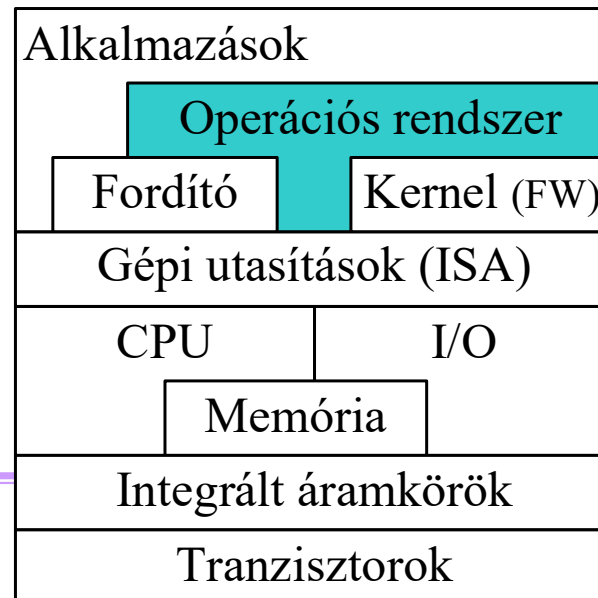


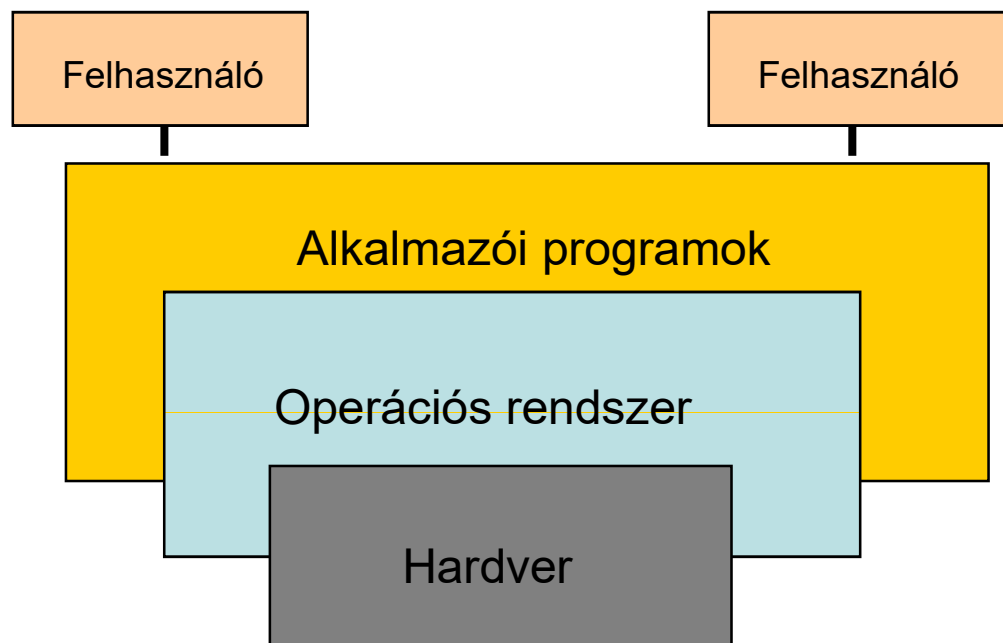
INFORMATIKA I.

