez, szerkesztéséhez és lejátszásához;
- A fentieket együttesen kezelő rendszerek: szerzői rendszerek, videóvágók és feliratozók stb.

Kimenetek a multimédia tartalom megjelenítésére:

- Szöveg- és képmegjelenítők: monitor, projektor, nyomtató, rajzgép, digitális nyomdagép, háttértár, internet stb.;
- Audió eszközök: hangszóró, hangszóró, fejhallgató, háttértár, internet stb.;
- Mozgókép eszközök: képernyő, háttértár, internet stb.;

- Egyéb, a fentieket együttesen kezelő lehetőségek: weblapok, interaktív multimédia alkalmazások stb.;

7.4. Veszteséges és veszteség nélküli tömörítés

A **tömörítést** elsősorban azért használjuk, hogy a tárolandó adat kisebb helyen férjen el. Az adattömörítésnek az a célja, hogy a rendelkezésre álló háttértárat minél jobban ki tudjuk használni. A tömörítést úgy érhetjük el, hogy a gyakran ismétlődő jelsorozatokat adattömörítő programokkal rövidebb kódokra cseréljük. Attól függően, hogy a tömörített jelsorozatból visszaállítható-e teljes egészében az eredeti jelsorozat, beszélhetünk **adatvesztéses vagy adatvesztés nélküli tömörítésről**:

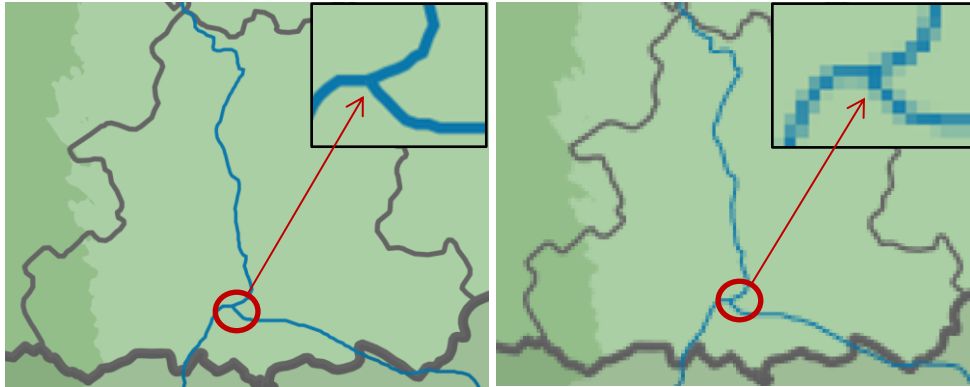
- **Adatvesztés nélküli tömörítés:** Ezek a programok az eredeti állományt betömörítve, majd újra kibontva olyan állományt szolgáltatnak, amelyek bitről bitre megegyeznek az eredeti állománnyal. Így tömörítik a programokat, dokumentumokat, de néha előfordul hang- és képtömörítés esetében is. Ilyen tömörített állományok a 7z, cab, rar és a zip.
- **Veszteséges tömörítési algoritmusok:** A képek és hangok tárolásánál engedhető meg a használatuk. Az itt alkalmazott algoritmusoknál az eredeti állományt nem lehet létrehozni a tömörített állományból, de nincs is rá szükség, hiszen a szemünk és a fülünk becsapható. Elég csak arra gondolni, hogy a gyorsan változó állóképek sorozatát a szem, illetve az agy folyamatos mozgásnak érzékeli. A veszteséges algoritmusok többnyire olyan információkat hagynak el a képből, vagy hangból, amelyek nélkül az emberi szem, vagy fül az eredetivel csaknem azonos, megkülönböztethetetlen jelként fogja fel a tömörített képet vagy hangot. Így tömörített állományok az mp3, jpg, mpg, mp4 stb. (szinte az összes kép és videó állomány).

7.5. Vektorgrafika, pixelgrafika

A **számítógépes grafika** képek, különféle rajzok számítógépen történő előállításával, valamint ezek számítógépes adathordozón való tárolásával foglalkozik. A grafika két csoportba osztható: a **pixelgrafikus** (más néven bittérképes, rasztergrafikus) és a **vektorgrafikus** képek (15. ábra).

- A pixelgrafikus rajzolóprogramok a képeket mátrixszerűen elrendezett képpontokból, pixelekből, építik fel. A pixelgrafikus képállomány egy kép tartalmát egy négyzetrácson elhelyezkedő színes képpontok összességéként, ún. pixelekkel írja le. A pixelgrafikus kép felbontásával és a színek számával (színmélység) a képfájl mérete is növekszik, hiszen több képpont adatait kell tárolni.
- A vektorgrafikus rajzolóprogramok a képek felépítésére egyszerű alakzatokat (téglalap, ellipszis, sokszög stb.) és ún. Bézier-görbéket (csomópontokkal, a csomópontok közt húzott görbékkel és érintőszakaszokkal felépített görbéket) használnak. A képek tehát át nem képpontokból állnak; tetszőlegesen nagyíthatók és kicsinyíthetők, a végeredmény minősége csak a képmegjelenítő eszköztől függ. Mivel a vektorgrafikus rajzolóprogramok a képeket csomópontok segítségével építik fel, a képfájlok méretét a csomópontok és a görbék száma határozza meg. Minél több csomópont szükséges a kép leírásához, azaz minél több görbéből áll a kép, annál nagyobb a vektoros állomány mérete. A kép nagyításával, illetve

kicsinyítésével nem változik a csomópontok száma, így nem változik az állomány mérete sem.



15. ábra. Vektorgrafikus kép (balra) és pixelgrafikus kép (jobbra).

Forrás: Google Maps (2018) alapján a szerző szerkesztése.

A képszerkesztő programok fő szolgáltatásai:

- Alapvető manipulációk: forgatás, vágás, felbontásváltoztatás, képek kombinálása, rétegek alkalmazása, retusálás, vörös szem eltávolítás és egyéb képkorrekciók.
- Színkezelés, színekorrekció: a kép színének javítása, színcsere, átszínezés, színtónus módosítása, a kép színmélységének (színek számának) módosítása.
- Albumok készítése: A képek rendezése különböző szempontok (például téma, méret, dátum) szerint. Lehetőség van a képekhez leíró információkat hozzárendelni. Az elkészült albumok diasorként levetíthetők.
- Animáció: A képekhez animáció hozzárendelése.
- Konvertálás: Képformátumok közötti átalakítás és ezzel együtt a tömörítés mértékének meghatározása.

Gyakori pixelgrafikus állókép formátumok:

- BMP (Bitmap Image File): Windowsban használt fájlformátum 24 bites digitalizált képek veszteséges tömörítéssel vagy tömörítés nélküli tárolására.
- GIF (Graphics Interchange Format): 8 bites pixelgrafikus képek tárolására alkalmas formátum, veszteségnélküli tömörítést használ.
- JPG, JPEG (Joint Photographic Experts Group): Az internet egyik elterjedt képformátuma. A 24 bites pixelgrafikus képet igen nagy hatékonysággal, veszteségesen tömöríti, az eredeti méretnél akár 5-100-szor kisebb is lehet a JPG állomány.
- PNG (Portable Network Graphics): 32 vagy 24 bit színmélységű kép veszteségmentes tömörítését teszi lehetővé, elsősorban a GIF leváltására készült (a licenszdíj elkerülése érdekében).
- TIFF (Tagged Image Format): Nagyon sok szerkesztőprogram és kiadványszerkesztő által támogatott képformátum, lehetővé teszi a 32, vagy 64 bit színmélységű képek veszteséges tömörítéssel vagy tömörítés nélkül történő tárolását.

A pixelgrafikus állományokat **szerkesztőprogramok** közé tartoznak: Adobe PhotoShop, Corel PhotoPaint, Irfan View, GIMP (GNU Image Manipulation Program), Microsoft Paint.

Elterjedtebb **vektorgrafikus állókép formátumok**:

- CDR (CorelDraw): A CorelDraw Graphics Suite kanadai grafikus programcsomagot fejlesztő cég saját vektorgrafikus formátuma.
- DWG (Drawing): Két- vagy háromdimenziós tervrajzok tárolására szolgál azok metaadataival együtt. Elsősorban tervezőprogramokban használt formátum.
- DXF (Drawing Interchange Format vagy Drawing Exchange Format): Az AutoCAD által kifejlesztett formátum, a különböző tervező programok közötti adatcsere megvalósítására.
- EPS (Encapsulated PostScript): Az Adobe által kifejlesztett formátum. Eredetileg olyan szöveges dokumentumokat írt le, amelyek beágyazva tartalmaztak képet. Mivel ma már sok vektorgrafikus program ismeri, a fájlformátumok közötti átjárás egyik eszköze.
- PDF (Portable Document Format): Az Adobe által kifejlesztett formátum. Létrehozásával egy olyan dokumentumformátum kialakítása volt a cél, amely szoftvertől, hardvertől és operációs rendszertől független (tehát gyakorlatilag bármilyen eszközön egyformán megjeleníthető, ha van az adott eszközön PDF megjelenítő szoftver). A PDF tartalmazhat szöveget, képet és egyéb objektumokat is.
- WMF (Windows Metafile): A Windows operációs rendszerhez kifejlesztett formátum. Érdekessége, hogy vektor- és pixelgrafikus komponenseket is tartalmazhat.

Vektorgrafikus állományokat **szerkesztő programok** közé tartoznak: Adobe Acrobat, AutoDesk AutoCad, Adobe Illustrator, CorelDraw, Inkscape, LibreOffice Draw, Google SketchUp, Xara.

7.6. Mozgókép

Mozgóképek esetén nagymennyiségű képpont kezelésére van szükség, amelyhez megfelelő teljesítményű processzor, grafikus kártya és háttértár szükséges. Abban az esetben, ha például az 1920 × 1080-as (full HD) felbontást vesszük alapul 8 bites színmélységgel (2⁸ színárnyalat RGB vörös/zöld/kék színcsatornánként, ami körülbelül 16,7 millió színárnyalat), az 25 képkocka/másodperc sebesség mellett nagyjából 155,5 MB-nyi adat folyamatos mozgatását és tárolását jelenti egyetlen másodperc alatt (1 percnyi videó pedig kicsivel több mint 9 GB). (Ehhez jön még a hangcsatorna, de az elenyésző ehhez az adatmennyiséghez képest.) Ez egy mai átlagos számítógép processzora, buszrendszere és háttértára számára is megterhelő, ráadásul igen gyorsan megtelhet a tárolásra használt háttértár. A szükséges adatátviteli sebességet – rögzítéskor a folyamatos írást, lejátszáskor a folyamatos olvasást – minden körülmények között biztosítani kell, ellenkező esetben képkockák fognak kiesni, szakadozni fog lejátszáskor a videó, ami akár élvezhetetlenné is teheti azt.

A fentiekből következően mozgóképeknél van a legnagyobb jelentősége a különböző **tömörítési eljárásoknak**. A videó tömöríthető azonnal, a rögzítés közben, valós időben

(real-time tömörítés). Erre használhatók bizonyos processzorok, grafikus kártyák, vagy alkalmazható célhardver; kizárólag szoftveres megoldás a nagy teljesítményigény miatt itt nem jöhet szóba. A már rögzített videó utólag, nem valós időben is átalakítható más formátumra és eközben újratömöríthető, ekkor alkalmazhatók kizárólag csak tömörítő szoftverek is (de ez hardvertámogatás nélkül jelentős időt vesz igénybe).

Alkalmazható veszteséges és veszteség nélküli tömörítési eljárás is. Mozgóképeknél a tömörítés **két alapvető eljárást** követhet:

- A videó minden képkockáját tömörítve tároljuk (általában JPEG tömörítéssel: Motion JPEG, MJPEG)
- Bizonyos időközönként, vagy képkockánként a teljes képet tároljuk, ezek lesznek a kulcskockák. A kulcskockák között pedig csak a változásokat, eltéréseket tároljuk (ide tartozik a videótömörítési eljárások nagy része).

A **lejátszáshoz és szerkesztéshez** a videót **valós időben kell kicsomagolni** (ez automatikusan történik a videó megnyitásakor); a művelet sokkal kevésbé erőforrás- és időigényes mint a tömörítés.

A **videószerkesztő és konvertáló** programok fő **szolgáltatásai** alapvetően megegyeznek a kép szerkesztők szolgáltatásaival, azzal a kiegészítéssel, hogy általában

- egyszerre több videó és hangsáv kezelésére képesek,
- ezek közötti áttűnéseket, animációkat kezelnek,
- feliratozást és feliratokhoz kapcsolódó effektusokat is támogatnak.

A **médialejátszók** fő **szolgáltatásai**: (videólejátszó helyett helyesebb a médialejátszó elnevezés, mivel videót és hangot is lejátszanak)

- videó és hang lejátszása különböző sebességekkel,
- ugrás a videó egy meghatározott pontjára,
- hangsávok és feliratok közötti váltás,
- valós idejű képmanipulációk (vágás, nagyítás, forgatás stb.),
- albumok készítése, leíró információk hozzáadása, vagy letöltése az internetről.

Elterjedtebb **mozgóképfarmátumok**:

- AVI (Audio Video Interleave): Ez a Windowsban használt fájlformátum tulajdonképpen egy tároló, egy keret, amely több és többféle módon tárolt/tömörített videó és audió sávnak ad helyet.
- MJPEG (Motion JPEG): Veszteséges tömörítési eljárást alkalmazó formátum, amelyben az egyes képkockák külön-külön JPEG tömörítési eljárással vannak tömörítve és tárolva.
- MKV (Matroska Video): Az AVI-hoz hasonlóan egy tároló, egy keret, amely több és többféle módon tárolt/tömörített videó, hang, valamint felirat sávnak ad helyet.
- MPG, MPEG2, VOB (Moving Picture Experts Group Phase 1/Phase 2, Video Object): Videóállományok veszteséges tömörítéssel történő tárolására szolgáló formátumok, mely lehet MPEG1 (a videó cd-k esetében használt tömörítés,

felbontás általában 352×288 képpont), ill. MPEG2 (felbontás általában 720×576 képpont). A DVD lemezeken használt tömörítés MPEG2, itt a fájlok egyéb információt is tartalmaznak, az állományok kiterjesztése VOB. Az összes képkocka tárolása helyett bizonyos időközönként kulcskockákat tartalmaz, a köztes képkockák teljes eltárolása helyett csak a kulcskockák közötti különbségeket tárolja.

- MP4 (MPEG-4): Nagyfelbontású videó tárolására szolgáló, MPEG-4 veszteséges tömörítést alkalmazó formátum. Leggyakrabban blu ray lemezeken tárolt filmeknél (1920×1080) találkozhatunk vele.

Az ismertebb **videószerkesztő és konvertáló programok** közé tartoznak: Adobe Premiere, Avidemux, Blender, FFmpeg, Fmake, HandBrake, Magix Video Edit, VirtualDub.

Néhány elterjedt **médialejátszó**: Media Player Classic Home Cinema, PowerDVD, VLC.

7.7. Hangkezelés

A **hanghullám digitalizálása** során a digitális hangrögzítő eszköz (hangkártya) bizonyos időközönként mintát vesz az analóg hanghullámból (**analóg-digitális konverzió** történik). Minél nagyobb gyakorisággal történik a mintavétel (Hz) és minél nagyobb részletességgel történik egy jel leírása (bit), annál jobban fog hasonlítani a digitalizált hangminta az eredeti hangra. A megfelelő mintához a Nyquist-Shannon mintavételi elvnek kell teljesülnie: a mintavételezési frekvenciának nagyobbnak kell lennie, mint az analóg jel sávszélessége kétszeresének. Ez az oka például annak, hogy bár az egészséges fiatal emberi fül a 20 Hz – 20 000 Hz közötti frekvenciát hallja, a CD lemezen 44 100 Hz-en digitalizált hanghullámot tárolunk. Egy másodpercnyi CD minőségű hang tömörítés nélküli rögzítéséhez $44\,100\text{ Hz} \times 16\text{ bit}$ ($2^{16} = 65\,536$ jelszint, hangmagasság) $\times 2$ hangcsatorna = 1 411 200 bit = 176 400 byte = 176,4 kilobyte szükséges, ami a mai, korszerű számítógépek számára nem okoz különösebb problémát.

A digitalizált, digitális formátumban tárolt hangot hangkártya, ill. a kártyára csatlakoztatott fejhallgatón, hangszórón keresztül **játszhatjuk vissza a digitális-analóg konverzió** után.

Egyes hangkártyák képesek szintetizátorhoz kapcsolódni ún. MIDI (Musical Instrument Digital Interface) csatlakozón keresztül és a csatlakoztatott eszközt vezérelni, vagy **szintetizátorként** működni.

