

1. Szoftver alapismeretek

A számítógép működését két fő tényező határozza meg:

- **Hardver** – a fizikai eszközök (processzor, memória, háttértár, perifériák).
- **Szoftver** – a programok és adatok összessége, amelyek a hardvert működtetik.

A szoftver feladata: a hardver erőforrásainak vezérlése és a felhasználó számára értelmezhető szolgáltatások biztosítása. A számítógép használhatóságát nagymértékben a szoftver határozza meg.[1]

flowchart LR HW[Hardver] --> OS[Operációs rendszer] OS --> APP[Alkalmazások] APP --> USER[Felhasználó] USER --> APP APP --> OS OS --> HW

1.1 A szoftver fogalma

- A **szoftver** olyan programok és adatok összessége, amelyek a hardveren futnak és konkrét feladatokat látnak el.
- A szoftver mindig a hardverhez kötődik, de ugyanaz a szoftver különböző hardverplatformokon is megjelenhet (pl. Windows, Linux, macOS).
- A szoftver jellemzői:
 1. nem kézzel fogható, hanem logikai entitás,
 2. fejlesztők készítik (programozás útján),
 3. a futtatáshoz megfelelő hardver és operációs rendszer szükséges.

1.2 A szoftverek fő csoportjai

- **Rendszerszoftverek** – az alapvető működést biztosítják (pl. operációs rendszer, driverek).
- **Alkalmazói szoftverek** – konkrét felhasználói feladatok elvégzésére szolgálnak (pl. CAD, szövegszerkesztő).
- **Fejlesztői szoftverek** – programok létrehozására szolgálnak (pl. fordítók, IDE-k).

graph TD SZOFT[Szoftver] --> REND[Rendszerszoftver] SZOFT --> ALK[Alkalmazói szoftver] SZOFT --> FEJL[Fejlesztői szoftver]

1.3 Szoftver és hardver kapcsolata

- A hardver önmagában nem használható → szükséges a szoftver.
- A szoftver a hardver képességeit „láthatóvá” és felhasználhatóvá teszi.
- A felhasználó → alkalmazói szoftveren keresztül → rendszerszoftveren keresztül → hardvert használja.

flowchart LR USER[Felhasználó] --> APP[Alkalmazói szoftver] APP --> OS[Rendszerszoftver] OS --> HW[Hardver]

1.4 Példák a gyakorlatból

- **Rendszerszoftver:** Windows 11, Linux Ubuntu, macOS.
- **Alkalmazói szoftver:** AutoCAD (mérnöki tervezés), MATLAB (számítás), Word/Excel (irodai).
- **Fejlesztői szoftver:** Visual Studio Code, GCC, Python interpreter.

2. Rendszerszoftverek, operációs rendszerek

A rendszerszoftverek azok a programok, amelyek a számítógép alapvető működését biztosítják, összekötik a hardvert a felhasználóval, és környezetet nyújtanak az alkalmazások futtatásához.[2]

2.1 A rendszerszoftver fogalma

- A rendszerszoftver olyan program, amely a számítógép hardverét vezérli, kezeli az erőforrásokat és lehetővé teszi az alkalmazások futtatását.
- Fő összetevői:
 1. **Operációs rendszer (OS)** – központi vezérlő program.
 2. **Eszközmeghajtók (driverekek)** – perifériák működéséhez szükséges szoftverek.
 3. **Segédprogramok (utility programok)** – karbantartást, diagnosztikát biztosítanak.

flowchart TD
HW[Hardver] --> OS[Operációs rendszer]
OS --> DRIVER[Eszközmeghajtók]
OS --> UTIL[Segédprogramok]
OS --> APP[Alkalmazások]
APP --> USER[Felhasználó]

2.2 Az operációs rendszer feladatai

- Hardver erőforrások kezelése (CPU, memória, háttértár, perifériák).
- Folyamatkezelés (programok párhuzamos futtatása).
- Fájrendszer kezelése (adatok tárolása és elérése).
- Felhasználói felület biztosítása (grafikus GUI, parancssor).
- Hálózati kommunikáció és biztonsági funkciók.

