

Operációs rendszerek az iskolában

Dr. Nagy Csaba

2016. szeptember 14.

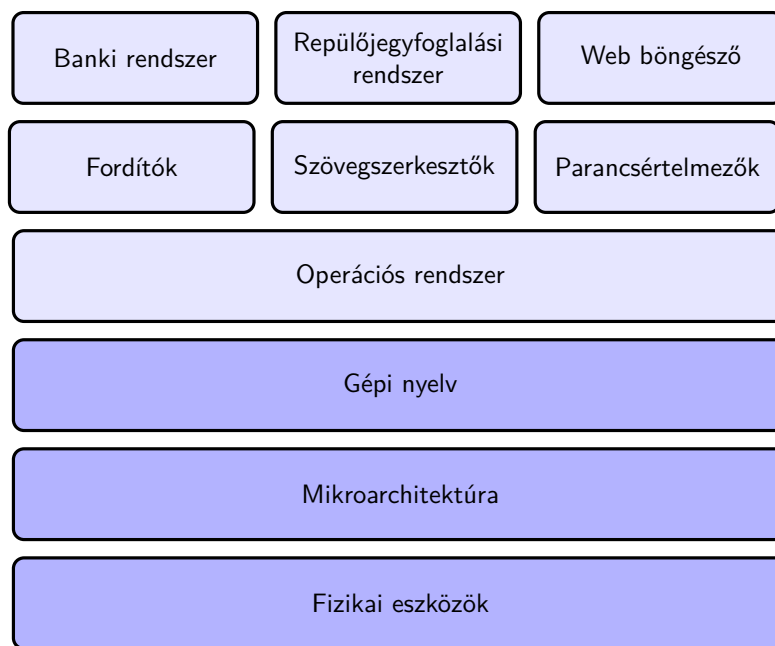
Tartalomjegyzék

1. Operációs rendszerek története	2
1.1. Mi az operációs rendszer?	2
1.2. Operációs rendszerek története	3
1.2.1. Első digitális számítógép	3
1.2.2. Első generációs gépek (1945-1955)	4
1.2.3. Második generációs gépek (1955-1965)	7
1.2.4. Harmadik generációs gépek (1965-1980)	8
1.2.5. Negyedik generációs gépek (1980-)	8
1.3. Operációs rendszer fogalmi	9
1.4. Operációs rendszer struktúrája	10
1.4.1. Monolitikus rendszerek	10
1.4.2. Rétegelt rendszerek	10
1.4.3. Exokernel	11
1.4.4. Virtuális gépek	11
1.4.5. Kliens-szerver modell	12
1.5. Irodalom	12

1. fejezet

Operációs rendszerek története

1.1. Mi az operációs rendszer?



1.1. ábra. Operációs rendszer helyzete az architektúrában

Az operációs rendszer (OS, Operating System) kettős feladatot lát el:

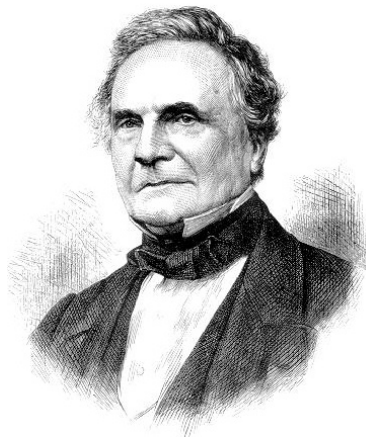
- *Kiterjesztett gép:* A felhasználónak olyan egyenértékű gépet nyújtson, amit egyszerűbb programozni, mint a mögöttes hardwaret.
- *Erőforrás-kezelő:* Hatékonyra, biztonságossá teszi a géphasználatot.

Az ISO nemzetközi szabványosítási szervezet definíciója szerint az operációs rendszer „*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.*”

1.2. Operációs rendszerek története

1.2.1. Első digitális számítógép

Charles Babbage (1792-1871), angol matematikus által tervezett gépek mechanikusan működő részekből álltak, amiket csak megtervezett, de sohasem fejezett be. Két ismert gépe az differenciálgép (Difference Engine, 1.3. ábra) és az analitikus (v. analitikai) gép (Analytical Engine, 1.4. ábra). A differenciálgépet 1991-ben az eredeti tervei alapján összeállították, ami tökéletesen működött. Gépeinek másolatát az Londoni Tudományos Múzeumban állították ki.