BMEVIIIAB04

Operációs rendszerek



Számítógép rendszer



Operációs rendszer
kapcsolat a hardver és a
felhasználó között

Cél
Hatékony hardver kihasználás
A felhasználó kényelme

Operációs rendszer

Környezet a felhasználó számára

Program, amely a hardver erőforrásait kiosztja

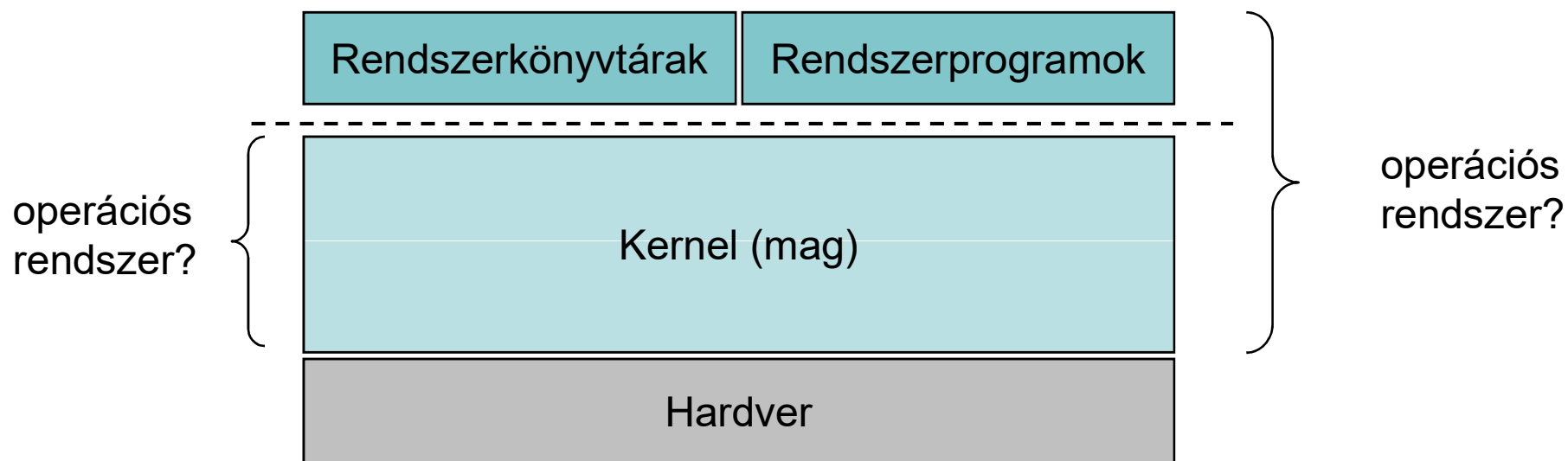
Program, amely vezérli a számítógép működését

Felhasználói programok közös műveleteinek gyűjteménye

...

Programok, amelyek vezérlik a hardvert és lehetővé teszik azon felhasználói alkalmazások végrehajtását

Operációs rendszer



Korai rendszerek

Nagy, drága hardver
Drága gépidő (nagy energiaigény)