2.3 Windows 11 bemutatása

- **Megjelenés:** 2021-ben adták ki, a Windows 10 utódjaként.
- **Felhasználói felület:**
 1. Új, középre rendezett **Start menü** és **tálca**.
 2. Lekerekített ablakok, modern ikonok.
 3. **Widgetek** integrálása (időjárás, hírek, naptár).
- **Beállítások:**
 1. Egyszerűsített **Settings** alkalmazás.
 2. Gyors keresési lehetőségek.
 3. Jobb támogatás több monitorhoz és érintőkijelzőkhöz.
- **Újdonságok:**
 1. Beépített Microsoft Teams integráció.

2. **Android alkalmazások futtatásának támogatása** (Amazon Appstore, WSA).
3. Javított játékfunkciók: DirectStorage, AutoHDR.
4. Fokozott biztonsági követelmények: TPM 2.0, Secure Boot.

graph TD
UI[Felhasználói felület] --> START[Start menü és tálca] UI --> WID[Widgetek] UI --> MULTI[Többmonitor támogatás] FUNC[Újdonságok] --> TEAMS[Teams integráció] FUNC --> ANDR[Android appok] FUNC --> GAME[Játékfunkciók] SEC[Biztonság] --> TPM[TPM 2.0] SEC --> SB[Secure Boot]

2.4 Operációs rendszerek típusai

- **Windows** - a legelterjedtebb asztali OS.
- **Linux disztribúciók** - nyílt forrású, stabil, sokféle verzió (Ubuntu, Debian, Fedora).
- **macOS** - Apple saját rendszere.
- **Ipari rendszerek** - valós idejű operációs rendszerek (RTOS) ipari és beágyazott eszközökben.

2.5 Példák mérnöki gyakorlatból

- **Windows 11:** mérnöki alkalmazások (AutoCAD, SolidWorks, MATLAB).
- **Linux:** szerverek, HPC klaszterek (FEM, CFD szimulációk).
- **RTOS:** beágyazott mérnöki vezérlők, PLC-k.

3. Alkalmazói szoftverek

Az alkalmazói szoftverek a felhasználók mindennapi munkáját segítik: ezek azok a programok, amelyek konkrét feladatokat oldanak meg a számítógépen. Ide tartoznak az irodai programcsomagok, mérnöki és tudományos szoftverek, valamint a szórakoztatásra szolgáló alkalmazások is.[3]

3.1 Az alkalmazói szoftver fogalma

- Olyan program, amely a felhasználói igények kiszolgálására készült.
- Az alkalmazói szoftver mindig az operációs rendszer fölött működik.
- Példák:
 1. Szövegszerkesztő → dokumentumok létrehozása.
 2. CAD program → mérnöki tervezés.
 3. Adatbáziskezelő → adatok tárolása és lekérdezése.

flowchart TD
OS[Operációs rendszer] --> APP[Alkalmazói szoftver] APP --> USER[Felhasználó] USER --> APP

3.2 Alkalmazói szoftverek típusai

- **Irodai programcsomagok** - szövegszerkesztő, táblázatkezelő, prezentációkészítő.
 1. Példa: Microsoft Office 365, LibreOffice, Google Workspace.
- **Grafikai és multimédia szoftverek** - képszerkesztés, videóvágás, 3D modellezés.

1. Példa: Photoshop, Blender, Premiere Pro.
- **Mérnöki és tudományos szoftverek** – tervezés, szimuláció, adatfeldolgozás.
 1. Példa: AutoCAD, SolidWorks, MATLAB, ANSYS.
 - **Adatbázis-kezelő rendszerek** – nagy adathalmazok tárolása, lekérdezése.
 1. Példa: Oracle, MySQL, PostgreSQL.
 - **Szórakoztató és speciális alkalmazások** – játékok, tanulást segítő alkalmazások.

graph LR
APP[Alkalmazói szoftver] --> OFF[Irodai] APP --> GFX[Grafikai] APP --> ENG[Mérnöki/Tudományos] APP --> DB[Adatbázis] APP --> GAME[Szórakozás]

