

### Császármorzsa

Keverj össze 25 dkg grízt 1 mokkás kanál sóval,  
4 evőkanál cukorral és egy csomag vaníliás  
cukorral!  
Adj hozzá két evőkanál olajat és két tojást, jól  
dolgozd el!  
Folyamatos keverés közben adj hozzá apránként  
fél liter tejet!  
Adj hozzá egy bögre előre beáztatott mazsolát!  
Süssd ki 3 evőkanál olajon!

**Ez egy program.  
De ki tudja végrehajtani?**

Máté: Architektúrák

1. előadás

1

A **digitális számítógép** olyan gép, amely a neki szóló  
**utasítások** alapján az emberek számára problémákat  
old meg.

Azt az utasítássorozatot, amely leírja, hogyan oldjunk  
meg egy feladatot, **programnak** nevezzük.

A legtöbb gépi utasítás ritkán bonyolultabb mint:

- Adj össze két számot!
- Ellenőrizd egy számot, vajon nulla-e!
- Egy adatot másolj a számítógép memóriájában egyik helyről a másikra!

Máté: Architektúrák

1. előadás

2

Egy számítógép utasításainak együttese egy olyan  
nyelvet alkot, amelyen az ember a számítógéppel  
képes kommunikálni. Az ilyen nyelvet **gépi  
nyelvnek** nevezzük.

Egyszerűbb gépi nyelv



Egyszerűbb elektronika



Olcsóbb gép



Az ember számára nehézkes

Máté: Architektúrák

1. előadás

3

Legyen  $L_0$  a gépi nyelv, és  $L_1$  egy az ember számára  
kényelmesebb nyelv.

Hogy hajtható végre az  $L_1$  nyelven írt program?

Kellene olyan gép, amelynek gépi nyelve az  $L_1$  nyelv.

Más megoldás:

**Fordítás és értelmezés**

Máté: Architektúrák

1. előadás

4

**Fordítás:** Először az  $L_1$  nyelvű program minden  
utasítását helyettesítjük az  $L_0$  nyelv utasításainak  
egy vele ekvivalens sorozatával. Az így nyert  
program teljes egészében az  $L_0$  utasításaiból áll.  
Ezután az eredeti  $L_1$  nyelvű program helyett a  
számítógép ezt az  $L_0$  nyelvű programot hajtja végre.

**Értelmezés:** Az  $L_1$  nyelvű program következő  
utasítását elemezzük, és a vele ekvivalens  $L_0$  nyelvű  
utasítássorozatot azonnal végrehajtjuk a  
számítógéppel.

Máté: Architektúrák

1. előadás

5

A fordítás és az értelmezés is elvégezhető  
az  $L_0$  nyelvű számítógéppel.

Olyan, mintha lenne olyan gépünk, amely végre tudja  
hajtani az  $L_1$  nyelven írt programot: **virtuális gép**.

A gépi és az ember számára kényelmes nyelv között  
oly nagy az eltérés, hogy annak áthidalásához  
nyelvek és virtuális számítógépek hierarchiája  
alakult ki.

**Strukturált számítógép-felépítés**

Máté: Architektúrák

1. előadás

6

n. szint  $L_n$  nyelv,  $M_n$  virtuális gép

Az  $L_n$  nyelvű programokat vagy egy alsóbb szinten futó értelmező hajtja végre, vagy egy alsóbb szinten futó fordítóprogram fordítja alsóbb szintre

n-1. szint  $L_{n-1}$  nyelv,  $M_{n-1}$  virtuális gép

...

0. szint  $L_0$  nyelv,  $M_0$  valódi gép

Végrehajtja az  $L_0$  szintű programot.