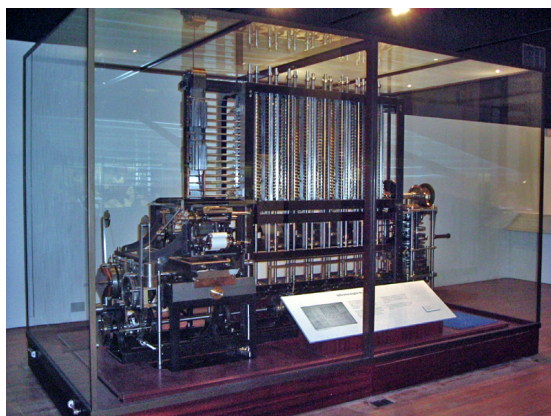


1.2. ábra. Charles Babbage, 1791-1871 (forrás: Wikipédia)



1.3. ábra. A differenciálgép egy része, amit a Babbage műhelyében talált darabokból raktak össze (forrás: Wikipédia)

Babbage differenciálgépe egy sor oszlopból állt, ahol minden oszlop egy tízes számrendszerbeli számot tárolt. A gép egyetlen műveletre volt képes: egy



1.4. ábra. Az analitikus gép a Londoni Tudományos Múzeumban kiállítva (forrás: Wikipédia)

oszlop értékét az előző oszlopéhoz hozzáadni. Az utolsó oszlop egy konstanst tudott tárolni, az első oszlop jelezte (vagy nyomtatta) ki az eredményt. (Forrás: Wikipédia)

Az analitikai gép képes volt programokat tárolni, és volt memóriája is, mely címezhető volt. Képes volt a programokon belül feltételes elágazásokat használni. A gép nevezhető volt a számítógépek elődjének.

Az analitikai gép első programozója Ada Lovelace (Lord Byron lánya) volt (1.5. ábra), aki 1842-ben fordította Luigi Menabrea olasz matematikus írását Babbage analitikai gépének a tervéről. A fordításhoz jegyzeteket fűzött, amelyben részletesen leírta, hogyan lehet Bernoulli-számokat számítani a gép segítségével. Ahogy, a Wikipédia a programról írja: *„Történészek szerint ez a világ legelső számítógépes programja, életrajzírók szerint azonban a programokat maga Babbage írta, Ada pedig mindössze egy hibát talált bennük, és visszaküldte javításra. A bizonyítékok, valamint Babbage és Ada levelezése azt mutatja, hogy a programokat Babbage írta, de Ada írásaiban előfordul a gép használatának néhány olyan lehetősége, amelyet Babbage sosem tett közzé, például hogy a gép képes lehet bármilyen összetettségű zenedarabok komponálására.”*

1.2.2. Első generációs gépek (1945-1955)

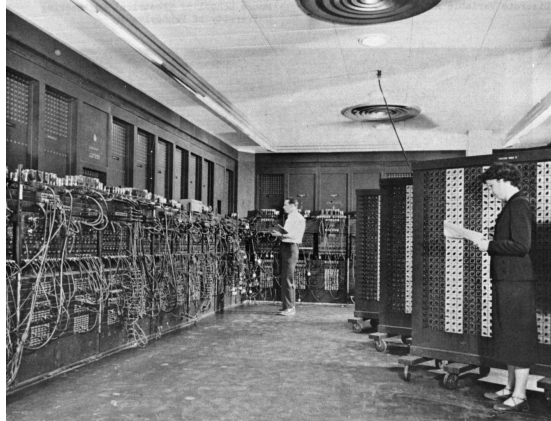
- Vákuumcsövek, kapcsolótáblák.
- Nincs munkamegosztás: egybeesik a számítógép építő, működtető, programozó, felhasználó szerepkör.
- Howard Aiken (Harvard), Neumann János (Inst. For Advanced Study, Princeton), J. Presper Eckert és John Williams Mauchley (Un. Pennsylvania), Konrad Zuse (Németo.)
- Gépi nyelvű programozás
- Nincs operációs rendszer
- 1950-es évek eleje, lyukkártyák



1.5. ábra. Augusta Ada Byron, 1815-1852 (forrás: Wikipédia)

- Az első programozható, elektronikus, digitális számítógép: ENIAC (Electronic Numerical Integrator And Computer); Tíz-es számrendszerben működött. A négy alapművelet mellett háromdimenziós, másodfokú differenciálegyenletek kiszámítására is képes volt. Az amerikai hadseregnek fejlesztették bonyolult katonai (pl. bombázási, tüzérségi) feladatok számításainak az elvégzésére. Építését 1943-ban a Pennsylvániai Egyetemen kezdték; 1946. február 14-én készült el majd 1947-ben átszállították a marylandi Aberdeen Proving Ground telephelyére, és 1947. július 29-től folyamatosan itt működött 1955. október 2-ig.
 - 30,5 méter hosszú
 - 1 méter széles
 - 3 méter magas
 - 140 kW teljesítmény felvétel
 - órajel: 100 kHz
 - 17 468 elektroncső
 - 7200 kristálydióda
 - 1500 jelfogó
 - 70 000 ellenállás
 - 10 000 kondenzátor
 - 6000 kapcsoló
 - 200 mikroszekundumos összeadási sebesség
 - 3 milliszekundumos szorzási sebesség
 - 30 milliszekundumos osztási sebesség