Kézi üzemeltetés

Bevitel: kapcsolók

Eredmény: kezdetleges nyomtató

Egy felhasználó

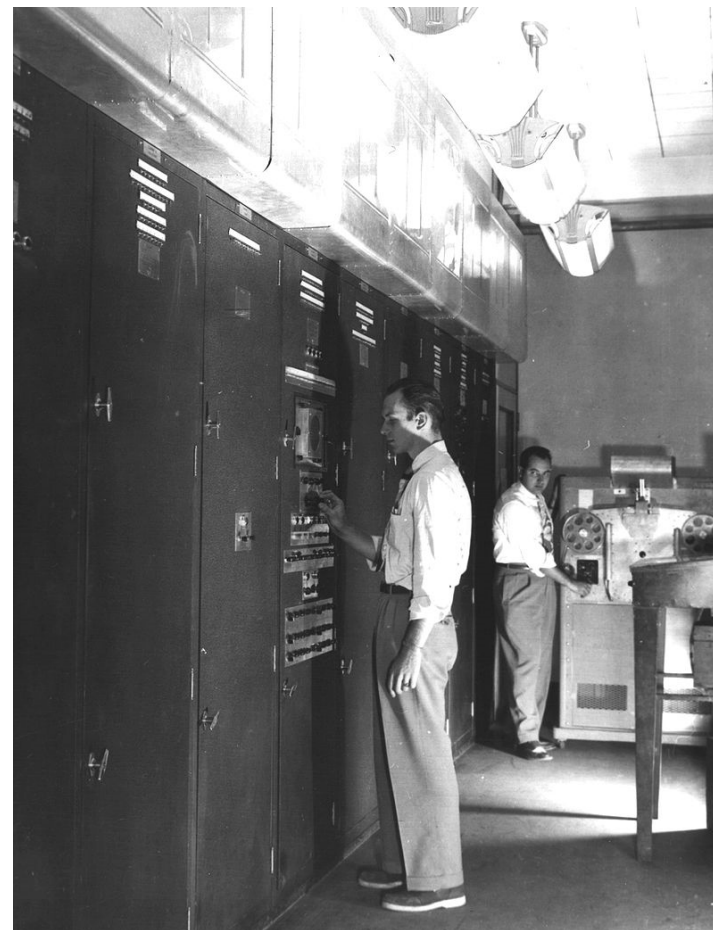
Programozó = operátor → OPEN-SHOP

A hibakeresés ideje megbecsülhetetlen

Egyszerre csak egy program fut

A teljes hardver kezelése a program feladata

A hardver csak az idő töredékében dolgozik



Hardver fejlődés

Gyorsabb CPU

Gyorsabb memória hozzáférés

Hatékonyabb perifériák

szalag/kártya lyukasztó

szalag/kártya olvasó

mágnesszalag

sornyomtató