Gyakori mintavételezések, hangállományok minőségi adatai:

- Telefonvonal minőség: 11 kHz, 4 bit vagy 8 bit, monó;
- Szalagos magnó minőség: 22,05 kHz, 8 bit, monó (1 hangcsatorna) vagy sztereó (2 hangcsatorna);
- FM minőség: 22,05 kHz, 16 bit, sztereó (2 hangcsatorna);
- CD minőség: 44,1kHz, 16 bit, sztereó (2 hangcsatorna);
- Hifi minőség: 48kHz vagy 96kHz, 16 vagy 24 bit, sztereó (2 vagy több hangcsatorna);

A **hangtömörítésnél** alkalmazott **elvek** a következők:

- amit az emberi fül nem hall meg, illetve az átlagos lejátszó eszköz nem képes visszajátszani, azt felesleges tárolni (például a 16 ezer Hz feletti hanghullámokat);

- az emberi agy nem használ fel minden, a fül által hallott információt, így azt nem is kell tárolni (bár, hogy mikor mit „hall” meg az agy a fül által hallottakból, nagyon szubjektív és függ például az érzelmi állapottól);
- a hanghullámok bizonyos alakzatok szerint növekednek vagy csökkennek és a maximumról a minimumra nem egyik pillanatról a másikra térnek át;
- a sztereó hang két csatornája nem sokban különbözik egymástól, így elegendő a két csatorna eltéréseit tárolni;
- egymás után ismétlődő hangok esetében elegendő a hangot egyszer és azután csak az ismétlődések számát tárolni.

A hangszerkesztők fő szolgáltatásai:

- Hang rögzítése analóg vagy digitális forrásból (egyszerre akár több forrásból is);
- Több hangcsatorna kezelése egyszerre;
- Alapvető szerkesztési funkciók (másolás, törlés, kivágás, többszörözés, megfordítás stb.);
- Hangminőséggel kapcsolatos műveletek (zajszűrés, mintavételezési frekvencia változtatása stb.);
- Effektusok hozzáadása (visszhang, sebesség változtatása, hangmagasság változtatása, halkítás és erősítés stb.);
- Elemzés (a hanghullám elemzése, kimutatás készítése különböző szempontok szerint);
- Konvertálás hang- és fájlformátumok között és ezzel együtt tömörítés;
- A megnyitott, szerkesztett hanganyag lejátszása és tárolása;

Néhány ismertebb **hangszerkesztő**: Audacity, Creative Wave Studio, Nero Wave Editor, Sound Forge, WaveLab Pro.

A szintetizátoros dalszerkesztők fő szolgáltatásai:

- MIDI eszközök, szintetizátorok vezérlése;
- Általában 128 különböző típusú hangszer, illetve hangzás kezelése és akár 16 hangszer egyidejű megszólaltatása;
- Effektusok hozzáadása: visszhang, késleltetés stb.
- Felvétel szintetizátorról (akár kottába);
- A felvett zene szerkesztése és visszajátszása, továbbítása a MIDI eszközre;
- Mentés a számítógépre;

Ismertebb **dalszerkesztők** közé tartozik a Cakewalk és a Cubase.

Néhány gyakrabban előforduló **hangállomány kiterjesztés**:

- WAV (RIFF WAVE, Resource Interchange Format Wave): A Windows audióállomány egy fajtája, általában tömörítés nélkül tárolja a hangot (de támogatja a tömörített tárolást is); akár kettőnél több hangcsatorna is tárolható.

- MID (Musical Instrument Digital Interface): MIDI szabványnak megfelelő – szintetizátorok vezérlésére alkalmas – adatok tárolására fejlesztették ki; csak a hangmagasságot, ritmust és a lejátszandó hangszert tartalmazza, a hang megszólaltatása a MIDI hangeszköz feladata.
- MP3 (MPEG 1 vagy MPEG 2 Layer 3): Veszteséges tömörítési eljárás alapján, jellemzően zene tárolására használt formátum. A tömörített állomány az eredetinek akár tizede is lehet, a hang minőségromlása (átlagos minőségű lejátszó eszközökön) nem, vagy alig hallható.
- WMA (Windows Media Audio): A Microsoft zenei formátuma, veszteséges, vagy veszteség nélküli tömörítési algoritmust használ, tartalmaz beépített másolásvédelmi lehetőséget is.
- OGG (Ogg): Veszteséges tömörítési eljárás alapján, jellemzően zene tárolására használt formátum, azonban az MP3-mal ellentétben nem kell jogdíjat fizetni a használatáért.
- AC3 (Dolby Digital): A Dolby Laboratories által kifejlesztett veszteséges tömörítést használó formátum, akár 6 hangcsatorna tárolására.

7.8. Ellenőrző kérdések

1. Mi segítette elő a multimédiát támogató számítógépek megjelenését? [76]
2. Mit jelent a multimédia, milyen különböző értelmezései vannak? [76]
3. Mi a hipermédia és hogyan kapcsolódik a multimédiához? [77]
4. Mi az MPC specifikációk jelentősége? [77]
5. Milyen paraméterekkel rendelkezik egy mai multimédia személyi számítógép? [77]
6. Milyen bemenetei lehetnek a multimédia személyi számítógépnek? [78]
7. Milyen megjelenítő és feldolgozó eszközök állnak rendelkezésre multimédia állományok készítéséhez? [78]
8. Milyen kimenetei lehetnek a multimédia személyi számítógépnek? [78]
9. Mi a tömörítés célja és mi a különbség az adatvesztéses és adatvesztés nélküli tömörítés között? [79]
10. Milyen két csoportba oszthatók a képek és mik ezeknek a csoportoknak a jellemzői? [79]
11. Milyen szolgáltatásokat nyújtanak a képszerkesztő programok? [80]
12. Melyek az elterjedtebb pixelgrafikus állókép formátumok és milyen pixelgrafikus képszerkesztőket ismer? [80]
13. Melyek az elterjedtebb vektorgrafikus állókép formátumok és milyen vektorgrafikus szerkesztőprogramokat ismer? [81]
14. Mi okoz nehézséget a digitális mozgóképek esetében? [81]
15. Miért van szükség tömörítésre mozgóképeknél és milyen alapvető eljárások léteznek erre? [81]
16. Mik a videószerkesztő és konvertáló programok fő szolgáltatásai? [82]
17. Mik a médialejátszók fő szolgáltatásai? [82]
18. Milyen elterjedtebb mozgókép formátumokat ismer és mik ezek fő jellemzői? [82]
19. Milyen elterjedtebb videószerkesztő, konvertáló és médialejátszó programokat ismer? [83]
20. Mit jelent a hanghullám digitalizálása, az analóg-digitális, illetve a digitális-analóg konverzió? [83]
21. Sorolja fel a gyakori hang mintavételezések fő adatait! [83]
22. Melyek a hangtömörítésnél alkalmazott fő elvek? [83]
23. Milyen szolgáltatásokat nyújtanak a hangszerkesztők és milyen hangszerkesztőket ismer? [84]
24. Milyen szolgáltatásokat nyújtanak a szintetizátoros dalszerkesztők és milyen dalszerkesztőket ismer? [84]
25. Melyek a gyakran előforduló hangállományok és mi jellemző rájuk? [84]

8. IRODAI ALKALMAZÁSOK

Elvárt tanulási eredmények

Tudás

- Érti az alkalmazás, irodai alkalmazás és alkalmazói programcsomag fogalmakat
- Tudja, hogy honnan ered a WYSIWYG betűszó
- Felsorolja a szövegszerkesztők fő és kiegészítő szolgáltatásait
- Felsorolja a szövegszerkesztők típusait és érti a közöttük lévő különbségeket
- Ismeri a szöveg tárolásának módjait
- Felsorol különböző típusú szövegszerkesztőket
- Felsorolja a bemutatókészítés céljait
- Ismeri a bemutatókészítő programok fő szolgáltatásait
- Ismeri a táblázatkezelő alapvető szolgáltatásait
- Felsorol táblázatkezelő programokat
- Ismeri a munkafüzet részeit és a cellákban tárolható adattípusokat
- Elmagyarázza a szövegszerű és adatszerű tárolás közötti különbséget
- Elmagyarázza az operációs és az analitikus adatbázis közötti különbséget
- Átfogóan ismer több adatbázis definíciót
- Ismerteti az adatbázis-kezelő rendszer fő feladatait
- Felsorol néhány ismertebb adatbázis-kezelő rendszert

Képesség

- Létrehoz és módosít formázatlan és formázott szöveget
- Létrehoz és módosít adatokat tartalmazó táblázatot

Attitűd

- Elfogadja az irodai programcsomagok alkalmazásának jelentőségét az irodai munkavégzés során és saját szakterületén is
- Törekszik az irodai rendszerek valamennyi szolgáltatásának megismerésére, és a szolgáltatások használatának beépítésére a mindennapi munkájába
- Rugalmasan, egymással összefüggésben használja az irodai alkalmazásokat figyelembe véve azok szolgáltatásait

Autonómia, felelősség

- Alkotó módon állít elő információt az irodai programcsomagok segítségével
- Önállóan javaslatokat fogalmaz meg irodai rendszerekre konkrét feladatokhoz kapcsolódóan és ezért felelősséget vállal

8.1. Az irodai alkalmazások fogalma, szolgáltatásai

Az **alkalmazás**, alkalmazói szoftver, vagy felhasználói program a végfelhasználó számára valamilyen feladat végrehajtásának elősegítésére készített számítógépprogram. A kifejezés a felhasználó eszközeit megkülönbözteti a gépet működtető operációs rendszertől és a programfejlesztés eszközéül szolgáló programozási nyelvektől. Az **irodai alkalmazások** (office rendszerek) olyan programok, amelyek kifejezetten irodákban felmerülő feladatok megoldásában nyújtanak hatékony segítséget. Az **alkalmazói programcsomag** a program(ok) (alkalmazások) mellett tartalmazza a dokumentációk összességét is, amelyet egy csomagban árulnak.

Az **irodai alkalmazások (alkalmazáscsomagok)** tartalmaznak legalább szövegszerkesztőt, táblázatkezelőt és bemutató készítőt. Számos programcsomagban találunk ezek mellett adatbázis-kezelőt, hagyományos és elektronikus levelezést lebonyolító levelező rendszert, titkársági rendszert (ez utóbbi a levelező rendszer kiegészítve határidőnaplóval, partneryilvántartással), és egyéb más alkalmazást is; az alkalmazások száma és a kínált szolgáltatások köre egyre bővül. Irodai alkalmazáscsomagok: Apple iWork, Microsoft Office, a LibreOffice, Google G Suite stb.

A szöveg és grafika manipulálására használható irodai alkalmazások mindegyikére jellemző a grafikus felhasználói felület és a **WYSIWYG** (What You See Is What You Get: azt kapod, amit látsz) rendszerű működés. Ez utóbbi azt jelenti, hogy a képernyőn szerkesztés közben ugyanazt látjuk, ami a nyomtatáskor a papíron, vagy lejátszáskor a képernyőn megjelenik.

8.2. Szövegszerkesztés

A **szövegszerkesztő programok lehetővé teszik**: a szöveg

- begépelését,
- formázását,
- tördelését,
- tárolását (exportálását) és megnyitását (importálását), valamint
- megjelenítését és
- nyomtatását.

E programok egyéb, **kiegészítő szolgáltatásokat is nyújthatnak**, például keresést és cserét a szövegben, helyesírás ellenőrzést végezhetnek, támogathatják különböző objektumok beszúrását (képek, táblázatok, diagramok, hangeffektusok, videók stb.), az automatikus oldalszámozást, ábra-, táblázat- és fejezetszámozást, tartalomjegyzék készítést, a formázás megkönnyítése érdekében formázó stílusok készítését és alkalmazását stb.

Az **elkészült szöveg** vagy **papíron** jelenik meg nyomtatáskor, vagy valamilyen szöveges állományban tárolódik, amelyet a mentés formátumát ismerő szövegszerkesztővel, vagy egyéb – szöveg megjelenítésére alkalmas – programmal lehet megjeleníteni a **képernyőn**.

A szövegszerkesztők között találunk olyan alkalmazásokat, amelyek kizárólag **formázás nélküli szöveges állományok előállítására** szolgálnak. Ezeket az ún. **editorokat** (egyszerű szövegszerkesztőket) elsősorban programírás során használjuk az emberi nyelvhez közel álló, magas szintű programozási nyelven írt forrásnyelvi program begépeléséhez, amelyet a programírás következő fázisában lefordítunk a gép számára érthető gépi kódra. Ilyenkor

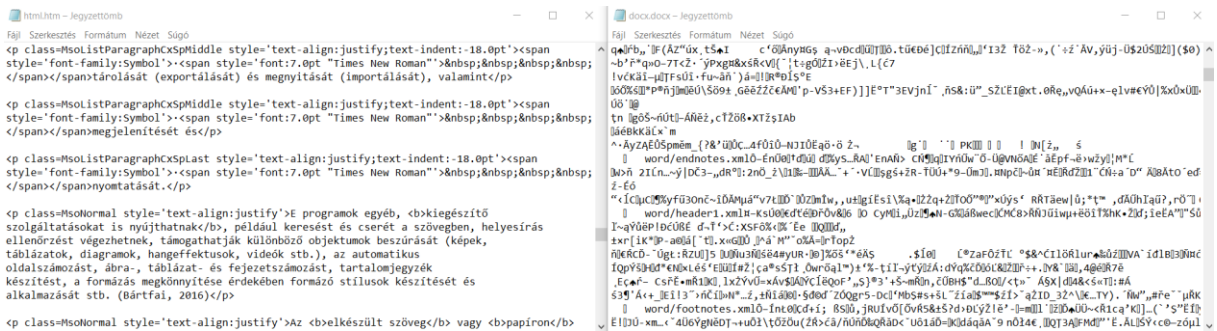
az utasítások és a hozzájuk tartozó egyéb programnyelvi elemek fontosak, nem az, hogy hogyan néz ki a szöveg. (Egyes editorok az adott programozási nyelv különböző elemeit képesek felismerni és azokat automatikusan eltérően formázni az áttekinthetőség kedvéért, de ezek a formázások nem kerülnek mentésre.) A szerkesztés végeredménye tehát egy formázás nélküli szöveg típusú állomány (általában TXT kiterjesztéssel, de erre vonatkozóan nincs kötelező szabály), amely csak alfanumerikus karaktereket (számokat, betűket, írásjeleket), és ezek mellett esetleg sor- és oldaltörés jeleket tartalmaznak.

A szövegszerkesztők **másik nagy csoportja formázott szöveget állít elő**. Itt találjuk azokat az alkalmazásokat, amelyeket „**szövegszerkesztőnek**” szoktunk nevezni, de ebbe a körbe sorolhatók a **kiadványszerkesztők**, a **weblapszerkesztők** és a **prezentáció készítő** programok is. Mindegyik tartalmazza a fejezet elején felsorolt szolgáltatásokat. Amiben **eltérnek egymástól**:

- a felkínált szolgáltatások közül melyekre helyezik a hangsúlyt, mik azok, amiknek a megvalósítását leginkább támogatják. Például szövegszerkesztővel több tíz, vagy akár több száz oldalas dokumentumot szerkeszthetünk, automatikusan számozhatjuk a fejezeteket és az oldalakat, majd ez alapján tartalomjegyzéket is automatikusan készíthetünk, míg egy kiadványszerkesztő elsősorban nagyobb posztterek, vagy néhány oldalas, nagyfelbontású grafikát tartalmazó szórólapok készítését támogatja hatékonyan.
- a kimenet típusa és sajátosságai, hiszen egy szövegszerkesztővel és kiadványszerkesztővel készült dokumentumot általában kinyomtatásra, míg a webszerkesztővel, vagy prezentáció készítővel készült munkát elsősorban képernyőn történő megjelenítésre szánjuk.