**1.1. ábra**

Máté: Architektúrák 1. előadás 7

**Számítógép architektúra (architecture)**  
**Strukturált felépítés**

**A felhasználónak látszó gép:** adattípusok, utasítások, szolgáltatások összessége.

**Számítógép-felépítés (organization):** a közvetlenül nem szükséges részekkel is foglalkozik.

**Számítógép – utasítás rendszer – programozás:** gépi nyelv (kód).

**Bonyolultabb nyelvek:** fordítás vagy értelmezés.

Máté: Architektúrák 1. előadás 8

**Gépi, nyelvi szintek (1.2. ábra)**

5. Probléma orientált nyelv szintje fordítás (fordító program)
4. Assembly nyelv szintje fordítás (assembler)
3. Operációs rendszer szintje részben értelmezés (operációs rendszer)
2. Gépi utasítás szintje ha van mikroprogram, akkor értelmezés
1. Mikroarchitektúra szintje hardver
0. Digitális logika szintje

Máté: Architektúrák 1. előadás 9

**Gépi, nyelvi szintek (1.2. ábra)**

- 0: **digitális logika** szintje: kapu (gate): **AND**, **OR**, ...  
→ 1 bites, → több bites memória, regiszter, ...
- 1: **mikroarchitektúra** szintje: mikROUTASÍTÁSOK, mikroprogram - nem minden gépen létezik, de a gépi utasítások végrehajtását gyakran mikroprogram végzi, ekkor ez a szint **értelmezi** a 2. szintet. Pl. a szorzást összeadásra és léptetésre vezeti vissza.
  - Regiszterek, aritmetikai-logikai egység - **ALU**
  - Adatfolyam - **adatút**
- 2: **gépi utasítás** szintje (tényleges gépi utasítások): itt dől el a kompatibilitás kérdése.

Máté: Architektúrák 1. előadás 10

- 3: **operációs rendszer** szintje: speciális kiegészítők (memóriakezelés, párhuzamos futtatás, ...).  
Általában **értelmezés**. A szint utasításait  
– az operációs rendszer  
– vagy közvetlenül a 2. szint hajtja végre

Az eddigi szintek programjai hosszú számsorozatok (természetesen ma már szimbolikusan készülnek)  
----- Eddig: rendszerprogramozók területe -----

- 4: **assembly nyelv** szintje, szimbolikus leírás
- 5: **probléma orientált nyelv** szintje: pascal, C, C++, ... , adatbázis kezelők, ...

Ezek tényleges nyelvek, fordítás

Máté: Architektúrák 1. előadás 11

**Gépi utasítás szintje**

Az utasítások a memóriában vannak tárolva.

addr) **command dest, source1, source2, next**

addr: az utasítás címe a memóriában

**command** (utasítás): az utasítás kódja

**dest** (cél): itt képződik az eredmény

**source1** (forrás1): a művelet 1. operandusa

**source2** (forrás2): a művelet 2. operandusa

**next**: a következő végrehajtandó utasítás címe. Ez legtöbbször az utasítás utáni első rekesz címe, ezért általában nem kell megadni (**implicit operandus**), csak akkor, ha más utasítással folytatódik a program (ugró utasítás).

Máté: Architektúrák 1. előadás 12

cím) **add dest, source1, source2**  
 hatására **dest** fölveszi a **source1 + source2** értéket.  
 Ilyenkor természetesen elvész **dest** régi értéke.

**További implicit operandusok:**  
 Sokszor egyszerűsítik az utasításokat, pl.:

cím) **add op1, op2**  
 hatására **op1** fölveszi az **op1 + op2** értéket. További egyszerűsítés:

cím) **add op**  
 hatására **A** fölveszi az **A + op** értéket, ahol **A** egy kitüntetett regiszter (**accumulator**).

Máté: Architektúrák 1. előadás 13

**Hardver, szoftver fejlődése**

Többszintű gépek kialakulása.

- Kezdetben két szint:
  - digitális logika,
  - utasítások.
- Mikroprogram (hardver bővítése programozással): Wilkes, 1951. Gyorsan elterjedt. Csúcs: hatvanas, hetvenes évek; nagyon sok új utasítás (\*, /, ..., ciklusszervezés, megszakítások) - később ezek az utasítások hardverrel is megvalósíthatókká váltak, és úgy gyorsabbak lettek.

Folyamatosan változó határok.