3.3 Szoftverlicenck és terjesztés

- **Kereskedelmi szoftver** – fizetős, zárt forráskódú.
- **Nyílt forráskódú szoftver (Open Source)** – ingyenes, módosítható.
- **Szabad szoftver** – felhasználó szabadon tanulmányozhatja, módosíthatja és terjesztheti.
- **SaaS (Software as a Service)** – felhőalapú szolgáltatás, előfizetéses modell.

graph TD
LIC[Szoftverlicenck] --> COM[Kereskedelmi] LIC --> OSS[Nyílt forráskódú] LIC --> FREE[Szabad szoftver] LIC --> SAAS[SaaS - előfizetés]

3.4 Példák mérnöki alkalmazásokra

- **Gépészmérnökök:** AutoCAD, SolidWorks, CATIA.
- **Villamosmérnökök:** MATLAB, LabVIEW.
- **Anyagmérnökök:** ANSYS, COMSOL.
- **Mechatronikus mérnökök:** robotikai szoftverek, PLC programozókönyezetek.

4. Fejlesztői szoftverek

A fejlesztői szoftverek olyan programok, amelyek más szoftverek létrehozására szolgálnak. Segítségükkel a programozók, mérnökök és kutatók új alkalmazásokat, vezérlőprogramokat, szimulációs eszközöket készíthetnek.[4]

4.1 A fejlesztői szoftver fogalma

- Célja: programok írása, fordítása, tesztelése és hibakeresése.
- A fejlesztői környezet tartalmazhat:
 1. **Fordítóprogramokat (compiler)** → magas szintű nyelvből gépi kód.
 2. **Értelmezőket (interpreter)** → utasítások közvetlen végrehajtása.
 3. **Linker** → összekapcsolja a külön modulokat.
 4. **Debugger** → hibakereső eszköz.
 5. **IDE (Integrated Development Environment)** → integrált fejlesztőkörnyezet.

graph TD
SRC[Kódforrás] --> COMP[Fordító] SRC --> INT[Értelmező] COMP --> BIN[Binaris program]

INT --> RUN[Futtatás] BIN --> DEBUG[Debugger] RUN --> DEBUG

4.2 Programozási nyelvek

- **Alacsony szintű nyelvek** - gépközeli (Assembly, C).
- **Magas szintű nyelvek** - könnyebben érthető, hordozható (C++, Java, Python).
- **Domain-specifikus nyelvek** - speciális célra (SQL, MATLAB, VHDL).
- **Modern trendek:**
 1. Python → gépi tanulás, adatfeldolgozás.
 2. Rust → biztonságos rendszerprogramozás.
 3. JavaScript/TypeScript → webes fejlesztés.

4.3 Fejlesztői környezetek és eszközök

- **IDE-k:**
 1. Visual Studio, Eclipse, PyCharm, IntelliJ IDEA.
- **Verziókezelő rendszerek:**
 1. Git, GitHub, GitLab.
- **Konténerizáció és DevOps:**
 1. Docker, Kubernetes, CI/CD rendszerek.

flowchart LR
DEV[Fejlesztő] --> IDE[IDE]
IDE --> GIT[Verziókezelő]
GIT --> CI[CI/CD pipeline]
CI --> DOCKER[Konténerizáció]
DOCKER --> DEPLOY[Telepítés]

4.4 Példák mérnöki alkalmazásokra

- **Beágyazott rendszerek** fejlesztése: C, C++, RTOS környezet.
- **Számítástudomány:** Python + MATLAB a mérnöki számításokhoz.
- **Ipari vezérlés:** PLC programozó környezetek (TIA Portal, Codesys).
- **Szimulációk:** ANSYS APDL, COMSOL scriptnyelvek.

From:

<https://edu.iit.uni-miskolc.hu/> - Institute of Information Science - University of Miskolc

Permanent link:

https://edu.iit.uni-miskolc.hu/tanszek:oktatas:szamitastechnika:szoftver_alapismeretek?rev=1758050320

Last update: 2025/09/16 19:18