E második csoportba tartozó alkalmazások tehát formázott szöveget készítenek és mentenek. A szöveg az **alfanumerikus karaktereken kívül formázó jeleket** (például a betűtípusok, a félkövér és dőlt betűk, a színek jelölésére) és **egyéb információkat** (például szerző, lapméret, karakterkódolás, szöveg nyelve) is tartalmaz. A formázott szövegfájlban **a tárolás történhet**

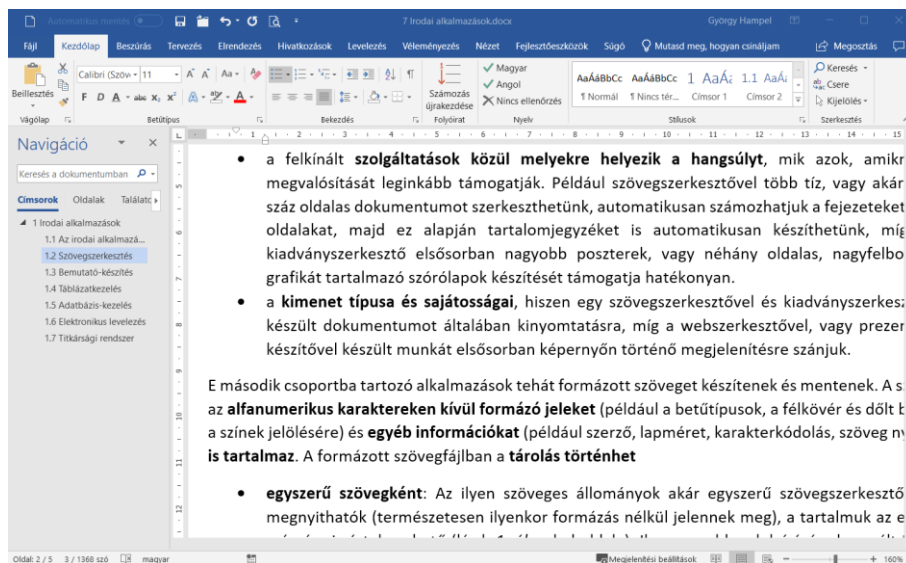
- egyszerű szöveggént: Az ilyen szöveges állományok akár egyszerű szövegszerkesztővel is megnyithatók (természetesen ilyenkor formázás nélkül jelennek meg), a tartalmuk az ember számára is értelmezhető (lásd: 16. ábra bal oldala). Ilyen a weblapok leírására használt HTML (HyperText Markup Language) és az RTF (Rich Text Format) szövegfájl.
- bináris formában: Itt minden binárisan van kódolva, az ember számára nem értelmezhető módon, a helyes értelmezéshez és megjelenítéshez az adott fájlformátumot ismerő program szükséges (lásd: 16. ábra jobb oldala). Ilyen állományok közé tartozik a DOCX (Microsoft Word által használt dokumentum formátum – ez a fájl valójában tömörített állomány, a tartalma tömörítő programmal kicsomagolható) vagy a PUB (Microsoft Publisher kiadványszerkesztő formátuma – szintén tömörített állomány).



16. ábra. Egyszerű szöveges tárolás (ábra bal oldala) és bináris tárolás (ábra jobb oldala).

Számos **szövegszerkesztésre alkalmas program** létezik, így:

- Egyszerű szövegszerkesztő: Windows Jegyzetömb, Notepad++;
- Szövegszerkesztő: Microsoft Word (17. ábra), OpenOffice Write, Google Docs;
- Kiadványszerkesztő: Adobe InDesign, Microsoft Publisher, Scribus;
- Weblapszerkesztő: Aptana Studio, BlueGriffon, KompoZer;
- Bemutat készítő: Google Slides, OpenOffice Impress, Microsoft PowerPoint (18. ábra);



17. ábra. A Microsoft Word szövegszerkesztő.

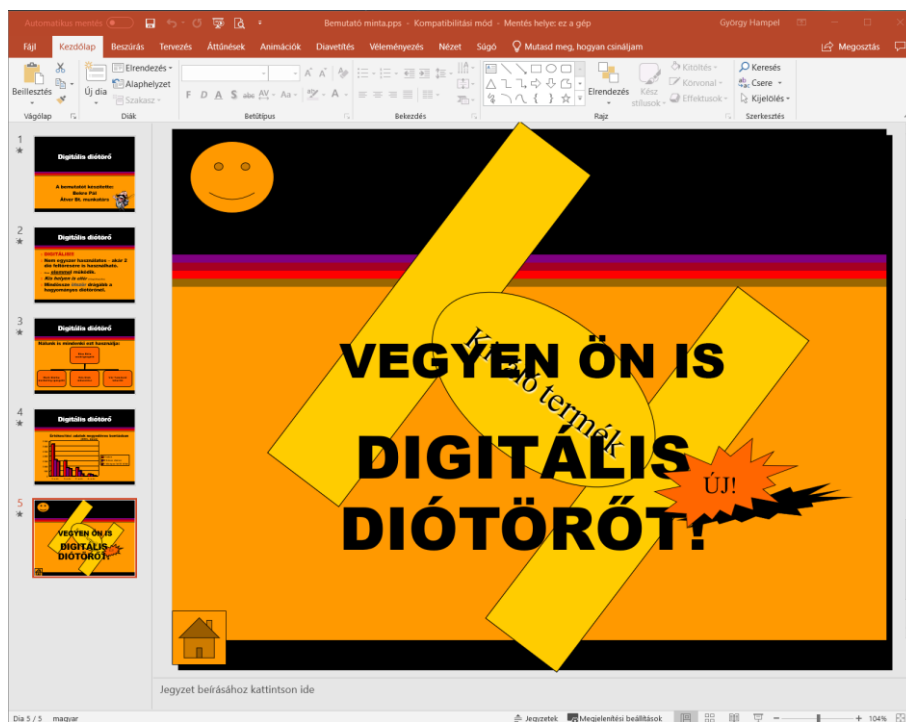
8.3. Bemutató-készítés

Ahogy az előző fejezettrészből kiderült, a bemutató készítésére alkalmas programok tulajdonképpen a szövegszerkesztők egy csoportját képviselik. **Bemutatót** (prezentációt) **többféle céllal** készíthetünk:

- Előadáshoz a mondanivaló szemléltetésére vetíthetünk projektoron, monitoron – ezt előszó melletti bemutatónak nevezzük. Azzal, hogy a mondanivalónkat multimédia elemekkel megerősítjük, kiegészítjük, hatékonyabbá válhat az információátadás (amennyiben e felkínált eszközöket okosan használjuk).

Interneten, illetve könyv formájában számos megszívlelendő tanácsot, ajánlást fogalmaztak meg az előszó melletti bemutatóra vonatkozóan; néhány ezek közül:

- vetítsünk kevés szöveget, ne legyen információval túlzsúfolva a vetített kép, hiszen az csak megerősíti, kiegészíti, és nem helyettesíti az előadót,
- alkalmazzunk nagy és könnyen olvasható betűméretet és betűtípust,
- az ábrák, diagramok legyenek könnyen és gyorsan áttekinthetőek,
- az effektusok (animáció, áttűnés, hanghatás) ne vonják el a figyelmet.
- Emlékeztetőket nyomtathatunk az előadáshoz kapcsolódóan, amelyet kioszthatunk a hallgatóságnak előadás után, vagy akár az előtt (hogy a hallgatóságnak legyen mire jegyzetelnie).
- Bármilyen információközlési céllal (például egy termék vagy szolgáltatás ismertetésére, reklámozására) szerkeszthetünk önállóan futó (kirakati) prezentációt.
- Az információközlő bemutató lehet akár interaktív is, ha lehetővé tesszük, hogy a felhasználó beavatkozzon a vetítés folyamatába.



18. ábra. Microsoft PowerPoint bemutató készítő.

A **bemutató készítő** programok fő **szolgáltatásai**:

- a szövegszerkesztőkben általában megtalálható szolgáltatások (lásd: az előző fejezetrészen);
- az oldalak, másnéven diák megjelenítése időzített áttűnésekkel;
- minden egyes diához jegyzetek készítése ez előadó számára;

- különböző objektumok (szöveg, állókép, táblázat, diagram) megjelenítése, mozgatása és törlése időzített animáció segítségével;
- hanghatások kezelése;
- videók kezelése;
- bármilyen külső helyi, vagy hálózaton elérhető tartalom kezelése, felhasználása;
- emlékeztető készítése;
- vetítés helyi gépen, vagy számítógép-hálózaton keresztül.

8.4. Táblázatkezelés

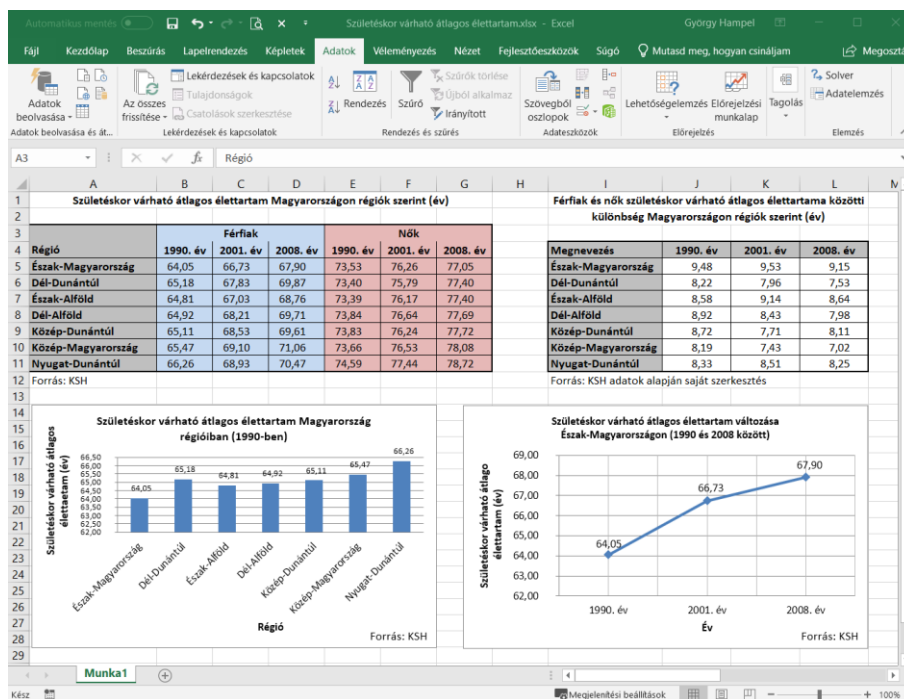
A **táblázatkezelők** alkalmasak arra, hogy adatokat egyszerűen, táblázatos formában tároljunk és jelenítsünk meg. Népszerű, sokak által használt táblázatkezelő a Microsoft Excel (19. ábra), a Google Spreadsheet és a LibreOffice Calc, de vannak, kevésbé elterjedt, az alapszolgáltatások szintjén ugyanazt nyújtó egyéb táblázatkezelő programok is.

Egy névvel ellátott **munkalap** betűvel jelölt oszlopokból (F) és számmal jelölt sorokból (3) áll, ezek metszéspontja a cella (F3). A cellák együttese a tartomány (blokk, tömb) (A1:D3). A munkalap oszlopainak és sorainak maximális száma programonként eltérő. Az adatokat a cellákba vihetjük be. Egye táblázatkezelőkben munkalapok egy csoportja **munkafüzetet** alkothat (ennek neve azonos a fájl nevével).

A **fő szolgáltatások**:

- Táblázatokban rögzített adatok rendszerezése, felsorolások, listák, kombinációs táblák, kimutatások létrehozása;
- Az adatok egy vagy több szempont szerinti rendezése, szűrése;
- Egyéni, valamint előre elkészített sablonokkal történő formázás, továbbá bizonyos feltételeknek megfelelő (vagy éppen nem megfelelő) adatok kiemelése feltételes formázással;
- Számolás matematikai és logikai műveletekkel, valamint rengeteg beépített matematikai, statisztikai, mérnöki, pénzügyi stb. függvény felhasználásával;
- Adatok szemléletes ábrázolása különböző diagramtípusokkal (oszlop-, sáv-, vonal-, pont-, kör- és térképdiaagram stb.);
- Az adatok importálása (megnyitás) és exportálása (mentés) segítségével kapcsolattartás különböző adatforrásokkal, helyi és távoli adatbázisokkal;
- Adatok elemzése (a matematikai-statisztika módszereivel), szimuláció és modellezés biztosítása és ezen keresztül döntéshozatal támogatása.
- A munkaterület egy meghatározott részének, illetve a diagramnak a kinyomtatása.

A **táblázatok dinamikusak**, ami azt jelenti, hogy ha egy alapadatot tartalmazó cella értéke megváltozik, akkor újraszámolódik az összes olyan cella, amelyek a megváltozott cellára hivatkozik.



19. ábra. Microsoft Excel táblázatkezelő.

A munkalap cellatartalma a következő lehet:

- Szám: A cella ilyen esetben csak számot tartalmazhat. Ha például mértékegységre is szükség van a cellában, azt a cellatartalom formázásával tudjuk megjeleníteni. A táblázatkezelők a dátumot és az időt is számként kezelik.

Példa: 12345. A cellában a szám jelenik meg (kiegészítve a formázásként hozzáadott mértékegységgel). Ha szöveggént íránk oda mértékegységet, vagy bármilyen egyéb betűt, akkor onnantól kezdve a cella tartalma nem szám!

- Függvény, képlet: A cella tartalmazhat egy vagy több függvényt, képletet, vagy valamilyen matematika-logikai műveletet. Valamilyen speciális jellel (+- előjel, =, @) kell kezdődjön.

Példa: =54321-12345 vagy =átlag(A1:A5) A cellában (alapesetben) a művelet eredménye jelenik meg.

- Hivatkozás: Függvényben vagy képletben hivatkozhatunk egy másik cella, vagy tartomány tartalmára az azonosítójával.
 - Cellahivatkozás: A G1-es cellába =F3-at beírva a G1 cellában mindig az F3 cella aktuális tartalma jelenik meg. A cellahivatkozás lehet relatív (F3), abszolút (\$F\$3) és vegyes (\$F3 vagy F\$3), ahol a \$ jellel az adott oszlopot, illetve sort rögzítjük, aminek akkor van jelentősége, ha a hivatkozást tartalmazó cellát másik cellába másoljuk.

- Tartományhivatkozás: Cellák együttesére hivatkozhatunk vele:
 - Ha a cellák egymással szomszédosak: A1:D5,
 - Ha a cellák egymással nem szomszédosak: A1:B5;D1:D5
 - Sorhivatkozás: Az 1:1 a munkalap teljes első sorára hivatkozik.
 - Oszlophivatkozás: Az A:D a teljes A, B, C és D oszlopra hivatkozik.
- A fentiek kombinációja: A G1-cellába az =54321-12345+F3 kifejezést beírva annak végeredménye jelenik meg.
- Szöveg: Minden, ami nem csak számot tartalmaz és nem függvény vagy képlet.

8.5. Adatbázis-kezelés

Az információ feldolgozására készített számítógépes programok az adatokat különböző strukturáltságban tárolják. Az adatok lehetnek lazább szerkezetben vagy szigorúbb struktúrában tárolva; ennek megfelelően beszélhetünk: **szövegszerű** rendszerekről és **adatszerű**, finoman strukturált rendszerekről.

- A szövegszerű tárolásnál dokumentumok (könyvek, cikkek stb.) alkotják a legkisebb elérési egységet. A dokumentum belső struktúra nélkül, ömlesztve tartalmazza az információt.
- Az adatszerű tárolásnál sokkal kisebb adatelemek (egyedek, objektumok) tulajdonságai is elérhetők és kezelhetők.

Az adatbázis-kezelés **kétféle adatbázistípust** különböztet meg:

- Az operációs adatbázisokat elsősorban az elektronikus tranzakció-feldolgozás során alkalmazzuk. Itt az adatok begyűjtése, módosítása és karbantartása napi rendszerességgel történik. Az ilyen adatbázisban tárolt adatok dinamikusak, ami azt jelenti, hogy folyamatosan változnak és mindig a friss állapotot tükrözik.
- Az analitikus adatbázisokat elsősorban elektronikus elemzési feldolgozásra használjuk. Ilyenkor az a fontos, hogy az adatok időbeli változását nyomon lehessen követni. Ez akkor hasznos, ha trendeket kell elemezni, statisztikákat kell készíteni, vagy stratégiai döntéseket kell hozni. Az ilyen adatbázisok statikus adatokat tartalmaznak, ami azt jelenti, hogy tartalmuk soha (vagy csak nagyon ritkán) változik, legtöbbször csak bővül.

Az adatbázis fogalmára koránt sincs egységes, mindenki által elfogadott meghatározás. Néhány adatbázis definíció: **Az adatbázis**

- adatok összessége.
- összetartozó és kapcsolódó adatok rendszere.
- alatt az adatoknak kapcsolataikkal együtt való ábrázolását, tárolását értjük.
- olyan adathalmaz, amely több, különböző felépítésű, egymással kölcsönös kapcsolatban lévő adatok gyűjteménye.
- általában és szigorúan véve olyan adatállomány, amely adatbázis-kezelő rendszerrel hozható létre és érhető el.

- a felhasználók által rugalmasan kezelhető adatok rendszere.
- összetartozó adatok azon rendszere, mely megosztott több felhasználó között, és az elérést egy központi vezérlőprogram szabályozza. A felhasználónak nem kell ismernie az adatok fizikai tárolási mechanizmusát.
- véges számú egyed-előfordulásnak, azok egyenként is véges számú tulajdonságértékének és kapcsolat-előfordulásainak az adatmodell szerint szervezett együttese.
- egy olyan integrált adatszerkezet, mely több különböző objektum előfordulási adatait adatmodell szerint szervezeten, tartósan tárolja olyan segédinformációkkal, metaadatokkal együtt, melyek a hatékonyság, integritásőrzés, adatvédelem biztosítását szolgálják. (A metaadat jelentése: adat az adattól.)
- mezőket tartalmazó bejegyzéseket, valamint azokat kombináló keresési, rendezési és egyéb műveleteket tartalmazó állomány.
- adatbank; adatok szervezett gyűjteménye, ahonnan információk kereshetők ki, rendezhetők és módosíthatók.

Az adatbázisok elvileg **tetszőleges méretűek** lehetnek: az adatok száma a nullától (az üres adatbázistól) a végtelen értékig terjedhet. Az elméletileg végtelen kapacitást a gyakorlatban a rendelkezésre álló hely, vagy a belső tárolási struktúra korlátozza.