Máté: Architektúrák 1. előadás 14

**Operációs rendszerek**

A hatvanas években készültek először:

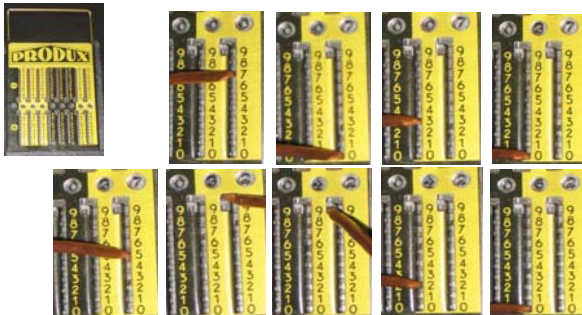
- supervisor, rendszerhívások,
- kötegelt (batch) feldolgozás,
- közvetlen telefonos összeköttetés remote terminálok, időosztás (timesharing).

Máté: Architektúrák 1. előadás 15

**Történelmi áttekintés (1.4. ábra)**

**0. generáció: Mechanikus gépek**


- **Pascal (1642):** összeadás, kivonás ( $37+25=62$ )



Máté: Architektúrák 1. előadás 16

**Elektronikus zsebszámológép**

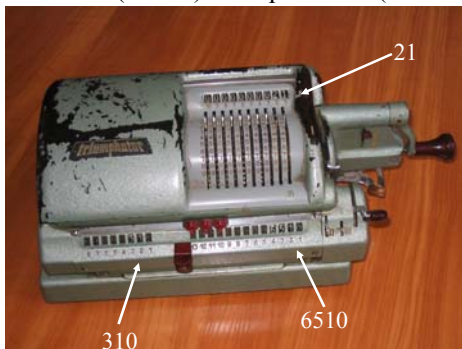
**37 + 25 = 62**



Hova lett a 37?

Máté: Architektúrák 1. előadás 17

- **Leibniz (~1700):** 4 alpművelet (szorzás, osztás is)



21  
 310  
 6510

triumphator

$21 * 310 = 6510$

Máté: Architektúrák 1. előadás 18



- Babbage (1834): differencia gép (csak egy programja volt) – különböző táblázatok készítésére volt alkalmas. Összeadás, kivonás. Kiírás fémlemeze (nyomtató a fémlemezről).

Máté: Architektúrák

1. előadás

19



A differenciagép egy része, amit a Babbage műhelyében talált darabokból raktak össze

Máté: Architektúrák

1. előadás

20



Máté: Architektúrák

1. előadás

21



Máté: Architektúrák

1. előadás

22



Máté: Architektúrák

1. előadás

23

Ada Augusta Lovelace



Ada Byron Lovelace

Máté: Architektúrák

1. előadás


24



- Babbage: analitikus gép (nem készült el).  
Programozható: utasítások lyukkártyáról  
Ada Augusta Lovelace.

Egységek:

- memória (1000 db 50 jegyű szám),
- malom (+, -, \*, /),
- input (lyukkártya),
- output.



Máté: Architektúrák 1. előadás 26

- Zuse, 1936: elektromágneses relék használata.
- Aiken (1944): Babbage nyomán: jelfogós gépet készített (Mark I, később Mark II)