Szoftver fejlődés

Assembler

Compiler

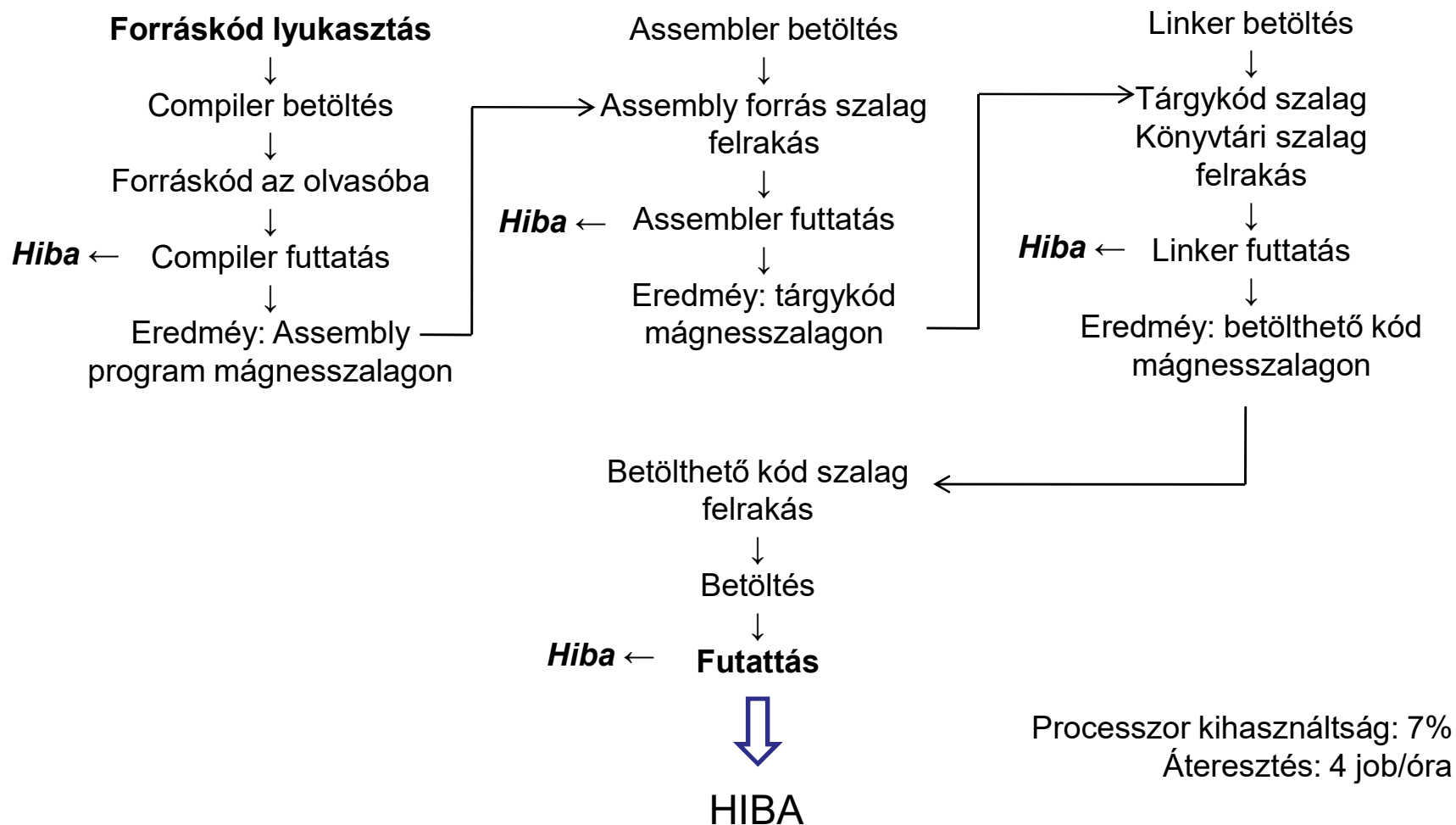
Linker

Loader

Programkönyvtárak

I/O műveletekhez → device driver

Programkészítés / futtatás tevékenységei



Növeljük a gépkihasználságot

Használjunk képzett segéderőt: operátor

→ CLOSED-SHOP

Programozó → adathordozó + leírás  **JOB**

Operátor → futtat

Mi van hiba esetén? → Dump

Tevékenység optimalizálás

Egy compiler betöltéssel több forrás is fordítható

Igény: automatikus átkapcsolás a következő műveletre