Az adatbázis mindig egy **adatmodellhez kapcsolódik**; így beszélhetünk például hierarchikus, hálós, relációs modellről, illetve létezik olyan adatmodell is, amelynek nincs számítógépes megvalósítása, ilyen például az **egyed-kapcsolat** modell. A legtöbb adatbázis-kezelő és adatbázis a **relációs** modellre épül. (A relációs és az egyed-kapcsolat modellről az adatbázis-tervezésről szóló fejezetben lesz szó részletesebben.) Az adatbázis a valódi adatokon kívül tartalmazza az adatok típusait és kapcsolatait leíró ún. **metaadatokat** is.

Az adatbázis-kezelő (rendszer) a táblázatkezelőkkel szoros kapcsolatban áll (a legkorszerűbb táblázatkezelők képesek olyan egyszerű adatbázis-kezelő funkciók ellátására, mint amilyen például a szűrés), de az adatbázis-kezelők egyszerre akár sok táblázatban szétszórta hatalmas mennyiségű adatot is képes hatékonyan kezelni.

Az **adatbázis-kezelő** rendszer fő **feladatai** közé tartoznak a következők:

- Támogatnia kell valamilyen adatmodellt, hogy a valóságot le tudja képezni egy számunkra megfogható objektumra.
- Lehetővé kell tennie új adatbázisok létrehozását az adatbázis sémájának megadásával.
- Le kell tudnia írni és kezelnie kell az adatok közötti komplex kapcsolatokat.
- Biztosítani kell az adatbázisban tárolt adatok létrehozását, kezelését, különböző szempontok szerinti lekérdezését.
- Gondoskodnia kell az integritási (sértetlenségi), hatékonysági és védelmi feltételek megőrzéséről hosszú időn keresztül.

Az adatbázis-kezelő rendszer mindezek miatt egy **bonyolult programrendszernek tekinthető**, mely sok funkcióját, összetettségét tekintve leginkább az operációs

rendszerekhez hasonlítható. Az integritási, hatékonysági és védelmi feltételek ellenőrzését és betartatását az adatbázis-kezelő rendszer a háttérben végzi el a felhasználó közvetlen parancsa, tudta nélkül.

Adatbázis-kezelő rendszerek közé tartozik például: IBM DB2, Microsoft Access (20. ábra), Microsoft SQL Server, MySQL, Oracle SQL Developer, PostgreSQL.

Minden Access-ob...

autó

rendszám	típus	ár	iaji	gyártás dátum	csc	ember egyedi azonosítója	Hozzáadás
ADW-626	OPEL	1 450,00 Ft	5	2008.10.15.			20
AFG-218	LADA	150,00 Ft	5	1989.11.05.			11
BMW-419	TRABANT	45,00 Ft	3	1982.04.20.			
BNJ-768	LADA	195,00 Ft	5	1995.10.05.			33
BRF-280	LADA	400,00 Ft	3	1992.01.03.			20
CFL-655	TRABANT	25,00 Ft	5	1987.01.05.			
COB-468	OPEL	1 100,00 Ft	5	2000.02.05.			17
COO-381	DACIA	450,00 Ft	3	1997.10.05.			37
CQO-348	SKODA	1 700,00 Ft	5	2005.10.05.			1
CTS-109	LADA	350,00 Ft	3	1990.03.01.			32
CSY-238	OPEL	1 750,00 Ft	5	2006.12.12.			4
EEV-549	SUZUKI	1 000,00 Ft	5	2005.03.17.			11
EJC-752	SUZUKI	1 250,00 Ft	3	2002.03.22.			38
FBO-805	BMW	3 750,00 Ft	5	2008.05.09.			18
FSN-342	SKODA	400,00 Ft	5	1992.05.20.			5
FWF-749	SUZUKI	2 500,00 Ft	5	2007.03.11.			34
HSY-145	LADA	250,00 Ft	5	1989.12.05.			8
IAE-279	OPEL	1 300,00 Ft	5	2005.10.07.			35
KDX-418	SUZUKI	2 300,00 Ft	5	2008.05.30.			10
LUF-463	BMW	2 100,00 Ft	5	2003.05.17.			18
NEG-274	LADA	170,00 Ft	3	1992.10.08.			25
NGX-043	LADA	150,00 Ft	5	1987.10.05.			15
ORQ-879	OPEL	2 350,00 Ft	3	2009.04.05.			20
OXK-306	OPEL	1 170,00 Ft	5	2001.07.12.			6
PHC-834	OPEL	2 150,00 Ft	3	1997.04.11.			25

20. ábra. A Microsoft Access adatbázis-kezelő.

8.6. Ellenőrző kérdések

1. Mi az alkalmazás, irodai alkalmazás és az alkalmazói programcsomag? [88]
2. Mit jelent a WYSIWYG? [88]
3. Milyen szolgáltatásokat nyújtanak a szövegszerkesztők? [88]
4. Milyen típusú szövegszerkesztők léteznek és mi a különbség ezek között? [88]
5. Milyen formában tárolható a szöveg? [89]
6. Milyen szövegszerkesztőket ismer? [90]
7. Milyen céllal készíthetünk bemutatót? [90]
8. Mik a bemutató-készítő programok fő szolgáltatásai? [91]
9. Mire való a táblázatkezelő és milyen táblázatkezelő programokat ismer? [92]
10. Milyen részei vannak a munkafüzetnek? [92]
11. Milyen szolgáltatásokat nyújt a táblázatkezelő? [92]
12. Milyen típusú adatot tartalmazhat a munkalap egy cellája? [93]
13. Mi a különbség a szövegszerű és adatszerű tárolás között? [94]
14. Mi a különbség az operációs és az analitikus adatbázis között? [94]
15. Ismertessen néhány adatbázis meghatározást! [94]
16. Melyek az adatbázis-kezelő rendszer fő feladatai? [95]
17. Miért hasonlít az adatbázis-kezelő rendszer az operációs rendszerhez? [95]
18. Milyen adatbázis-kezelő rendszereket ismer? [96]

9. ADATBÁZIS-TERVEZÉS

Elvárt tanulási eredmények

Tudás

- Ismeri az adatbázis, a modell, az adatmodell és az adatmodellezés fogalmát
- Felsorolja az adatmodellek elemeit
- Felsorolja az adatbázis-tervezés lépéseit és átfogóan ismeri azok fő jellemzőit
- Ismeri az egyed-kapcsolat modell alapelemeit és ábrázolási módját
- Ismeri a relációs modell elemkészletét és ábrázolását

Képesség

- Rajzol egyed-kapcsolat modellt
- Elkészíti az adatbázis relációs modelljét
- Leírja egy adatbázis relációs sémáját
- Átkonvertál egyed-kapcsolat modellt relációs adatmodellre

Attitűd

- Elfogadja, hogy az adatok, információk kezelését tervezni kell
- Törekszik az adatok, adatbázisok strukturált leírására adatmodellek segítségével
- Elkötelezett a jól áttekinthető adatmodellek készítése iránt

Autonómia, felelősség

- Önállóan és kreatívan, az adatmodellre vonatkozó valamennyi szabály betartásával tervezi meg az adatbázist
- Önállóan, a vonatkozó szabályok alapján konvertál egyed-kapcsolat modellt relációs modellbe

9.1. Az adatbázis fogalma

Az előző fejezetben már volt szó az adatbázis-kezelőkről és az adatbázisra is adtunk már definíciót. Összefoglalva: **Adatbázisnak** nevezzük a káros és felesleges redundancia nélkül közösen tárolt, egymással kapcsolatban lévő adatok halmazát, amelynek alapvető célja egy vagy több alkalmazás optimális kiszolgálása. Az adatokat az azokat kezelő programoktól függetlenül tároljuk, így az adatokat és az azokat használó programokat is megváltoztathatjuk anélkül, hogy a másikat módosítanánk. Az adatbázisok legfontosabb jellemzője valamilyen fokú redundancia (adatismétlődés), továbbá, hogy változnak (többnyire növekszenek).

Azt is tudjuk már, hogy az adatbázis adatmodellhez kapcsolódik, vagyis az adatbázis egy **megvalósított adatmodell**, amely a valódi adatokon kívül tartalmazza az adatok típusait és kapcsolatait leíró metaadatokat is.

9.2. Az adatmodell fogalma és elemei

A **modellek** alapvető szerepet játszanak a környező világ megértésében, leképzésében és alakításában. A modellek teszik lehetővé a lényeg kiemelését és szemléltetését. A modell fogalma alatt rendszerint két különböző dolgot érthetünk:

- olyan rendszert, amely a valóság egy vizsgált szeletével szerkezetében, felépítésében vagy viselkedésben megegyezik, vagy hasonló jelleget mutat és célja a vizsgálatán keresztül a valóság állapotára, viselkedésére vonatkozó következtetések levonása, továbbá
- a modell kifejezéssel jelöljük azt az eszközrendszert is, amellyel az előző értelemben vett modell leírható, megadható.

A modell tehát jelenthet egy **jelölésrendszert** és egy elkészült **leírást** is.

Azokat a modelleket, amelyek az adatok struktúrájának leírására szolgálnak, **adatmodelleknek** nevezzük. Az adatmodell központi szerepet játszik az adatbázis-kezelésben, hiszen ezen keresztül adhatjuk meg a megvalósítandó rendszer leírását az adatbázis-kezelőnek. Ahhoz, hogy használni tudjuk az adatbázis-kezelőt, ismernünk kell a hozzá tartozó adatmodellt.

Az adatbázis-kezelés fejlődésével újabb és újabb adatmodellek jöttek létre, így például a hierarchikus, a hálós, az egyed-kapcsolat, a relációs vagy az objektum-orientált adatmodellek. Az **adatbázis-kezelő rendszerek** egyik fő jellemzője, hogy **mely adatmodellhez kapcsolódnak**, például a relációs adatbázis-kezelő rendszer a relációs adatmodellen nyugszik. Az adatmodell nem igényli feltétlenül egy megvalósított adatbázis-kezelő rendszer létét; léteznek olyan adatmodellek, melyekhez nem létezik adatbázis-kezelő rendszer – ilyen például az egyed-kapcsolat modell.

Az **adatmodellezés** tehát az **adatállományok tervezésének korszerű módszere**. Az adatmodellezéssel az a **cél**, hogy egy információs rendszer adatait és az adatok között fennálló kapcsolatokat következetesen ábrázolva, elősegítsük a számítógépes információ-feldolgozást.

Egy **adatmodell** a következő **elemeket tartalmazza**:

- Az egyedek a valóság olyan dolgai, amelyek más dolgoktól megkülönböztethetők, azaz az egyed a valós világban létező, fogalmi vagy fizikai léttel rendelkező dolog

(elem, objektum) amelyet tulajdonságokkal akarunk leírni: ember, dolgozó, vásárló, áru, jármű, ingatlan, könyv, irat stb. Ugyanaz a dolog többféle egyednek is tekinthető, például egy ember lehet dolgozó és vásárló is. A képviselt konkrét elemek halmazát egyedhalmaznak nevezzük: egy céges adatbázisban a dolgozó mint egyedhalmaz az összes ott dolgozót tartalmazza.

- Tulajdonságnak nevezzük a valóság dolgainak azon jellemzőit, amelyekkel az egyes egyedeket jellemezzük. Egy embernél tulajdonság a név, születési hely és idő, anyja neve, végzettség, lakcím, munkahely, beosztás stb.; a név értékei lehetnek: Alma Alfonz, Birs Bea, Citrom Cecil stb. Az egyes egyedek az egyedhalmazban a felsorolt tulajdonságértékeikkel azonosíthatók.
- Kapcsolatnak nevezzük az egyedek közötti viszonyok fogalmi tükörképeit. Egyedek közötti viszony lehet például az, hogy a dolgozó egy adott cég alkalmazottja és ennek egy konkrét előfordulása, ha Alma Alfonz az XYZ Kft. egyik munkavállalója.

9.3. Az adatbázis-tervezés lépései

Egy adatbázis létrehozását – jó esetben – mindig az **adatbázis tervezés** előzi meg. Gondosan fel kell mérni az igényeket és meg kell fogalmazni a problémákat, elvárásokat és csak utána szabad megkezdeni a tényleges, fizikai megvalósítást.

Egy vállalati adatbázis kifejlesztése hónapokat vehet igénybe. Az időnek körülbelül 70-75%-a a tervezésre, 10-15%-a a programozásra, a maradék pedig a tesztelésre fordított idő. Az alapos tervezés azért is fontos, mert enélkül az elkészült rendszer átláthatatlan lesz és az utólagos módosítás már nagyon körülményes.

A tervezés lépései:

1. Az igények összegyűjtése, elemzése.
2. A koncepcionális terv elkészítése.
3. Az adatbázis-kezelő rendszer kiválasztása.
4. Az adatbázis-kezelő rendszertől függő leképezés.
5. Fizikai tervezés.
6. Megvalósítás.

Az igények összegyűjtése, elemzése, a feladat specifikációja

Ebben a fázisban nagyon gondosan és átgondoltan kell felderíteni a fő alkalmazási területeket; tanulmányozni kell az adott területtel rokon, már meglévő alkalmazásokat és azok dokumentációit. Meg kell vizsgálni a jelenlegi megvalósításokat (akár még a nem számítógépes megoldásokat is), valamint azok alkalmazásának körülményit. A felhasználói igények, elvárások összegyűjtése érdekében célszerű a leendő felhasználókkal is elbeszélgetni.

Az adatok, információk összegyűjtése után olyan specifikációt (leírást) kell készíteni, amely tartalmazza a felhasználói igényeket kielégítő tárolandó adatokat és a szükséges feldolgozási műveleteket, lekérdezéseket.

A koncepcionális terv elkészítése

A koncepcionális terv készítésének folyamán kell a magas szintű modellt kialakítani. A modell segítségével meg kell fogalmazni az előre tervezhető lekérdezéseket és feldolgozási műveleteket. A terv előnyei:

- könnyen érthető formában mutatja az adatbázis szerkezetét, az adatcsoportokat és azok kapcsolatait, valamint az esetleges korlátozásokat,
- olyan terv, leírás, amit az adatbázis-kezelő rendszer kiválasztásával, módosításával nem kell megváltoztatni,
- mind a felhasználóknak, mind pedig a programozóknak újabb ötleteket ad,
- mivel könnyen megérthető, segíti a felhasználó és a programozó közötti párbeszédet.

A koncepcionális terv elkészítéséhez gyakran az egyed-kapcsolat modellt használják, mivel kifejezők (az egyed típusok és tulajdonságok mellett a kapcsolatok típusait is ábrázolják), egyszerűek (nem szakemberek is viszonylag könnyen megérthetik), kevés fogalmat használnak (így ez az adatmodell rövid idő alatt megtanulható), szemléletes ábrákkal dolgozik és egyértelműek, szinte nem lehet félreérteni azokat. (Megjegyzendő, hogy komplex adatbázisok leírása nehézkessé válhat ezzel a modellel.)

Amennyiben hasonló igényeket fogalmaznak meg felhasználók egy-egy adatra vagy műveletre vonatkozóan, akkor az igényeket össze kell fűsülni. A legnehezebb talán az eltérések és egyezőségek megtalálása és egyértelműsítése (például a Dolgozó és Alkalmazott egyed jelentheti ugyanazt, de Dolgozó és Vezető már nem feltétlenül.)

Az adatbázis-kezelő rendszer kiválasztása

Az adatbázis-kezelő rendszer kiválasztásában igen sok tényező játszhat szerepet, így:

- a feladat természete: meghatározza, hogy milyen adatmodellre épülő rendszert célszerű használni; relációs, vagy esetleg más adatmodellre épülő adatbázis-kezelőt;
- gazdaságossági megfontolások: a hardver és szoftver beszerzési költségei, betanítási, működtetési, karbantartási és egyéb költségek;
- a rendszer szolgáltatásai, felhasználóbarát felület, hálózati működés, optimálisan kezelhető adatmennyiség nagysága stb.

Egyszerűbb, pontosan tervezhető feladatok elvégzésére nem kell feltétlenül adatbázis-kezelő rendszert használni. Néha előnyösebb lehet egy táblázatkezelő, vagy esetleg egy saját fejlesztésű program, mint egy bonyolult és költséges rendszer. Adatbázis-kezelő rendszert akkor érdemes használni, ha:

- több felhasználó és program is használja ugyanazt az adatbázist,
- az adatok gyorsan szaporodnak, módosulnak,
- nagy az adatbázis,
- a különböző adattípusok között bonyolult kapcsolatrendszer áll fenn és
- az adatellenőrzésre, adatbiztonságra nagy az igény.

Adatbázis-kezelő rendszertől függő leképezés

A leképezés lényegében a logikai adatmodellről függő szabályok alkalmazása és konvertálása a kiválasztott adatbázis-kezelő által kezelt formára (például relációs modellre).