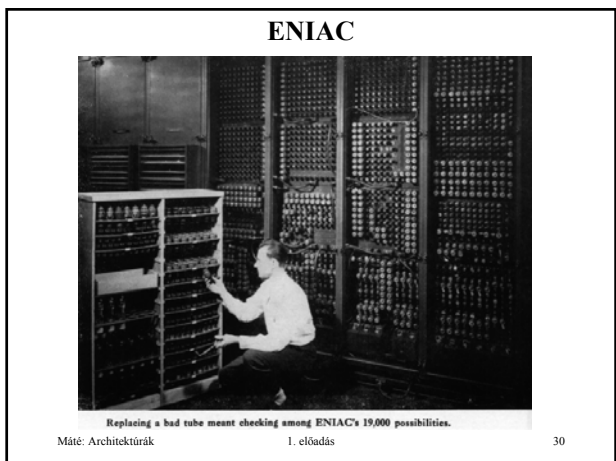
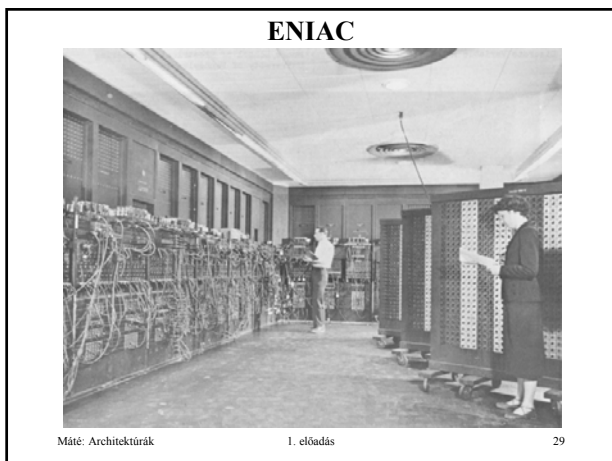
Máté: Architektúrák 1. előadás 27

**1. generáció:** elektroncső (1945-1955).

- **COLOSSUS (Turing, 1943):** titkosítások megfejtése - 30 évre titkosítva.
- **ENIAC (Electronic Numerical Integrator and Computer - Mauchley, Eckert, 1943):** 18000 cső, 140 KW, 30 tonna, 20 darab 10 decimális jegyes regiszter. 10 cső egy decimális számjegyhez! Dugaszolással programozható. 1946-ig nem sikerült befejezni.

• Nyári iskola - sok próbálkozás.

Máté: Architektúrák 1. előadás 28



**Neumann János** megismerte az **ENIAC**-ot, és új gépet tervezett (**IAS**): bináris aritmetika, tárolt program.

The diagram illustrates the Neumann architecture. At the top is a box labeled 'Memória'. Below it are three boxes: 'Vezérlő egység' (Control Unit), 'Aritmetikai-logikai egység' (Arithmetic-Logic Unit), and 'Akkumulátor' (Accumulator). To the right of the ALU are two boxes: 'Bemenet' (Input) and 'Kimenet' (Output). Arrows show data flow: from Memória to Vezérlő egység and ALU; from Vezérlő egység to ALU; from ALU to Memória; from ALU to Akkumulátor; from Akkumulátor to ALU; from Bemenet to ALU; and from ALU to Kimenet.

**1.5. ábra.** Az eredeti Neumann-gép

Máté: Architektúrák 1. előadás 31

**Neumann János**

Máté: Architektúrák 1. előadás 32

**Neumann elvű gép**

Tárolt programú elektronikus számítógép, amely az adatok és a program tárolására közös memóriát használ.

**Harvard típusú számítógép**

Külön memóriát használ az adatok és külön memóriát a program számára.

Máté: Architektúrák 1. előadás 33

**EDSAC (Wilkes, Cambridge, 1949)** az első Neumann elvű működő gép. 2-es számrendszer, 4096 szavas memória. 40 bites szavak: előjeles egész, vagy két utasítás. **Tárolt program**: 8 bites utasításkód, 12 bites cím. Akkumulátor. Nem volt lebegőpontos aritmetika!

**EDVAC (1949 Eckert és Mauchley)**, de elvérzett a projekt. Később ebből a vállalkozásból lett az **UNISYS**.

- Eckert és Mauchley sikertelenül próbálják találmánynak elfogadtatni.

Máté: Architektúrák 1. előadás 34

- IBM 701 (1953)** 2K 36 bites memória.
- Legnagyobb 1. generációs: **IBM 709 (1958)**.
- Szegeden: **M3 (1963)**.

Máté: Architektúrák 1. előadás 35

Máté: Architektúrák 1. előadás 36

**2. generáció:** tranzisztor (1955-1965).  
 Tranzisztor feltalálása: 1948, Nobel díj: 1956.

- Első jelentős tranzisztoros gép: **DEC PDP-1** (1961).  
 4K 18 bites szó,  
 200 000 utasítás/s,  
 512x512 display!