→ vezérlő kártya: utasítások

\$JOB – JOB kezdete

\$END – JOB vége

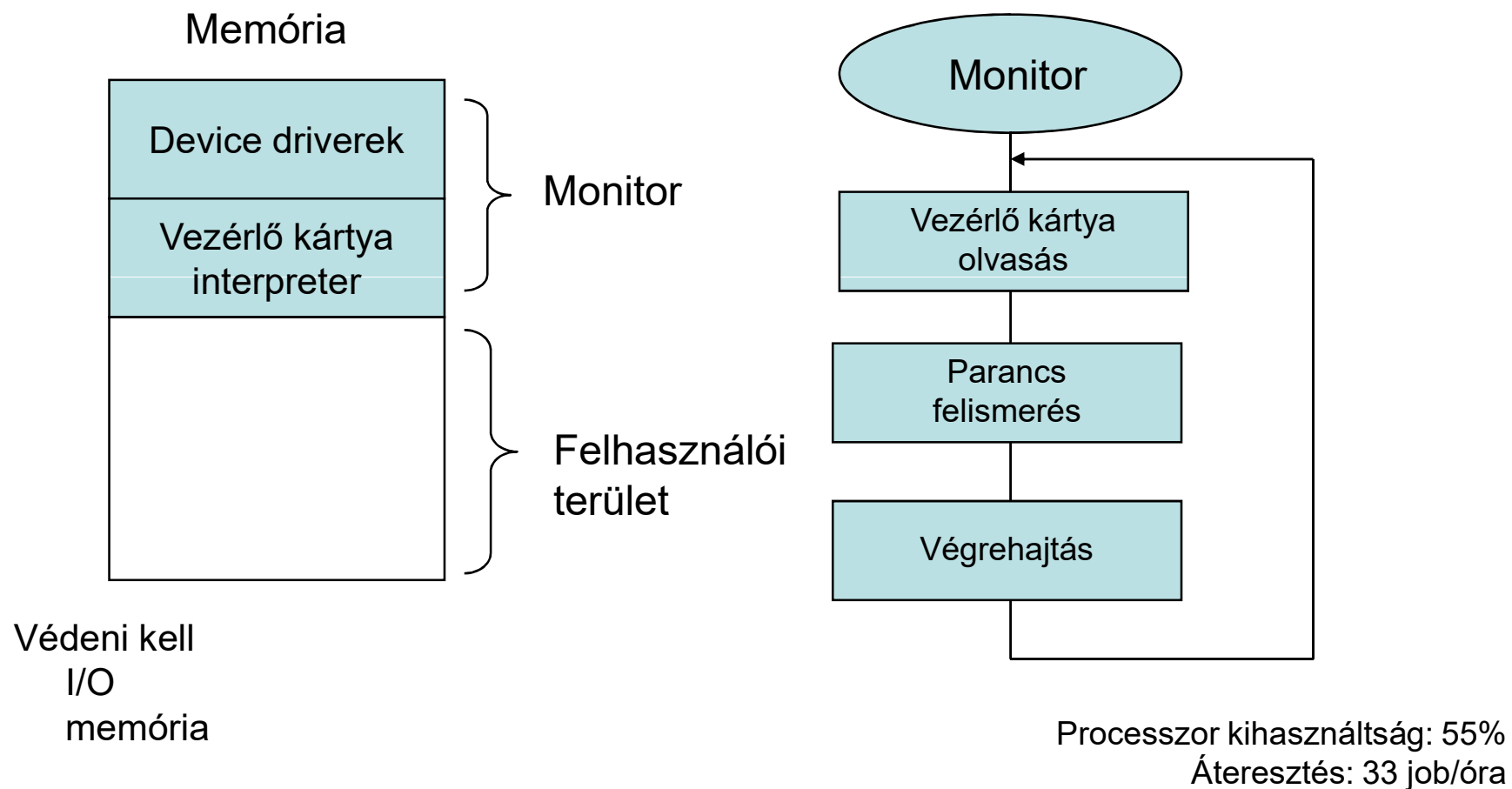
\$COMP – compiler végrehajtása

\$LOAD – program betöltés

\$RUN – futtatás

Ki értelmezza a vezérlő kártyákat?

Monitor: rezidens program, csak a gép indulásakor töltődik be (ROM?)



I/O védelem

Hogyan védjük ki azt, hogy egy program túlolvasson saját adatain?

→ Csak a monitor olvashat

Honnan tudjuk, hogy a monitor akar olvasni?

→ Két működési mód: USER – SUPERVISOR (SYSTEM)