A folyamat automatizálható az ún. CASE (Computer Aided Software Engineering Tools) programok segítségével. Ezek az eszközök a magas szintű modellből kiindulva a kiválasztott adatbázis-kezelő rendszer adatleíró nyelvén írják le az adatbázis szerkezetét.

Ebben a fázisban történik a lekérdezések megtervezése is. Például relációs adatbázis-kezelő használata esetén a lekérdezések leírásához használhatunk relációs algebrát, vagy használhatunk ún. SQL (Structured Query Language: strukturált lekérdező nyelv, a relációs adatbázis-kezelők többsége által támogatott) utasításokat.

Fizikai tervezés

Itt kell döntenie a tárolási szerkezetről és a hozzáférési módokról, amit az adatbázis-kezelő rendszer mellett az operációs rendszer is befolyásolhat. Döntő fontosságú lehet a lekérdező és aktualizáló – vagyis a beszűrő, törlő és módosító – műveletek gyakorisága és egymáshoz való viszonya, az adatok gyors elérést biztosító (de ugyanakkor több tárhelyet foglaló) megoldások alkalmazása stb.

Megvalósítás

Az előző fázisok végeredményeként létrejön az adatbázis szerkezete és az üres fájlok. Az így kapott adatbázist tölthetjük fel adatokkal.

A programozók megírják a műveletek kódjait, vagy a felhasználói programokat, amelyek használhatják az adatbázis-kezelő nyelv parancsait. Korszerű rendszerek felhasználóbarát grafikus felülete lehetővé teszi, hogy akár komolyabb programozási ismeretek nélkül hozzunk létre kódokat. Néhány szolgáltatás azok közül, amit a rendszerek biztosítanak:

- menük, gombok, űrlapmezők és egyéb objektumok létrehozása az adatokkal kapcsolatos műveletek végzésére,
- adatbevitel ellenőrzéséhez szükséges kódok biztosítása,
- összetartozó adatcsoportok adatainak bevitelére alkalmas űrlapok és a megjelenítésükre alkalmas lekérdezések létrehozása,
- adatokból képzett jelentések sémáinak elkészítése.

A tényleges adatok felvitele előtt célszerű a kész adatbázist mintaadatokkal kipróbálni, hogy ne túl későn derüljenek ki az esetleges hibák. Ezt azért is célszerű megtenni, mert még ekkor is előfordulhat, hogy mind a felhasználó, mind a programozó olyan újabb lehetőségeket fedez fel, amire eddig nem is gondolt.

A rendszer megvalósítása után használat közben felmerülhetnek hibák, problémák, amiket javítani kell. Egyéb módosítási igények is jelentkezhetnek – részben új felhasználói kívánságok, részben a változó külső körülmények következtében –, amelyeket utólag szintén be kell építeni.

9.4. Az egyed-kapcsolat modell elemkészlete, ábrázolás a modellben

Az **egyed-kapcsolat modellt** általában a relációs adatmodellezés bevezetőjeként alkalmazzák. Előnyei közé tartozik az egyszerűség és a könnyű átalakíthatóság a relációs modell felé. Az egyed-kapcsolat modell, **három alapelemen** nyugszik:

- az egyedtípusok (egyedhalmazok),
- az egyedek tulajdonsága és
- az egyedek közötti kapcsolatok foglmain.

A modell fontos jellemzője, hogy **grafikus jelölésrendszert alkalmaz**. A grafika a szöveges leíráshoz képest sokkal kifejezőbb, lényegre törőbb, könnyebben és gyorsabban áttekinthetőbb az emberek számára, így kiválóan alkalmas a fontosabb fogalmak és kapcsolatok megjelenítésére.

Egyedtípusok: Egy, a külvilág többi részétől egyértelműen megkülönböztethető dolgot jelöl.

- Erős egyedtípus: Benne az egyes egyedek egyértelműen azonosíthatók, mivel azoknak van azonosító jellegű (azaz kulcs-) tulajdonsága. Például emberek azonosíthatók személyi igazolvány számmal. Jele egy téglalap, melynek belsejében az egyedtípus azonosító neve áll. (Lásd: 21. ábra bal oldala.) (A 21-29. ábra a <https://erdplus.com/#/> adatbázis-modellező weboldallal készült.)
- Gyenge egyedtípus: Az egyes egyedeket a tulajdonságok nem határozzák meg egyértelműen, csak a kapcsolatai révén lesznek meghatározottak. Példa: a szervizben lévő műszaki cikk, amelyet a tulajdonos nevével azonosítanak. Jele egy kettős téglalap, melynek belsejében az egyedtípus azonosító neve. (Lásd: 21. ábra bal oldala.)



21. ábra. Normál egyed (bal oldal) és gyenge egyed (jobb oldal).

Az egyedtípus önmagában nem állhat, kell legyen tulajdonsága.

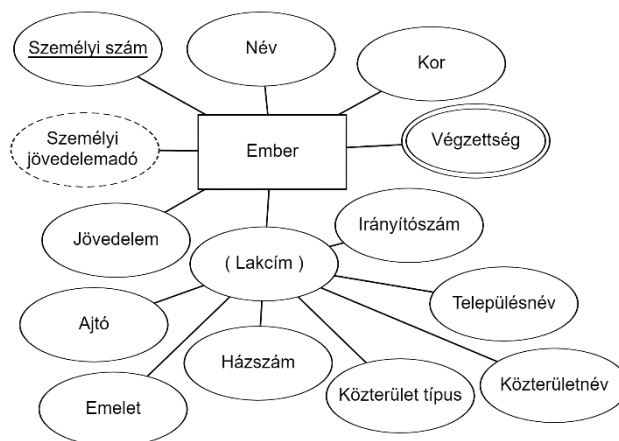
Tulajdonság: Ez az egyed egy meghatározott jellemzője. Az egyed olyan tulajdonságcsoporthal rendelkezik, amely azonosítja az egyedet. Például embernél a személyi szám, név és cím vagy autónál a rendszám, típus, szín és ár.

- Egyszerű tulajdonság: Egy elemi értékkel leírható tulajdonságot ad meg. Az ember kora például egyszerű tulajdonság, hiszen egy skalár szám elegendő a megadásához. A végzettség már nem egyszerű tulajdonság, több értéket is felvehet egyidejűleg, hiszen egy embernek egyszerre többféle végzettsége is lehet. A tulajdonságokat ellipszisben adjuk meg, az ellipszis közepébe írva a tulajdonság azonosító nevét. A tulajdonság önmagában nem állhat, ezért mindig meg kell adni, hogy mely egyedtípushoz, vagy kapcsolathoz kötődik. A kapcsolódást egy vonallal jelöljük, ami a megfelelő tulajdonságot és az egyedtípust köti össze.
- Kulcs tulajdonság: Az egyed egyértelmű azonosítására szolgáló tulajdonság. Az ember egyedtípus esetén például a személyi szám, személyi igazolvány szám, TAJ

szám, útlevekszám töltheti be ezt a szerepet. Az egyedi azonosításra alkalmas tulajdonságok (azonosítók) közül egyet kiválasztunk, ez lesz a kulcs tulajdonság és úgy jelöljük, hogy a tulajdonság azonosító nevét aláhúzzuk egy folytonos vonallal.

- **Összetett tulajdonság:** Olyan tulajdonság, amely több egyszerű tulajdonságra bontható. Ilyen tulajdonság lehet például a cím, hiszen ez felbontható az irányítószám, település, utca, házszám elemi adatok együttesére. Az összetett tulajdonságot is ellipszissel jelöljük, melyhez hozzákötjük az illeszkedő elemi tulajdonságok szimbólumait.
- **Többértékű tulajdonság:** Olyan tulajdonság, amely nem csak egy elemi értéket, hanem több elemi értéket, értékek egy tömbjét veheti fel. Így például az ember egyed iskolai végzettség tulajdonságának leírására több elemi értéket is meg lehet adni, hiszen többféle végzettsége is lehet valakinek. A többértékű tulajdonságot egy dupla keretű ellipszissel jelöljük.
- **Leszármaztatott tulajdonság:** Olyan tulajdonság, melynek értéke más tulajdonságokból származtatható. Így például egy ember személyi jövedelemadója kiszámolható a jövedelem nagyságából és az SZJA-kulcsból. A leszármaztatott tulajdonság jele egy szaggatott vonallal határolt ellipszis. Azt nem jelöljük, hogy mely más tulajdonságokból és mi módon származtatható az érték (a modell ezt nem támogatja).

Példa: A 22. ábra az ember normál egyedtypust mutatja a következő tulajdonságokkal: Személyi szám (kulcs tulajdonság), Kor és Név (egyszerű tulajdonság), Végzettség (többértékű tulajdonság), Lakcím (összetett tulajdonság, hozzá kapcsolva a résztulajdonságok: Irányítószám, Településnév, Közterületnév, Közterület típus, Házszám, Emelet, Ajtó), Jövedelem (egyszerű tulajdonság), Személyi jövedelemadó (leszármaztatott tulajdonság).



22. ábra. Normál egyed tulajdonságokkal.

Kapcsolat: az egyedek között fennálló viszonyt mutatja. (Megjegyzés: 23-28. ábráról az áttekinthetőség kedvéért az egyedtypusok tulajdonságait nem jelöltük. A kapcsolattípusok jelölésének az itt leírttól eltérő módjai is léteznek.)

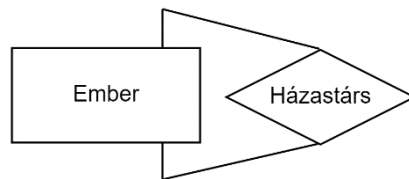
- **1:1 kapcsolat:** A kapcsolatban mindkét egyedtypus előfordulásai csak egyetlenegy előforduláshoz rendelődnek a másik egyedtypusból. Példa: a férfiak és nők külön egyedhalmazt alkotnak és egy férfi csak egy nőnek lehet a férje és egy nő is csak egy

férfinak lehet a felesége. A kapcsolatot egy rombuszsal jelöljük, amelybe beírjuk kapcsolatot leíró azonosító nevét. A rombusz átellenes csúcaiból egy-egy vonalat húzunk (23. ábra).



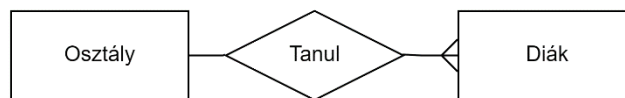
23. ábra. 1:1 kapcsolat.

- Rekurzív (visszamutató) kapcsolat: Egy egyedtípus előfordulásai saját egyedtípusuk előfordulásaihoz kapcsolódnak. Ha a férfiak és nők egyetlen egyedhalmazt alkotnak, ez a 24. ábra szerint jelölhető.



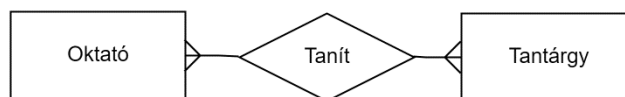
24. ábra. Rekurzív kapcsolat.

- 1:N kapcsolat: Abban különbözik az 1:1 kapcsolattípustól, hogy az egyik egyedtípus előfordulásai több előfordulással tarthatnak kapcsolatot a másiktól, de a másik előfordulás továbbra is csak egy előforduláshoz kapcsolódhat. Például egy iskola tanulója csak egy osztályba járhat, egy osztályban viszont több diák is tanulhat. Az 1:N kapcsolat ábrázolásánál azon egyedbe, melyből több is kapcsolódhat a másik egyedhez, egy „varjúláb” vezet (25. ábra).



25. ábra. 1:N kapcsolat.

- N:M kapcsolat: Olyan kapcsolattípus, melyben mindkét egyedtípus előfordulásai több előfordulással is tarthatják a kapcsolatot a másik egyedtípusból. Például egy oktató több tantárgyat taníthat és egy tárgy oktatásában több oktató is részt vehet. A kapcsolat ábrázolásánál mindkét kapcsolódó egyedhalmaz felé „varjúláb” mutat (26. ábra).



26. ábra. N:N kapcsolat.

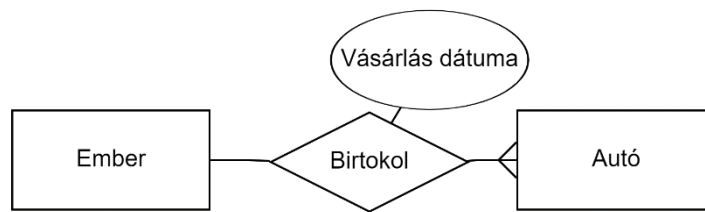
- N-ed fokú kapcsolat: A kapcsolatban nemcsak kettő, hanem ennél több egyed vesz részt. Ilyen kapcsolatra példa a vásárol kapcsolat, melyben a vásárló, az eladó és az áru kapcsolódik össze, ahol a vásárlásnál vevő(k) terméke(ke)t vesz(nek) egy vagy több árustól.

- Totális (teljes) és parciális (részleges, opcionális) kapcsolat: Egy egyed típus totálisan vesz részt a kapcsolatban, ha minden egyed-előfordulása részt vesz egy kapcsolat előfordulásban, azaz nincs olyan egyed-előfordulás, mely nem kapcsolódna egy másik egyed típus valamely előfordulásához. Azok az egyedek, amelyek nem totálisan vesznek részt a kapcsolatban, parciális kapcsolatot alkotnak. Például abban az esetben, ha minden autónak van tulajdonosa, akkor az autó totális kapcsolatban van egy emberrel a tulajdonosi kapcsolatban. Az ember viszont csak parciálisan vesz részt a tulajdonosi kapcsolatban, mivel lehetnek olyan emberek, akiknek nincs autójuk. Jelölése: a teljes oldalról folytonos, az opcionális oldal felől szaggatott vonal (27. ábra).



27. ábra. N:M kapcsolat.

Tulajdonságot nem csak egyed típushoz, hanem kapcsolathoz is hozzárendelhetünk. (28. ábra)



28. ábra. Kapcsolat tulajdonsággal.

9.5. A relációs adatmodell elemkészlete, ábrázolás a modellben

A **relációs adatmodell** napjaink legelterjedtebb – adatbázis-kezelők által támogatott – adatmodellje. Egyszerűségének következtében gyorsan népszerűvé vált a felhasználók körében és sok implementációja (megvalósítása) született még a személyi számítógépek piacán is. Az elméleti megalapozottság a kutatók, a szakemberek szimpátiáját is kiváltotta, ezért a modell számos fejlesztés alapját képezi.

A **relációs adatmodell elemkészlete** és azok **ábrázolása**:

- Egyedhalmazok: A relációs modellben az adatokat kétdimenziós táblában tároljuk. Ábrázoláskor az egyedhalmaz lesz a táblázat neve. A táblának minden sorában azonos számú oszlop található. A táblázat első sora a reláció fejrésze. Itt található az egyértelmű oszlopnevek (a reláció attribútumai, tulajdonságai), melyek az oszlopokban lévő adatok jelentését adják meg. A táblázat fejrésztől különböző oszlopait szokták mezőnek is nevezni. A táblázat sorai a reláció sorai, más néven rekordok. Az oszlopok és a sorok sorrendjét felcserélhetjük, ez a relációt nem, csak megjelenését változtatja meg. A modell fontos eleme a kulcs. Az elsődleges kulcs a táblázat sorainak egyértelmű megkülönböztetésére, azonosítására használható. A kulcs nem veheti fel kétszer ugyanazt az értéket egy táblán belül. Elemi kulcsról, elsődleges kulcsról beszélünk akkor, amikor a reláció valamelyik eleme (a táblázat valamelyik attribútuma, oszlopa) alkalmas a rekordok (sorok) egyértelmű megkülönböztetésére. Ha egynél több attribútum szükséges egy rekord

megtalálásához, összetett elsődleges kulcsról beszélünk. Ha egy relációt több attribútum is egyértelműen meghatároz, akkor szabadon választhatjuk meg, hogy melyik legyen az elsődleges kulcs, a többit alternatív kulcsnak nevezzük. Az elsődleges kulcsot a többi tulajdonságtól aláhúzással különböztetjük meg. A kulcs kiválasztásánál ügyelnünk kell arra, hogy a kulcsban szereplő attribútumok száma a lehető legkevesebb legyen. Az 5. táblázat az Ember relációt (táblát) és néhány rekordját mutatja, ahol a sorszám az emberek egyértelmű beazonosítását lehetővé tevő elsődleges kulcs.

Ember

<u>Sorszám</u>	Név	Születési idő	Neme	Jövedelem eFt
52	Alma Alfonz	1975.10.23.	Férfi	305
48	Birs Bea	1989.01.12.	Nő	195
64	Citrom Cecil	1995.08.28.	Nő	210
27	Dinnye Dénes	1982.04.07.	Férfi	225

5. táblázat. Az Ember tábla adatokkal.

Egy táblából a táblával logikai kapcsolatban lévő másik tábla egy meghatározott sorára az idegen kulcs segítségével tudunk hivatkozni. Az idegen kulcsnak megfelelő érték abban a táblában, amelyiknek rekordjára hivatkozunk, elsődleges kulcs.