Teljesítménye fele az **IBM 7090**-nek (az **IBM 709** tranzisztoros változata)  
 Ára csak töredéke:  
 120 000 US\$. Több tucat eladott gép.

Máté: Architektúrák 1. előadás 37

- Néhány évvel később: **PDP-8**: omnibusz (**általános sín, 1.6. ábra**) – 15 000 US\$, 50 000 eladott gép.

**1.6. ábra.** A PDP-8 „omnibus”

- Magyarországon: KFKI, TPAi.
- IBM 7090, 7094.**

Máté: Architektúrák 1. előadás 38

- Első üzlet-orientált gép (1961): **IBM 1401** byte-szervezésű memóriával.
- Burroughs B5000** (1963),  
 Cél: hatékony ALGOL fordítót lehessen írni!
- Első super-gyors gép (1964): **CDC 6600**  
 Tervező: **Seymour Cray**. Párhuzamos működésre képes egységek, külön egység az összeadásra, szorzásra, osztásra.  
**Párhuzamos utasítás-végrehajtás.**

- Szegeden: **MINSZK 22.**

Máté: Architektúrák 1. előadás 39

**3. generáció:** integrált áramkör (1965-1980).  
**Robert Noyce** (1958): szilícium alapú integrált áramkör  
**IBM** problémája: **7094** és **1401** nem kompatibilis.

- (1964): **IBM 360**-as sorozat (**1.7 ábra**).  
 Később a szocialista országokban **R**-sorozat.

Tulajdonság	Model 30	Model 40	Model 50	Model 65
Relatív teljesítmény	1	3,5	10	21
Ciklus idő (ns)	1000	625	500	250
Maximális memória (KB)	64	256	256	512
Ciklusonként elérhető bájt	1	2	4	16
Adatsatornák max. száma	3	3	4	6

Máté: Architektúrák 1. előadás 40

Emuláció: a sorozat gépein futtathatók az előző **7094** és **1401** típusok programjai is. (mikroprogramozás).  
Multiprogramozás (DOS, POWER, OS).  
 24 bites címtartomány (16 Mbyte). A nyolcvanas évek közepéig elég, akkor áttérés a 32 bites címre.

- DEC: PDP-11** (1970): 16 bites  
 Magyarországon: **TPA70**. Nem kompatibilis a **PDP-11** -gyel.

----- Eddig csak számításigényes problémák -----

Máté: Architektúrák 1. előadás 41

**4. generáció: VLSI** (Very Large Scale Integration) (1980- ). Néhány millió elem egy lapkán (chipen).

- Személyi számítógépek.  
 Kezdetben zacskóban: nyomtatott áramkörtől, IC-k, köztük általában **INTEL 8080**, kábelek, tápegység, hajlékony lemez
- CP/M** operációsrendszer (Gary Kildall),
- Commodore, Apple, Spectrum-80.**
- IBM PC (I-8088)** alapú a terveket publikálták - klónok.
- Microsoft: **MS-DOS, OS/2, Windows.**
- ...

Máté: Architektúrák 1. előadás 42

### Technológiai fejlődés

- **Moore törvény (1965):**  
Az egy lapkán elhelyezhető elemek száma másfél évenként duplázódik (**1.8. ábra**). Azt várják, hogy 2020-ig teljesülni fog. Minden más területen (lemezek, adatátvitel, ...) hasonló sebességű a fejlődés.
- A szoftverek mérete, bonyolultsága is követi ezt:
- **Nathan első törvénye:**  
A szoftver gáz: kitölti a rendelkezésére álló teret.