- Az ENIAC-ot követte az EDVAC (Electronic Discrete Variable Automatic Computer), amit Neumann János elvei alapján és közreműködésével készítettek el. Az első, belső programvezérlésű, elektronikus, digitális, univerzális számítógép. Bináris számábrázolást és aritmetikát használ. J. Presper Eckert és John Williams Mauchley vezetésével épült 1944 és 1948 között, 1951-ben helyezték üzembe.



1.6. ábra. ENIAC (forrás: Wikipédia)



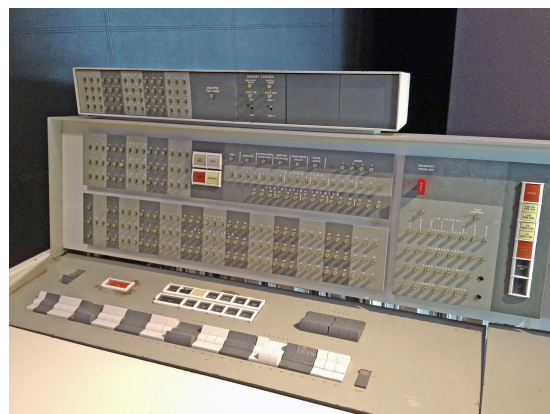
1.7. ábra. EDVAC (forrás: Wikipédia)

1.2.3. Második generációs gépek (1955-1965)

- Tranzisztorok és kötegelt rendszerek
- Karakterorientált (1401) és szószervezésű (7094) gépek.
- Tervezők és építők; karbantartók; operátorok; programozók, felhasználók
- FORTRAN nyelv
- A JOB fogalom: load-translate-load-exec szekvencia, eleinte „kézzel”, később „automatizálva”
- Tipikus a kötegelt (batch) feldolgozás



1.8. ábra. IBM 1401 vezérlő panelje (forrás: Wikipédia)



1.9. ábra. IBM 7094 vezérlő konzolja (forrás: Wikipédia)

1.2.4. Harmadik generációs gépek (1965-1980)

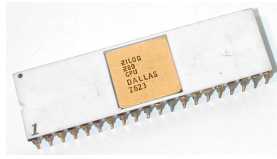
- Integrált áramkörök, multiprogramozás
- Interaktivitás igény: kifejlődnek a parancsnyelvi értelmezők
- „Kis” gépek (PDP-11, VAX 780 stb.)
- Sok operációs rendszer (OS360, MULTICS, RSX, Unix stb.)
- Hardveresek; operátorok; rendszerprogramozók; programozók; felhasználók
- Imperatív, funkcionális, logikai nyelvek. Szoftverkrízis (Algol, PL1, APL, LISP, PROLOG, C stb.)
- Unix; System V, BSD; POSIX



1.10. ábra. IBM System/360 a Volkswagen-nél (forrás: Wikipédia)

1.2.5. Negyedik generációs gépek (1980-)

- Személyi számítógépek
- Személyi számítógépek, munkaállomások,
- Szuperszámítógépek
- Visszaesés a védelemben
- Interaktivitás, felhasználóbarát kapcsolattartás. GUI is!
- MS DOS, Unix, VAX/VMS, Novell stb.
- Hálózati operációs rendszerek, osztott feldolgozás
 - Teljes a munkamegosztás
 - Objektumorientált programozás, párhuzamos programozás,
- szoftvertechnológiák, CASE



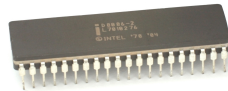
1.11. ábra. Zilog Z80 (forrás: Wikipédia)



1.12. ábra. Motorola 6800 (forrás: Wikipédia)



1.13. ábra. IBM PowerPC 601 (forrás: Wikipédia)



1.14. ábra. Intel 8086 (forrás: Wikipédia)



1.15. ábra. Apple II (forrás: Wikipédia)

1.3. Operációs rendszer fogalmai

- Rendszerhívások

- Processusok
 - Végrehajtás alatt lévő program
 - Saját címtartomány, regiszterkészlet
 - Processzushierarchia
- Fájlok
 - I/O egységek különlegességeinek elrejtése
 - Fájrendszer, jogosultságok, stb.
- Parancsértelmező (shell)