- Kapcsolatok ábrázolása: A relációs adatmodellben a kapcsolatokat is táblázatok segítségével ábrázoljuk. A legegyszerűbb esetben, 1:1 kapcsolat esetén eljárhatunk úgy, hogy az egyik táblát kiegészítjük a logikailag hozzá kapcsolódó másik tábla elsődleges kulcsa felhasználásával. (6. táblázat)

Férfi

<u>Sorszám</u>	Név	Születési idő	Jövedelem eFt
52	Alma Alfonz	1975.10.23.	305
27	Dinnye Dénes	1982.04.07.	225

Nő

<u>Sorszám</u>	Név	Születési idő	Jövedelem eFt	Házastárs sorszám
48	Birs Bea	1989.01.12.	195	52
64	Citrom Cecil	1995.08.28.	210	

6. táblázat. A Férfi és a Nő tábla adatokkal. Feltételezzük, hogy mindenkinek csak egy házastársa lehet (1:1 kapcsolat). A Nő tábla tartalmazza a férfi tábla elsődleges kulcsát idegen kulcsként. A 48-as nő és az 52-es férfi házastársak, 64-es nőnek és a 27-es férfinak nincs házastársa.

- 1:N kapcsolatnál is alkalmazhatjuk ezt a megoldást azzal a megkötéssel, hogy az N-oldali táblához kell hozzáírunk az idegen kulcsot. (7. táblázat)

Ember

<u>Sorszám</u>	<u>Név</u>	<u>Születési idő</u>	<u>Neme</u>	<u>Jövedelem eFt</u>
52	Alma Alfonz	1975.10.23.	Férfi	305
48	Birs Bea	1989.01.12.	Nő	195
64	Citrom Cecil	1995.08.28.	Nő	210
27	Dinnye Dénes	1982.04.07.	Férfi	225

Autó

<u>Rendszám</u>	<u>Típus</u>	<u>Szín</u>	<u>Ember sorszám</u>
ABC123	Toyota	Fehér	52
DEF456	Toyota	Fehér	52
GHI789	BMW	Kék	67
JKL159	Opel	Fehér	

7. táblázat. Ember és Autó tábla összekapcsolva. Feltételezzük, hogy egy embernek több autója, egy autónak egy tulajdonosa lehet (1:N kapcsolat).

- N:M kapcsolatnál mindenképpen egy új, kapcsolatot biztosító táblát kell létrehoznunk. Ha a kapcsolat rendelkezik valamilyen tulajdonsággal, szintén kapcsolat tábla jelentheti a megoldást – függetlenül a kapcsolat típusától. A kapcsolat táblába oszlopai a két összekapcsolt tábla elsődleges kulcsai lesznek (továbbá a kapcsolat tulajdonsága(i)). A kapcsolat táblát alkalmazhatjuk 1:1 és 1:N kapcsolat esetében is (8. táblázat). A kapcsolatot biztosító táblában N:M kapcsolat esetén a két tulajdonság együtt alkothat elsődleges kulcsot, 1:N kapcsolatnál az N oldali táblából származó oszlop lehet kulcs, 1:1 típusú kapcsolatnál bármelyik oszlopot választhatjuk elsődleges kulcsnak.

Ember

<u>Sorszám</u>	<u>Név</u>	<u>Születési idő</u>	<u>Neme</u>	<u>Jövedelem eFt</u>
52	Alma Alfonz	1975.10.23.	Férfi	305
48	Birs Bea	1989.01.12.	Nő	195
64	Citrom Cecil	1995.08.28.	Nő	210
27	Dinnye Dénes	1982.04.07.	Férfi	225

Birtokol

<u>Sorszám</u>	<u>Rendszám</u>
52	ABC123
52	DEF456
67	GHI789

Autó

<u>Rendszám</u>	<u>Típus</u>	<u>Szín</u>
ABC123	Toyota	Fehér
DEF456	Toyota	Fehér
GHI789	BMW	Kék
JKL159	Opel	Fehér

8. táblázat. Ember és Autó tábla kapcsolat táblával összekapcsolva.

Abban az esetben, ha csak a táblázat nevét és a tulajdonságokat adjuk meg, az adatokat nem, **relációs sémáról** beszélünk. Jelölése: relációnév (attribútumok felsorolása). A relációs modell egy vagy több relációs sémát (táblát) tartalmazhat. A relációs sémákból álló halmazt relációs adatbázissémának, vagy egyszerűen adatbázissémának nevezzük.

A 8. táblázat relációs sémája:

Ember(Sorszám, Név, Születési idő, Neme, Jövedelem eFt)

Autó(Rendszám, Típus, Szín)

Birtokol(Sorszám, Rendszám)

9.6. Egyed-kapcsolat modell átalakítása relációs modellé

Az adatbázis tervezés egyik fontos lépése a **logikai adatmodell adatbázismodellbe történő konvertálása**. Ha a logikai modellünket egyed-kapcsolat modellben rajzoltuk meg és relációs adatbázis-kezelővel dolgozunk, akkor az előbbi modellt át kell konvertálnunk az utóbbiba.

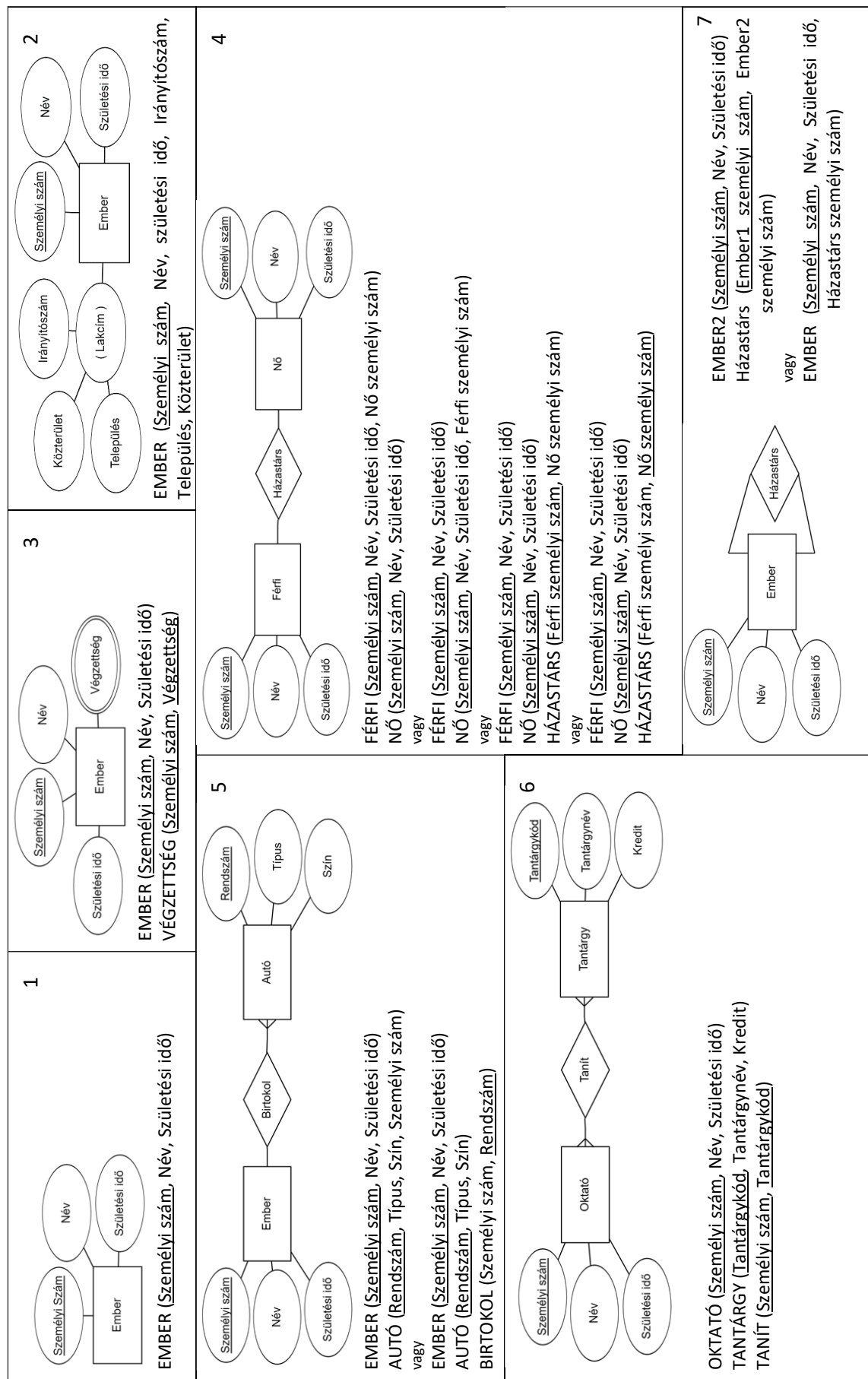
Az **egyed típusok konvertálása**: Egy táblázat egy egyedtípusnak felel meg. A reláció neve megegyezik az egyedtípus azonosító nevével. A táblázat szerkezete is az egyed-kapcsolat modellből származik.

A **tulajdonságok konvertálása**:

- Egyszerű tulajdonságok: a táblázat oszlopai lesznek. (29. ábra-1)
- Elsődleges kulcs: a táblázat megfelelő oszlopában a megnevezést aláhúzzuk. (29. ábra-1)
- Összetett tulajdonság: A táblázatból elhagyjuk, csak az alkotórészeit, a résztulajdonságokat vesszük fel; minden egyes résztulajdonság külön oszlopot alkot. (29. ábra-2)
- Leszármaztatott tulajdonság: nem vesszük fel a táblába, hiszen az számolható más adatokból.
- Többértékű tulajdonság: Az ilyen tulajdonságok számára külön táblát hozunk létre, amelynek egyik oszlopa a tulajdonság, másik oszlopa annak a táblának az elsődleges kulcsa, amelyhez hozzá akarjuk kapcsolni. (29. ábra-3)

A **kapcsolatok átkonvertálása**: A kapcsolatot leképezzük egy külön táblába oly módon, hogy a kapcsolat neve lesz a tábla neve, a tábla oszlopait pedig a kapcsolódó egyedek kulcsai és a kapcsolat tulajdonságai alkotják. Ezután meghatározzuk a kapcsolati tábla kulcsát, és ha az megegyezik valamely egyed kulcsával, akkor a kapcsolatból leképezett táblát beolvasztjuk az egyedből leképezett táblába.

- 1:1 kapcsolatnál: tetszőlegesen választhatunk, hogy melyik egyedhez ágyazzuk be. (29. ábra-4)
- 1:N kapcsolatnál: az N-oldali egyedből leképezett táblával vonható össze. (29. ábra-5)
- N:M kapcsolat esetén: nem vonható össze egyik egyedből leképezett táblával sem. (29. ábra-6)
- Rekurzív kapcsolatnál az összekapcsolandó egyedeket két külön táblába tesszük és a két tábla között kapcsolat táblát hozunk létre, amely a két tábla elsődleges kulcsát tartalmazza, vagy a másik megoldás, hogy az elsődleges kulcsot hozzáírjuk még egyszer táblához. (29. ábra-7)



29. ábra. Egyed-kapcsolat átalakítása relációs modellé.

9.7. Ellenőrző kérdések

1. Mi az adatbázis? [99]
2. Mi a modell, az adatmodell és mit értünk adatmodellezés alatt? [99]
3. Milyen elemeket tartalmaz az adatmodell? Jellemezze az egyes elemeket! [99]
4. Sorolja fel az adatbázis-tervezés lépéseit és ismertesse az első kettőt! [100]
5. Sorolja fel az adatbázis-tervezés lépéseit és ismertesse a másodikat és harmadikat! [101]
6. Sorolja fel az adatbázis-tervezés lépéseit és ismertesse az utolsó kettőt! [102]
7. Mik az egyed-kapcsolat modell fő jellemzői? [103]
8. Milyen egyedtípusok és hogyan ábrázolhatók az egyed-kapcsolat modellben? [103]
9. Hogyan ábrázolhatók az egyes tulajdonságtípusok az egyed-kapcsolat modellben? [103]
10. Készítsen egy emberek tulajdonságait tartalmazó modellt, amely valamennyi tulajdonságtípust tartalmazza! [104]
11. Hogyan ábrázolhatók a kapcsolattípusok az egyed-kapcsolat modellben? Példát is adjon meg! [104]
12. Melyik a legelterjedtebb adatmodell és miért? [106]
13. Hogyan ábrázoljuk az egyedhalmazokat a relációs modellben? Példával is! [106]
14. Hogyan ábrázoljuk a kapcsolatokat a relációs modellben? Példával is! [107]
15. Mi a relációs séma és hogyan adjuk meg? Példát is adjon meg! [108]
16. Hogyan konvertálhatunk át egyedtípusokat egyed-kapcsolat modellből relációs adatmodellbe? Példával is! [109]
17. Hogyan konvertálhatunk át tulajdonságokat egyed-kapcsolat modellből relációs adatmodellbe? Példával is! [109]
18. Hogyan konvertálhatunk át kapcsolatokat egyed-kapcsolat modellből relációs adatmodellbe? Példával is! [109]

10. A PROGRAMOZÁS ALAPJAI

Elvárt tanulási eredmények

Tudás

- Ismeri az algoritmus és a számítógépes algoritmus fogalmát, jellemzőit
- Felsorolja az algoritmusokkal szemben támasztott követelményeket és az algoritmusok alapvető lépéseit
- Érti, hogy mit jelent az algoritmizálhatóság
- Ismeri a programozás és a programozási nyelv fogalmát
- Felsorolja a programozás lépéseit
- Ismeri az alacsony és magas szintű nyelvek fő jellemzőit
- Érti a forrásnyelvű program és gépi kódú program közötti különbséget
- Érti az interpreter, a compiler és az assembler közötti különbséget
- Tudja, hogy milyen részekből áll a fejlesztői környezet
- Ismeri az utasítás, a változó és az értékadás fogalmát, az adatbevitel és adatmegjelenítés célját
- Érti a szekvencia, szelekció és iteráció közötti különbséget
- Ismeri a folyamatábra és a pszeudokód fogalmát

Képesség

- Lerajzol algoritmust folyamatábra segítségével
- Megfogalmaz algoritmust pszeudokód használatával

Attitűd

- Törekszik a fegyelmezett, algoritmikus gondolkodás elsajátítására és alkalmazására
- Törekszik egy adott feladat megoldása szempontjából a leghatékonyabb algoritmus megtalálására
- Elkötelezett a jól áttekinthető folyamatábra és pszeudokód készítése iránt
- Nyitott a magas szintű programozási nyelvek szolgáltatásainak és alkalmazási lehetőségeinek megismerésére a saját szakterületén

Autonómia, felelősség

- Önálló javaslatokat fogalmaz meg a szakterületén felmerülő problémák algoritmizálására
- Önállóan készít algoritmizálható feladatokhoz folyamatábrát és pszeudokódot
- Felelősséget vállal az elkészített algoritmusok helyes működéséért

10.1. Az algoritmus és a programozás fogalma, értelmezése

Általánosságban az **algoritmus** (vagy eljárás) olyan megengedett lépésekből álló módszer, utasítás(sorozat), részletes útmutatás, recept, amely valamely felmerült probléma megoldására alkalmas. Szűkebb értelemben az algoritmus egy matematikai, illetve logikai probléma megoldását jelenti véges számú lépéseken keresztül. A véges számú lépések meghatározott sorrendben követik egymást.

Az algoritmusok adatok és **utasítások**, **műveletek** halmazából állnak. A műveletek lehetnek elemiek, amelyek tovább már nem bonthatók és lehetnek összetettek, amelyek elemi műveletekből épülnek fel. Vannak műveletek, amelyek felcserélhetők, vagy párhuzamosan végrehajthatók.

Az **algoritmussal** szembeni **követelmények**:

- Teljesség: a megoldáshoz szükséges valamennyi információt tartalmaznia kell;
- Végrehajthatóság: a lépéseket végre lehessen hajtani;
- Egyértelműség: valamennyi lépést csak egyféleképpen lehessen végrehajtani;
- Végesség: véges számú lépésből kell állnia (legyen kezdete és vége);

Az **algoritmusok** alapvető **lépései**:

1. Input: az adatok bekérése;
2. Számítás: numerikus számítások elvégzése az adatokon;
3. Döntés: egy vagy több alternatív lépés kiválasztása az adatok, számítások értékének összehasonlítása, vagy számítási műveletek eredményeként;
4. Output: az eredmények megjelenítése;

Az **algoritmizálhatóság** azt jelenti, hogy egy feladat megoldása tetszőleges alkalommal, mindig azonos módon és véges számú lépésben érhető el. Algoritmizálható feladat például egy műszaki hiba elhárítása, de nem automatizálható feladat egy vállalati stratégiai döntés információhiányos környezetben.