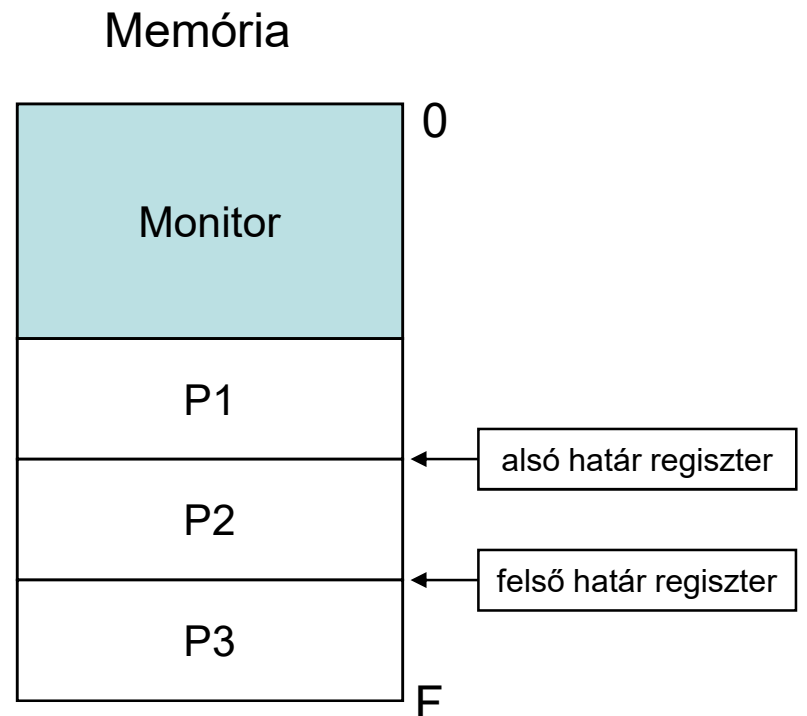
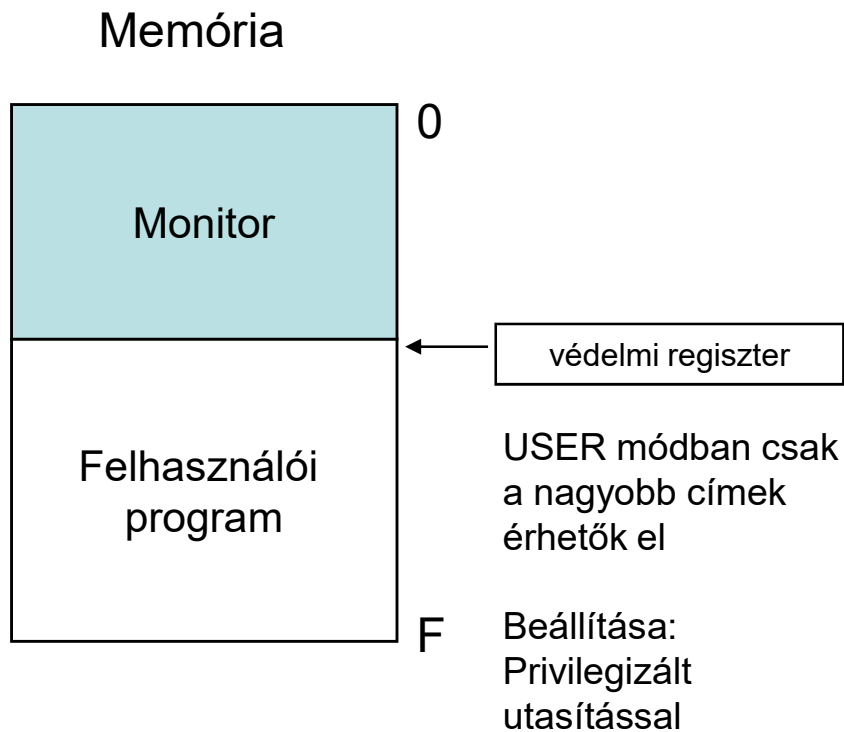
Privilegizált utasítások szükségesek

Hogyan tud egy felhasználói program olvasni?