1.4. Operációs rendszer struktúrája

1.4.1. Monolitikus rendszerek

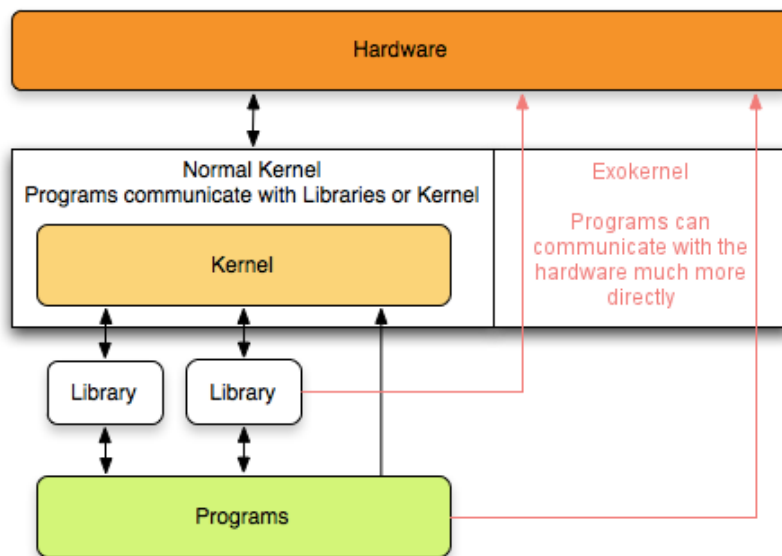
- Egyszerű szerkezet
- A kernel (mag): egy betölthető modulba összekapcsolt, szolgáltató rutinok összessége
- A kernel a rendszer indításkor (teljes egészében) betöltődik
- A szolgáltató rutinok hívása: rendszerhívással (system call), megszakítással (IT), vagy kivételes eseményre való reagálással.
- Futási mód váltás (user mód - kernel mód)
- Tipikus példa: korai Unix-ok

1.4.2. Rétegelt rendszerek

- A szolgáltató rutinok rétegekbe szervezettek.
 - A rétegezés előnye
 - * Egy rétegben jól meghatározott, összetartozó funkciók
 - * Egy réteg elrejt az alatta lévő komponensek részleteit
 - * Magasabb absztrakciós szintű szolgáltatásokat biztosít: virtuális gépet
 - A rétegeket külön-külön betölthető fájlokba linkelik
 - * A betöltés akár dinamikusan is történhet (DLL)
 - * Már a monolitikusban is lehet diszpécser réteg
 - Lehetséges: a rétegekhez egyre privilegizáltabb futási módok.
- Tipikus példa: The OS

Rétegek		
5	Operátor	Az operátor indít élő processzeket
4	Felhasználói programok	Független processzek
3	I/O menedzsmment	Felette virtuális (absztrakt) I/O
2	Operátor-processz kommunikáció	A processzeknek saját virtuális konzolok
1	Memória-dob menedzsmment	Virtuális memória, a processz látja a teljes címtartományát
0	Processz allokalás, multiprogramozás	Virtuális CPU-k, minden processztől elrejtve a többi

1.1. táblázat. THE OS. Dijkstra professzor és hallgatói készítették, 1968. A rétegezés tervezési koncepció volt: egyetlen modulba linkelték.



1.16. ábra. Exokernel (forrás: Wikipédia)

1.4.3. Exokernelek

- M.I.T, Engler, 1995.
- Alsó rétegen az exokernel közvetlenebb elérést biztosítva a hw-hez.

1.4.4. Virtuális gépek

- Szimulált számítógép: fizikailag nem létezik, egy szimuláció
- Fő fajtái
 - *Rendszer szintű virtualizáció*: komplett számítógépek szimulálására, operációs rendszerrel együtt.
 - *Folyamat szintű virtualizáció*: egyetlen program, egyetlen folyamat szimulálására. Pl.: JVM.

1.4.5. Kliens-szerver modell

- A kliens-szerver modell
- Igény a kisebb kernelre: mikrokernel
- Bizonyos rendszerszolgáltatások kiszolgálására önálló processzeket készítenek.
 - Ezek önállóan linkelhetők
 - Betölthetnek a rendszer egész életére, vagy időlegesen (daemon processzek)
 - Futhatnak kernel-, vagy akár felhasználói módban
- Processzek közti kommunikációval kérhetők a szolgáltatások. A kernel szinte csak a kommunikációt és az időosztást biztosítja.
- Ebből fejlődtek ki a hálózati osztott rendszerek.

1.5. Irodalom

- Andrew S. Tanenbaum - Albert S. Woodhull: Operációs rendszerek, 2.kiadás, Panem Kiadó, 2007. Bevezetés (15-68. oldal)