A **számítógépes algoritmus** a számítógép számára készült utasítássorozat, ami lehetővé teszi, hogy a számítógép akár önállóan is képes legyen feladatokat megoldani. Az utasítássorozatok elkészítését nevezzük programozásnak. A számítógép számára csak olyan algoritmust fogalmazhatunk meg, amelynek lépései felbonthatók a processzor által végrehajtható lépésekre, utasításokra.

Az algoritmus **hatékonyságát** a lépésszám (az algoritmus által megtett lépések száma) és a memóriaigény határozza meg.

10.2. A programozás fogalma és lépései

A **programozás** „a számítógépes programok készítésének művészete és tudománya” (Microsoft Press, 1999, 451. oldal). Szükséges hozzá legalább egy programozási nyelv ismerete, természetesen rendelkezni kell algoritmizálási ismeretekkel, és ezen túlmenően nem árt tisztában lenni a felhasználói felületek tervezésével és a hardvereszközök jellemzőivel is.

A számítógép-programozás egy vagy több algoritmus megvalósítását jelenti egy bizonyos programozási nyelven. **Lépései** a következők:

1. A megoldandó probléma meghatározása, felmérése a majdani felhasználók igényeinek figyelembevételével, majd az igények alapján a specifikáció készítése.
2. Valamilyen programtervezési módszerrel a programszerkezet megalkotása és a használandó eszközök kiválasztása (hardver- és szoftverkörnyezet, programozási nyelv stb.).
3. A program elkészítése a kiválasztott programozási nyelven (kódolás) és – ha szükséges – fordítása.
4. A kész program tesztelése, a hibák keresése, a talált hibák kijavítása. (Megjegyzés: Hibátlan program nincs, legfeljebb olyan, amelyben még nem találtak hibát. Annyit azért elvárhatunk, hogy a program minden előre látható helyzetben helyesen működjön és hiba esetén könnyen javítható legyen.)
5. A dokumentáció elkészítése, amelynek célja egyrészt a program használatának megkönnyítése, másrészt későbbi javítás és fejlesztés elősegítése.

10.3. A programozási nyelv fogalma és csoportjai

A programozási nyelv

- egy mesterséges nyelv, amellyel a számítógép által feldolgozható, végrehajtható utasítások sorozatát adhatjuk meg;
- számítógépes programok létrehozására, módosítására, továbbfejlesztésére alkalmas fejlesztői program(ok);
- olyan, az ember által olvasható és értelmezhető utasítások sorozata, aminek a segítségével közvetlenül, vagy közvetve (például: gépi kódra fordítás után) közelíthetjük a számítógéppel egy adott feladat elvégzésének módját.

Ahhoz, hogy egy program **helyesen működjön**, alapvetően **két feltételt** kell teljesítenie: egyrészt nem szabad logikai (szemantikai) hibákat tartalmaznia, másrészt a programíráshoz használt programnyelv helyesírási szabályait (szintaxis) be kell tartania.

A **programozási nyelveket két nagy kategóriába** sorolhatjuk:

1. Alacsony szintű programozási nyelvek
 - Gépi kód: Mindig egy adott géptípushoz, illetve gépcsaldhoz tervezik, tehát gépfüggő. A gép számára közvetlenül értelmezhető adatsor; az egyetlen „nyelv”, amit a számítógép központi parancsvégrehajtó egysége, a processzor megért. A számítástechnikában használt műveletek és adatok (általában bináris vagy hexadecimális számokkal ábrázolt) olyan sora, amely a számítógép processzora számára közvetlen utasításként értelmezhető. Az egyes processzortípusok utasításkészletei egymással egyáltalán nem kompatibilisek (ami azt jelenti, hogy egy másik processzorra készített programot nem tudnak végrehajtani. Mivel közvetlenül, fordítás nélkül értelmezhető a számítógép számára, a végrehajtása nagyon gyors, a hardver a programozó számára közvetlenül elérhető és programozható, ugyanakkor hátránya, hogy a gép architektúrájához, logikájához igazítva kell a programot megírni és ehhez nagyon kell ismerni az adott gépet.

Példa egy utasításra: 0000 0011 1100 0011 (vagy rövidebben hexadecimális számrendszerben leírva: 03C3h)

- Az assembly nyelv a gépi kódhoz legközelebb álló alacsony szintű programozási nyelv. Rövidítéseket, betű- és mozaikszavakat (mnemonikus kódokat) használnak. Az assemblyben írt program minden utasítása megfelel egy gépi kódú utasításnak. Az assemblyben írt programot futtatás előtt le kell fordítani gépi kódra assemblerrel. A nyelv alkalmazásának előnyei és hátrányai ugyanazok, mint gépi kód esetén, ugyanakkor az angol nyelvhez közel álló mnemonikus kód használata miatt a gépi kódnál könnyebben áttekinthető.

Példa egy utasításra: ADD AX, BX

2. Magas szintű programozási nyelvek: Ezek olyan számítógépes programozási nyelvek, amelyek a gépi nyelvhez képest egy absztrakciós réteget nyújtanak, amelynek feladata elrejteni a programozó elől a hardvert és a hozzá tartozó gépi utasításokat. Gépi kód vagy assembly kód helyett az emberi (vagy pontosabban az angol) nyelvhez közel álló kulcsszavakat használhatunk. Ide sorolható nyelvek: Basic, C, C++, C#, Fortran, Java, Pascal stb., gyakorlatilag minden, ami nem gépi kód és nem assembly.

Példa utasításra: PRINT "A téglalap kerülete: "; 2*(a+b)

Előnyei:

- platformfüggetlenség: hardvertől és operációs rendszertől (nagy részt) független nyelvek és
- egyszerűség: használatukkal a komplex programok írása egyszerűvé válik.

Hátránya, hogy az így megírt ún. forrásprogramot le kell fordítani a futtatás előtt és a lefordított gépi kódú program lassabb lesz, kevésbé lesz hatékony az eleve assembly-ben vagy gépi kódban írt változatához képest.

10.4. Forrás, értelmező, fordító

A számítógépes programokat a számítógép háttértárolóján találjuk a többi adatállománnyal együtt. A típusuk (kiterjesztésük például com vagy exe Windows operációs rendszerben) vagy (Unixban és Linuxban) attribútumuk alapján tudjuk eldönteni, hogy **futtatható állományról** van-e szó, vagy egyéb adatállományról. Ha ezeket a programokat szöveggént próbálnánk megjeleníteni a képernyőn, akkor egy számunkra teljesen értelmezhetetlen karakterhalmazt látnánk. Ennek az a magyarázata, hogy egy programot tartalmazó állományt nem nekünk, hanem a számítógépünknek kell megértenie és végrehajtania. Felmerülhet a kérdés, hogy amikor programozunk, akkor ilyen fájlokat nekünk kell-e gépeléssel létrehoznunk? Nyilván nem. Programozási nyelveket fejlesztettek ki erre a célra (lásd az előző fejezet részt).

Forrásnyelvű programnak nevezzük azt a fájlt, melyet egy programozási nyelven mi hozunk létre. Ezután vagy egy **értelmező program (interpreter)**, vagy egy **fordítóprogram (compiler)** segítségével alakíthatjuk át a számítógépünk processzora számára érthető és végrehajtható kódra. Az interpreter a forrásnyelvű programunknak egyszerre csak egy sorát értelmezi, lefordítja, végrehajtja a processzorral majd ezt elfelejtve, rátér a következő sorra. Ekkor tehát nem keletkezik gépi kódú, vagy más néven natív fájl. Ha így

akarjuk futtatni a programot, akkor mindig szükség van az interpreterre. Ezzel szemben egy compiler a teljes forrásnyelvű programunkat lefordítva létrehoz egy gépi kódú fájlt (Windowsban egy exe kiterjesztésű fájlt), amelyet már önállóan futtathatunk a számítógépünkön. Amennyiben assembly-ben írjuk a programot, azt az ún. **assemblerrel** alakíthatjuk át futtatható gépi kódú programmá.

Ahhoz tehát, hogy saját készítésű programokat futtathassunk számítógépünkön, ismernünk kell egy programozási nyelvet, rendelkezünk kell egy szövegszerkesztő programmal a forrásnyelvű program létrehozásához, valamint egy interpreterrel vagy compilerrel a lefordításhoz.

Léteznek ún. **fejlesztői környezetek** (például Delphi és Visual Studio), amelyek lehetővé teszik akár több programozási nyelv egyidejű használatát, kombinálását, tartalmaznak szövegszerkesztőt, fordítót és egyéb, a programozást megkönnyítő segédeszközt is kínálnak; a programkód egy részét önállóan megírhatják, még könnyebbé, hatékonyabbá téve a programozást.

10.5. Programok felépítése, szerkezete

A programok írása során az **alapl műveletek** az adatok bevitele (input), az adatok megjelenítése (output), a változók definiálása, az értékadás, az elágazás és a ciklusszervezés. Az ezekhez szükséges utasítások programnyelvenként eltérnek.

Az **utasítás** olyan tevékenység pontos megfogalmazása, amelyet tovább részletezni nem tudunk, vagy nem akarunk, és a végrehajtó számára egyértelmű jelentéssel bír. A számítógépes utasítás a gép belső működéséhez, állapotváltozáshoz nyújt információt. A számítógép az utasítások hatására a kiinduló, kezdeti állapotból több közbülső állapoton keresztül eljut egy végső állapotba, melynek hatására az input adatokból eredményadatok állnak elő.

Az **adatbevitel** célja, hogy a program megkapja azokat az adatokat, amelyek szükségesek a feladat elvégzéséhez. Az **adatmegjelenítés** célja a programvégrehajtás eredményeként létrejövő adatok megjelenítése (képernyőn, nyomtatón stb.) vagy átadása más programok számára. Meg kell tervezni az input és output felhasználóbarát megvalósítását is, ami ugyancsak a programnyelvtől függ.

A **változó** adat, amelynek értéke a program futása közben változhat. A változót definiálásakor egy azonosítóval látjuk el (ezzel lehet rá később hivatkozni) és meghatározzuk az adattípusát is (ez határozza meg, hogy milyen típusú értékeket vehet fel).

Az **értékadás** olyan utasítása, amellyel valamilyen – az adattípusának megfelelő – értéket adhatunk a változónak.

Bármely program megírható úgy, hogy az **utasítások három szerkezetben** találhatók benne:

- Szekvencia: Egymás után végrehajtandó tevékenységek sorozata.
- Elágazás (szelekció): valamilyen döntés (feltételvizsgálat) során a programvégrehajtás több irány közül valamelyiken (de csak egyen) haladhat tovább.

- Iteráció (ciklus): A cikluson belüli utasítások, utasításcsoportok (bizonyos feltételek teljesülése esetén) többször is végrehajtnak.

A szekvencia, szelekció és iteráció egymásba ágyazhatók, így például egy iteráció tartalmazhat szekvenciálisan, szelektíven vagy ciklikusan végrehajtandó tevékenységeket.

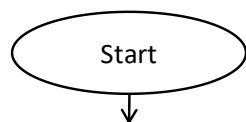
10.6. Az algoritmus és leíró eszközei: folyamatábra és pszeudokód

Az **algoritmusok** megtervezésére, **leírására** több módszer áll rendelkezésre, amelyek közül a folyamatábrát és a pszeudokódot nézzük meg.

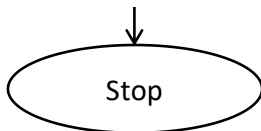
A folyamatábra

A **folyamatábra** egy könnyen áttekinthető grafikus ábrázolási mód, amelyben geometriai alakzatokat használunk. Az egyes algoritmust felépítő lépéstípusokhoz meghatározott síkidomokat rendelünk, és ezek belsejébe írjuk bele a lépés lényegét. A lépések végrehajtási sorrendjét (egyeses) nyilakkal jelöljük. Az egymást követő lépéseket rendszerint függőlegesen ábrázoljuk, fentről lefelé és balról jobbra olvassuk. Minden alakzathoz legalább két nyíl (egy bemenő és egy kimenő) tartozik – kivétel ez alól az algoritmust indító és lezáró alakzat. Az előbbihez csak kimenő, az utóbbihoz csak bemenő nyíl tartozik.

Határszimbólumok

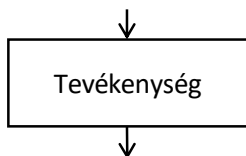


Az algoritmus eleje: A program kezdete, csak egy lehet belőle, egyetlen kimenete van. Jelölés: ellipszis, benne Start szóval



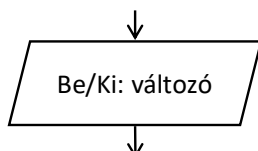
A program vége: Egy program többféleképpen is végződhet, azaz több is lehet belőle, ill. több bemenete is lehet, de kimenete nem lehet. Jelölés: ellipszis, benne Stop szóval

Értékadás, általános művelet



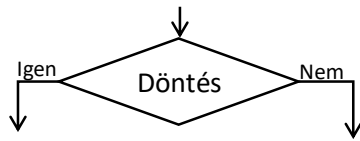
Tevékenység csomópont: Ezen áthaladva a tevékenység (utasítás) végrehajtnak. Legalább egy bemenete (több is lehet) és mindig egy kimenete van. Jelölés: téglalap, benne a tevékenység lényegével, illetve értékadással.

Input/output művelet



Adatbevitel, adatkirás jelölése: Adatbekérésre, illetve megjelenítésre, kiíratásra szolgál. Legalább egy bemenete (több is lehet) és mindig egy kimenete van. Jelölés: paralelogramma, benne a be- vagy kimenet jelzése és a változó.

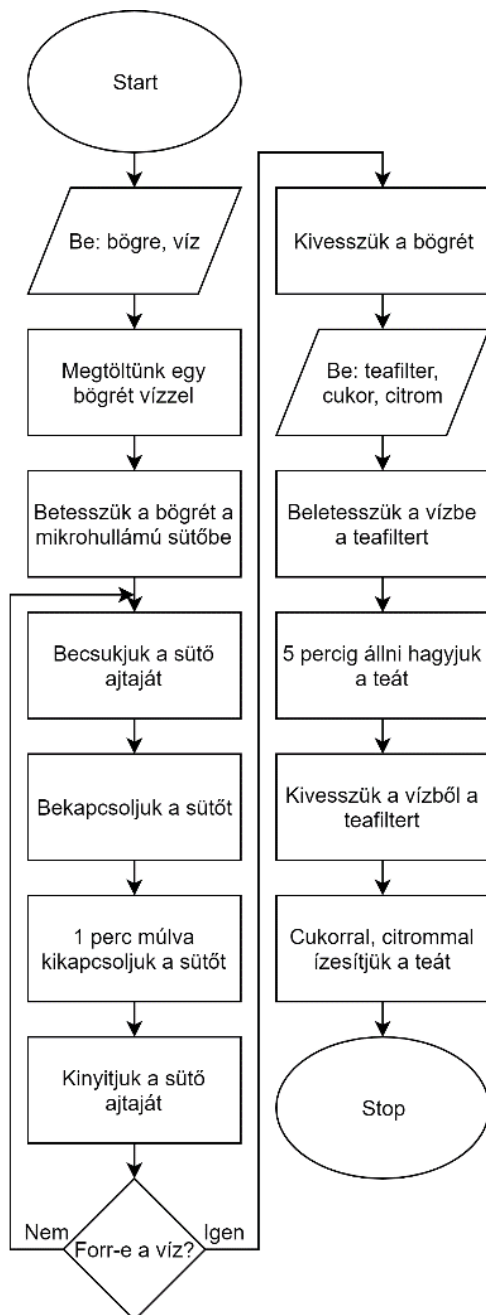
Elágazás



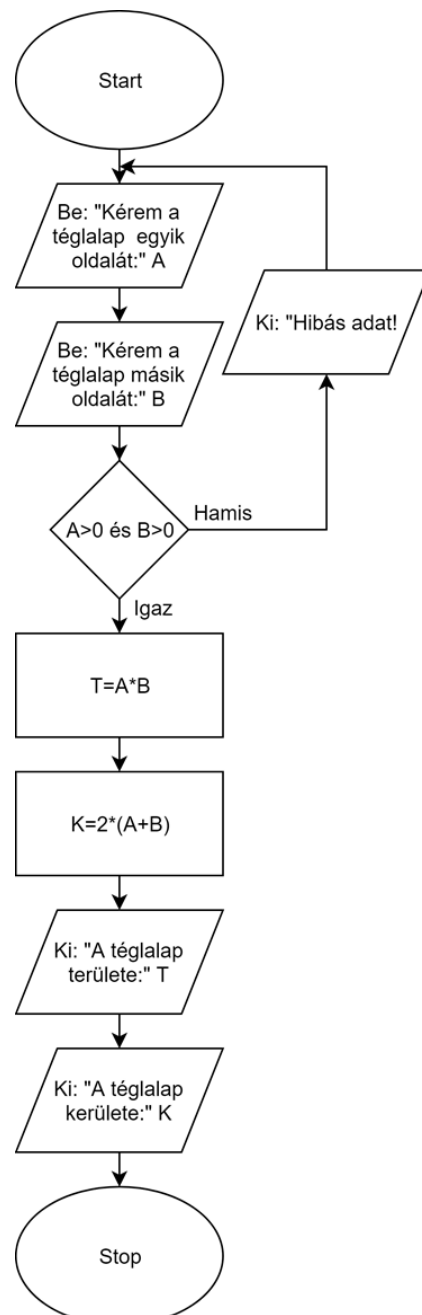
Döntés csomópont: A benne lévő feltételtől függően a vezérlés vagy az igen (igaz) vagy a nem (hamis) ágon folytatódik. Legalább egy bemenete (több is lehet) és mindig két kimenete van. Jelölés: rombusz, benne a feltétel.