Máté: Architektúrák 1. előadás 43

### Technológiai fejlődés

A népszerűsítő irodalom kedvenc hasonlata szerint, ha az autóipar az utóbbi hetven évben úgy haladt volna, mint a számítástechnika, egy Rolls-Royce-t 20 \$-ért lehetne kapni, motorja gyufafej nagyságú lenne, sebessége 100 000 km/h és egymillió kilométeren 3 liter benzint fogyasztana

Vámos Tibor  
1981

Máté: Architektúrák 1. előadás 44

Típus	Ár (US \$)	Felhasználható például
Eldobható	0.5	Üdvözlőlapok, <b>RFID</b> (Radio Frequency IDentification)
Mikrovezérlő	5	Órák, autók, eszközök
Játék	50	Videojátékok
Személyi számítógép	500	Asztali/hordozható
Szerver	5 000	Hálózati szerver
Munkaállomás-gyűjtemény (COW)	50 000-500 000	Tanszéki mini-szuperszámítógép
Nagyszámítógép	5 000 000	Időjárás előrejelzés...

**1.9. ábra.** A mai (2005) számítógép típusok választéka

Máté: Architektúrák 1. előadás 45

### Pentium 4. (1.11. ábra)

Lapka	Dátum	MHz	Tranz.	Mem.	Megjegyzés
I-4004	1971/4	0.108	2300	640	Első egylapkás mikroproc.
I-8008	1972/4	0.108	3500	16 KB	Első 8 bites mikroproc.
I-8080	1974/4	2	6000	64 KB	Első általános célú mikroproc.
I-8086	1978/6	5-10	29000	1 MB	Első 16 bites mikroproc.
I-8088	1979/6	5-8	29000	1 MB	Az IBM PC processzora
I-80286	1982/6	8-12	134000	16 MB	Memória védelem
I-80386	1985/10	16-33	275000	4 GB	Első 32 bites mikroproc.
I-80486	1989/4	25-100	1.2M	4 GB	8 KB beépített gyorsítótár
Pentium	1993/5	60-233	3.1M	4 GB	Két csövezeték, MMX
P. Pro	1995/3	150-200	5.5M	4 GB	Két szintű beépített gyorsítótár
P. II	1997/5	233-400	7.5M	4 GB	Pentium Pro + MMX
P. III	1999/2	650-1400	9.5M	4 GB	SSE utasítások 3D grafikához
P. 4	2000/11	1300-3800	42M	4 GB	Hyperthreading + több SSE

Máté: Architektúrák 1. előadás 46

### UltraSPARC III

Igény: UNIX-ot kisgépekre. Hálózati gépek: Ethernet.

- SUN (Stanford University Network - 1982).  
Motorola 68020 CPU alapú gépek.  
1987-ben félmilliárd \$ a bevételük.
- SPARC (Scalable Processor ARChitecture - 1987).  
32 bites, 36 MHz. Több cégnek átadták a gyártási jogot, verseny → gyors fejlődés!
- UltraSPARC I: 64 bites, multimédiás utasítások.
- UltraSPARC II, III: gyorsítás + kevés módosítás.
- UltraSPARC IV, kétprocesszoros UltraSPARC III.

Máté: Architektúrák 1. előadás 47

### 8051

Lapka	Program-memória	Memória típus	RAM	Időzítők	Megszakítások
8031	0 KB		128	2	5
8051	4 KB	ROM	128	2	5
8751	8 KB	EPROM	128	2	5
8032	0 KB		256	3	6
8052	8 KB	ROM	256	3	6
8752	8 KB	EPROM	256	3	6

**1.14. ábra.** Az MCS-51 család tagjai

Beágyazott rendszerekben használatos. Évente 8 milliárd mikrovezérlőt adnak el! Ez a család a legnépszerűbb! Nagyon olcsó (10-15 cent).

Máté: Architektúrák 1. előadás 48



**Feladatok**

Mi a különbség az értelmezés és a fordítás között?

Milyen gépi, nyelvi szinteket ismer?

Jellemezze a Neumann elvű számítógépet!

Jellemezze a Harvard típusú számítógépet!

Mi az előnye a Neumann elvű számítógépnek a Harvard típusúval szemben?

Jellemezze az egyes számítógép generációkat!

Valamikor 1 mikron átmérőjű volt a mikroprocesszor egy tranzisztora. Moore szabálya szerint mekkora volt az átmérője három évvel később?

**Az előadáshoz kapcsolódó**

**Fontosabb tételek**

Fordítás és értelmezés