→ I/O műveletek rendszerhívások

A rendszerhívás működési módot vált

Memória védelem



USER módban csak a két regiszter közötti címek érhetők el

Szűk keresztmetszetek

Ember ~ sec

Mechanikus periféria ~ msec

Elektronika ~ μ sec

← helyettesíthető automatizmussal

Szokásos programvégrehajtás

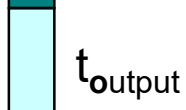
Olvasás



Feldolgozás



Írás



$$T = t_i + t_r + t_o$$

Kihasználtság

Olvasó $\frac{t_i}{T}$

CPU $\frac{t_r}{T}$

Nyomtató $\frac{t_o}{T}$

ha $t_i, t_o \gg t_r$
a CPU kihasználtsága
kicsi

Használjunk gyorsabb I/O eszközt a CPU táplálására

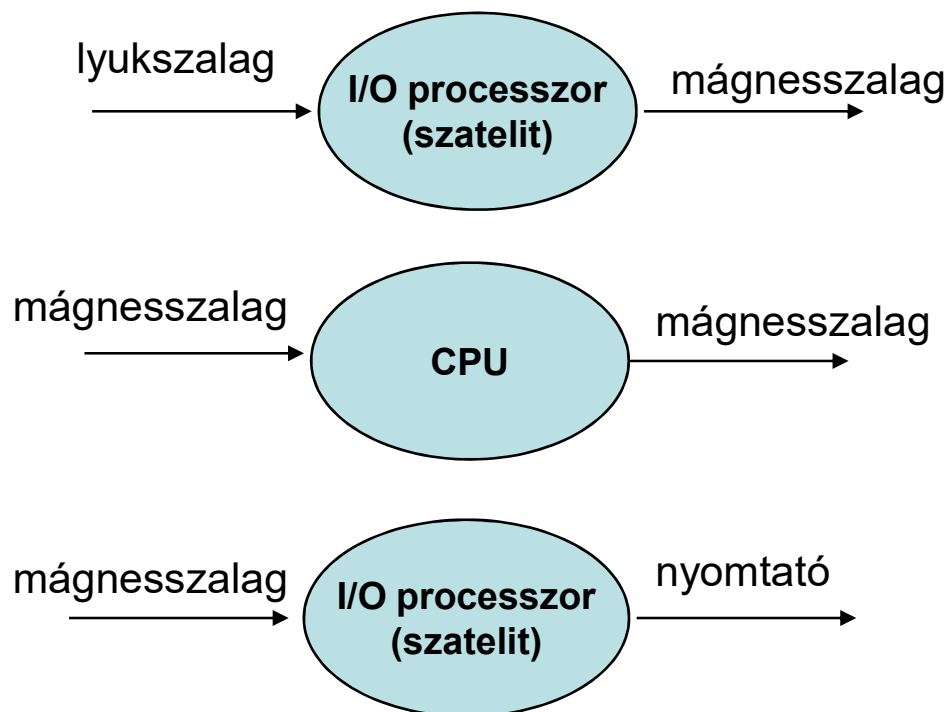
Lyukszalag olvasó: 1000 kar/perc

Sornyomtató: 1000 sor/perc

Mágnesszalag: 80000 kar/sec

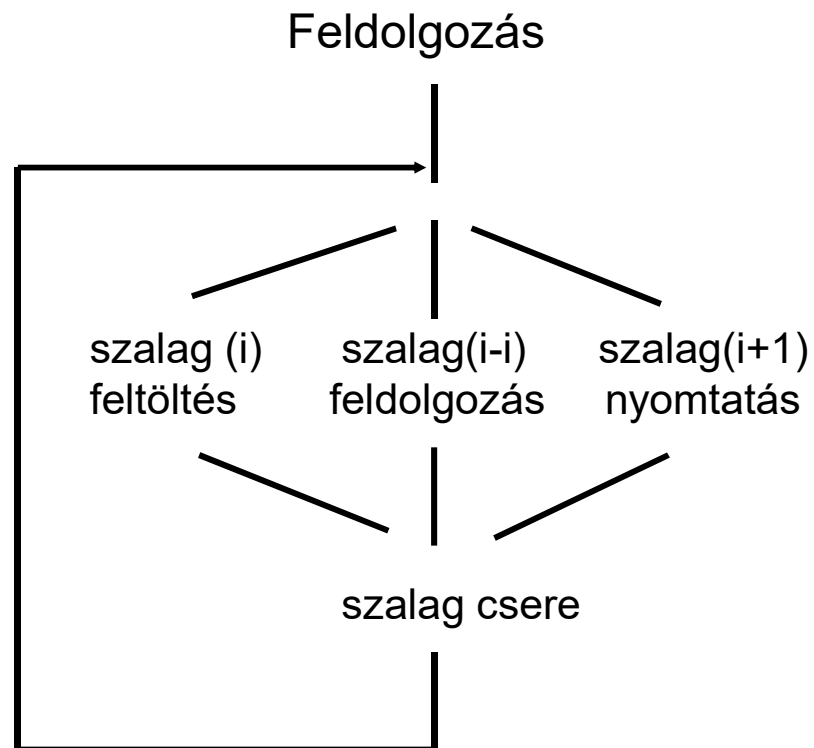
Off-line I/O

Szatelit processzorok (többszörözhetők)