A 30. ábra két, folyamatábrával leírt algoritmust mutat be. Bal oldalon a teafőzés lépései, jobb oldalon a téglalap területének és kerületének kiszámításának lépései láthatók. Mindkét ábra finomítható és további lépésekkel egészíthető ki.

Példa: Teafőzés mikrohullámú sütőben



Példa: A téglalap területe és kerülete



30. ábra. Példa algoritmus folyamatábrával történő leírására.
(az ábra a <https://www.draw.io/> weboldalon készült)

Pszudokód

Algoritmusokat **szöveges formában** is rögzíthetünk. Akár szabatosan, minden részletre kiterjedően, teljes mondatokban, a mondatokat bekezdésekre tagolva írhatjuk le az utasításokat; gondoljunk például a szakácskönyvekre. A módszer csak egyszerű problémák megoldásakor járható, sok lépésből álló bonyolult algoritmusoknál nem, mert nehezen lesz követhető.

Az algoritmusok másik, szöveggel szemléltetett változata a **pszudokód (mondatszerű leírás)**, ami bonyolultabb feladatok leírását is lehetővé teszi. Ez a természetes nyelvhez hasonló, de annál tömörebb, mesterséges formális nyelv; néhány állandó jelentésű szóból és változókból áll. Az algoritmus lépéseit nem egész mondatokkal fogalmazzuk meg, **csak a lényegét írjuk le.:**

- Adatbevitel és kivitel: Be/Ki: adat vagy felsorolás (egyéb paraméterek)

Be: ... felsorolás... [megszorítások]

Ki: ... felsorolás [kiírási formák]

- Szekvencia:

Tevékenység 1

Tevékenység 2

...

Tevékenység n

Vagy alternatív jelöléssel:

Tevékenység 1:Tevékenység 2:....:Tevékenység n

- Szelekció:

Ha Feltétel Akkor

Tevékenység(ek)

Elágazás vége

Vagy többágú szelekció esetében:

Ha Feltétel Akkor

Tevékenység(ek) 1

Egyébként

Tevékenység(ek) 2

Elágazás vége

- Ciklus:

- Elöltesztelő ciklus: A ciklus először megvizsgálja, hogy teljesül-e a feltétel. Amennyiben nem, akkor a ciklus vége utáni tevékenységek végrehajtása következik; amennyiben igen, akkor végrehajtódnak a cikluson belüli (ciklusmag) tevékenységek, majd újra kezdődik a feltételvizsgálat. Ebből következően előfordulhat, hogy a ciklusmag egyszer sem hajtódnak végre.

Ciklus amíg Feltétel Igaz

Tevékenység(ek) 1

Ciklus vége

- Hátultesztelő ciklus: A feltételvizsgálat a ciklusmag után áll, ennek eredményeként a cikluson belüli tevékenységek legalább egyszer végrehajtódnak. Programozási nyelvtől függ, hogy a hátultesztelő cikluson belül a folytatás vagy a kilépés feltételeit kell-e megadni.

Hátultesztelő ciklus belépési feltétellel: A ciklus mindaddig ismétlődik amíg a ciklus végén megadott feltétel igaz.

Ciklus

Tevékenység(ek)

Amíg Feltétel Igaz

Ciklus vége

Hátultesztelő ciklus kilépési feltétellel: A ciklus mindaddig ismétlődik amíg a ciklus végén megadott feltétel igazzá nem válik (ha igazzá válik, akkor következnek a ciklus utáni tevékenységek).

Ciklus

Tevékenység(ek)

Mígnem Feltétel Igaz

Ciklus vége

- Növekményes (számláló) ciklus: Egy változó értékét növeljük, vagy csökkentjük egy kezdeti értéktől egy megadott végértékig meghatározott lépésközzel. Ha a változó értéke eléri a végértéket, az algoritmus folytatódik a ciklust követő tevékenységek végrehajtásával.

Ciklusváltozó = ...-tól ... -ig Lépésköz ...

Tevékenység(ek)

Ciklus vége

Példa: Teafőzés mikrohullámú sütőben

Bögre beszerzése

Bögre megtöltése vízzel

Bögre berakása a mikrohullámú sütőbe

Ciklus

Mikrohullámú sütő ajtajának becsukása

Mikrohullámú sütő bekapcsolása

Várakozás 1 percig

Mikrohullámú sütő kikapcsolása

Mikrohullámú sütő ajtajának kinyitása

Víz hőmérséklet ellenőrzése

Mígnem forr a víz

Ciklus vége

Bögre kivétele sütőből

Teafilter berakása vízbe

Tea áll 5 percig

Teafilter kivétele a bögréből

Tea ízesítése cukorral

Tea ízesítése citrommal

Példa: A téglalap területe és kerülete

Ciklus amíg $A \leq 0$ vagy $B \leq 0$

Be: „Kérem a téglalap egyik oldalát:” A

Be: „Kérem a téglalap másik oldalát:” B

Ha $A \leq 0$ vagy $B \leq 0$ Akkor

Ki: „Hibás adat!”

Elágazás vége

Ciklus vége

$T = A * B$

$K = 2 * (A + B)$

Ki: „A téglalap területe:” T

Ki: „A téglalap kerülete:” K

10.7. Ellenőrző kérdések

1. Mi az algoritmus általánosságban és szűkebb értelemben? [113]
2. Milyen követelményeket támaszthatunk az algoritmusokkal szemben? [113]
3. Melyek az algoritmusok alapvető lépései? [113]
4. Mit jelent az algoritmizálhatóság? [113]
5. Mi a számítógépes algoritmus? [113]
6. Mi határozza meg egy algoritmus hatékonyságát? [113]
7. Mit értünk programozás alatt? [113]
8. Milyen lépései vannak a programozásnak? [114]
9. Mi a programozási nyelv? [114]
10. Mi szükséges ahhoz, hogy egy program helyesen működjön? [114]
11. Mi jellemzi az alacsony szintű programozási nyelveket? [114]
12. Mi jellemzi a magas szintű programozási nyelveket? [115]
13. Mi a futtatható állomány, a forrásnyelvi program, az interpreter, a compiler és az assembler? [115]
14. Mik azok a fejlesztői környezetek? [116]
15. Melyek a programok írása során alkalmazott alapl műveletek? [116]
16. Mi az utasítás? [116]
17. Mi az adatbevitel és adatmegjelenítés célja? [116]
18. Mit nevezünk változónak? [116]
19. Mi az értékadás? [116]
20. Mi a szekvencia, szelekció és iteráció? [116]
21. Mi a folyamatábra, mi a jelkészlete és hogyan használhatjuk? [117]
22. Folyamatábrával írja le a teafőzés egy lehetséges algoritmusát! [118]
23. Folyamatábrával készítsen egy lehetséges megoldást a téglalap oldalai hosszának bekérésére, a területének és kerületének kiszámítására, majd az eredmény megjelenítésére! [118]
24. Mi a pszeudokód, hogyan használhatjuk? [119]
25. Pszeudokóddal írja le a teafőzés algoritmusát! [121]
26. Pszeudokóddal készítsen algoritmust a téglalap oldalai hosszának bekérésére, a területének és kerületének kiszámítására, majd az eredmény megjelenítésére! [121]

IRODALOMJEGYZÉK

- Amdahl, G. M., Blaaw, G. A., & Brooks, F. P. (1964): Architecture of the IBM System 360. *IBM Journal of Research and Development*, 8(2): 21–36.
- Atkinson, C. (2008): *Ne vetíts vázlatot! A hatásos prezentáció*. Szak Kiadó. Bicske.
- Bakó A. (2006): Multimédia alapjai. (Széchenyi István Egyetem) Budapesti Műszaki Főiskola Könyvtár. Budapest. <<http://users.atw.hu/bmfgazdmen/data/index.php?dir=Multimedia+alkalmazasok%2F&download=mmalap1.pdf>>
- Bártfai B. (2016): *Office 2016*. BBS-Info. Budapest.
- Bártfai B. (2016): *Számítógéphasználat kezdőknek*. BBS-Info. Budapest.
- Béres I. (2006): *Alkalmazott számítástechnika*. Kereskedelmi és Idegenforgalmi Továbbképző Kft. Budapest.
- Berke J., Csák M. (2007): *Operációs rendszerek*. Debreceni Egyetem Agrár- és Műszaki Tudományok Centruma Agrárgazdasági és Vidékfejlesztési Kar. Debrecen.
- Bodnár I., Magyary G. (2005): *Az informatika elméleti alapjai*. Kiskapu Kft. Budapest.
- Boros J. (1995): *Információs rendszerek: az információs rendszerek alapelemei és életciklusa*. Magyar-Amerikai Vállalkozási Alap Nemzetközi Menedzser Központ. Budapest.
- Chikán A. (2008): *Vállalatgazdaságtan*. Aula Kiadó. Budapest.
- Czenky M. (2001): *Adatmodellezés. SQL és Access Alkalmazás*. Computerbooks Kiadó. Budapest.
- Debreceni L. (2005): *Multimédia, videó, tömörítés, rendszerarchiválás, Internetes levelezés*. Dunaújvárosi Főiskola Kiadói Hivatala. Dunaújváros.
- Dési I., Nagy I. (2001): *Informatikai fogalmak kisszótára*. Korona Kiadó. Budapest.
- Domán C., Szilágyi R., Varga B. (2007): *Statisztikai elemzések alapjai*. Közgazdasági-módszertani képzés fejlesztéséért Alapítvány. Miskolc.
- EUR-LEX (2009): Az Európai Parlament és a Tanács 2009/24/EK irányelve a számítógépi programok jogi védelméről. <<https://eur-lex.europa.eu/legal-content/HU/TXT/HTML/?uri=CELEX:32009L0024&from=HU>>
- Farkas A. (2004): *Számítástechnika vázlatok*. Szerzői Kiadás. Szombathely.
- Farkas C. (2006): *Informatikai ismeretek a 9-10. évfolyam részére*. Jedlik Oktatási Stúdió. Budapest.
- Fodor G. A. (2016): *Windows 10 és Office 2016 felhasználóknak*. Jedlik Oktatási Stúdió. Budapest.
- Galambos G. (2004): *Operációs rendszerek általános elmélete*. SZTE JGYF Kiadó. Szeged.
- Hadházy I. K., Györgyi G., Novák I. (2006): *Napjaink operációs rendszerei*. Bessenyei György Könyvkiadó. Nyíregyháza.

- Halassy B. (1994): *Az adatbázistervezés alapjai és titkai*. IDG Magyarországi Lapkiadó Kft. Budapest.
- Halassy B. (1996): *Ember, információ, rendszer*. IDG Magyarországi Lapkiadó. Budapest.
- Hallová, M., Fandel, I., Hennyeyová, K. (2014): *Information Technologies and Data Processing*. Szlovák Agrártudományi Egyetem. Nyitra.
- Hampel Gy., Fabulya Z. (2006): *Bevezetés a mérnökinformatikai eszközök használatába*. Szent István Egyetem, Gépészmérnöki Kar, Mechatronikai és Géptani Intézete. Gödöllő.
- Han, J., Kamber, M. (2004): *Adatbányászat. Konceptiók és technikák*. Panem Kiadó. Budapest.
- Herdon M., Fazekasné K., Magó Zs., Németh Z. (2007): *Informatika*. Debreceni Egyetem Agrár- és Műszaki Tudományok Centruma Agrárgazdasági és Vidékfejlesztési Kar. Debrecen.
- Hunyadi L., Vita L. (2008): *Statisztika I*. Aula Kiadó. Budapest.
- IEC 60027-2 szabvány
- Juhász S. (2011): *Vállalati információs rendszerek műszaki alapjai*. Szak Kiadó. Budapest.
- Kárpáti A. (2013): Az informatikai kompetenciától a digitális pedagógiáig, a nemzetközi kutatások tükrében. In: Dringó-Horváth I., N. Császi I. (szerk.): *Digitális tananyagok – oktatásinformatikai kompetencia a tanárképzésben*. L'Harmattan Kiadó. Budapest. pp. 15–32.
- Kacsukné B. L., Kiss T. (2009): *Bevezetés az üzleti informatikába*. Akadémiai Kiadó. Budapest.
- Katona E. (2004): *Bevezetés az informatikába*. Panem Kiadó. Budapest.
- Kónya L. (2001): *Számítógép-hálózatok*. LSI Oktatóközpont. Budapest.
- Kovács M., Knapp G., Ágoston G., Budai A. (1999): *Bevezetés a számítástechnikába*. LSI Oktatóközpont. Budapest.
- Kovács T., Ozsváth M., Nagy J., Kovácsné C. J. (2001): *Mit kell tudni a PC-ről? Az OKJ és ECDL vizsgákhoz*. Computerbooks Kiadó. Budapest
- Kőfalvi T. (2006): *Informatikai alapismeretek a tanári mesterségre készülők számára*. Nemzeti Tankönyvkiadó. Budapest.
- Kurose, J., Ross, K. (2009): *Számítógép-hálózatok működése*. Panem Kiadó. Budapest.
- Microsoft Press. (1999): *Számítógépszótár. Számítástechnikai fogalmak értelmezése*. Szak Kiadó. Bicske.
- MPC Standards. (1997) <<https://web.archive.org/web/19970109105038/http://www.spa.org/mpc/stand.htm>>
- Nemzeti Jogszabálytár (2019): 1999. évi LXXVI. törvény a szerzői jogról. <http://njt.hu/cgi_bin/njt_doc.cgi?docid=41066.366944>

- Nemzeti Jogszabálytár (2019): 2019. évi C. törvény a Büntető Törvénykönyvről.
<http://njt.hu/cgi_bin/njt_doc.cgi?docid=152383.243474>
- Pétery K. (2011): *Informatika II.* Műszaki Könyvkiadó Kft. Budapest.
- Raffai M. (2006): *Az információ: szerep, hatás, menedzsment.* Palatia Nyomda és Kiadó. Budapest.
- Sikos L. (2007): *PC hardver kézikönyv.* BBS-INFO Kft. Budapest.
- Sulinet. (2019): Algoritmusok szöveges, rajzos megfogalmazása. <<https://tudasbazis.sulinet.hu/hu/informatika/informatika/informatika-5-evfolyam/problemak-megfogalmazasa-felvetese/algoritmusok-szoveges-rajzos-megfogalmazasa>>
- Szabó B. (2013): *Adatbázis fejlesztés és üzemeltetés I.* Eszterházy Károly Főiskola. Eger.
- Tanenbaum, A. (2006): *Számítógép-architektúrák.* Panem Kiadó. Budapest.
- Tanenbaum, A., Woodhull, A. (2007): *Operációs rendszerek. Tervezés és implementáció.* Panem Kiadó. Budapest.
- Tímár L., Víg K., Tátrai J., Szigeti J., Vathy Á., Telekesi É., Vass I., Kocsis T., Priskinné R. Zs., Erdélyiné M. (19997): *Építsünk könnyen és lassan adatmodellt!* Veszprémi Egyetem & Műszertechnika-Veszprém Kft. Veszprém.
- Ullman, J., Widom, J. (2008): *Adatbázisrendszerek – Alapvetés.* Panem Kiadó. Budapest.
- Unicode Consortium (2019a): Unicode. <<http://www.unicode.org/>>
- Unicode Consortium (2019b): Unicode karakter tábla. <<https://unicode-table.com/hu/>>
- Wikipedia 1 (2019): Számítástechnika. <<https://hu.wikipedia.org/wiki/Sz%C3%A1m%C3%ADt%C3%A1stechnika>>
- Wikipedia 2 (2019): ASCII. <https://hu.wikipedia.org/wiki/ASCII#/media/File:ASCII_Code_Chart.svg>
- Wikipedia 3 (2018): Neumann-architektúra. <https://hu.wikipedia.org/wiki/Neumann-architekt%C3%BAr#/media/File:Von_Neumann_architecture.svg>
- Wikipedia 4.(2019): Logikai kapu. <https://hu.wikipedia.org/wiki/Logikai_kapu>
- Wikipedia 5 (2019): Informatika. <<https://hu.wikipedia.org/wiki/Port%C3%A1l:Informatika>>
- Wikipedia 6 (2018): Operációs rendszerek. <https://hu.wikipedia.org/wiki/Oper%C3%A1ci%C3%B3s_rendszer>
- Z. Karvalics L. (2004): *Bevezetés az információtörténelembe.* Gondolat-Infonia Kiadó. Budapest.