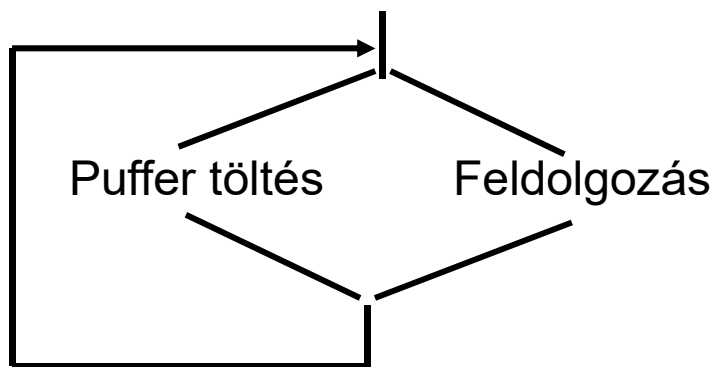
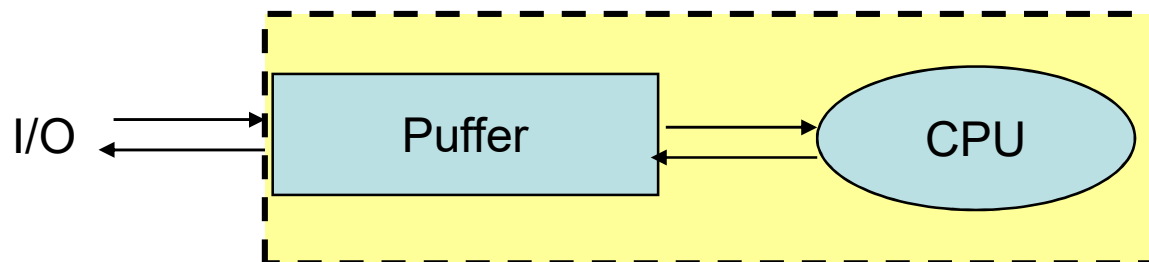
A felhasználó programnak tudnia kell arról,
hogyan szalagról olvas?

NEM → logikai perifériák



Processzor kihasználtság: 90%
Áteresztés: 55 job/óra

Pufferelés



Hardver támogatás igény:

Blokkos átvitel

Megszakítás

DMA

Akkor kiegyenlített, ha $\sum t_{io} = \sum t_r$

Egy JOB vagy I/O vagy CPU igényes

SPOOLING (Simultaneous Peripheral Operation On-line)

A puffer diszken van

A diszk nem soros hozzáférésű

→ a feladatok (JOB) sorrendje dinamikusan változtatható

I/O igényes és CPU igényes feladatokok együtt futatthatók

→ kiegyenlítés esélye nő

->> JOB scheduling (ütemezés)

BATCH

- Válasszunk ki egy JOB-ot.
- Fut, amíg tud.
- Ha már nem, függesszük fel.
- Válasszunk ki egy másikat, de jegyezzük meg, hogy az előző mire vár.
- Ha a felfüggesztett várakozási feltétele teljesül futtassuk tovább

MULTIPROGRAMOZÁS



Multiprogramozás – Megoldandó feladatok

Melyik JOB fusson

CPU ütemezés

Egyszerre több program lehet a memóriában

Memória gazdálkodás

A periféria használatot koordinálni kell

Periféria kezelés, ütemezés

Együttfutási problémák kezelése

Szinkronizálás

Ükzések, patthelyzet

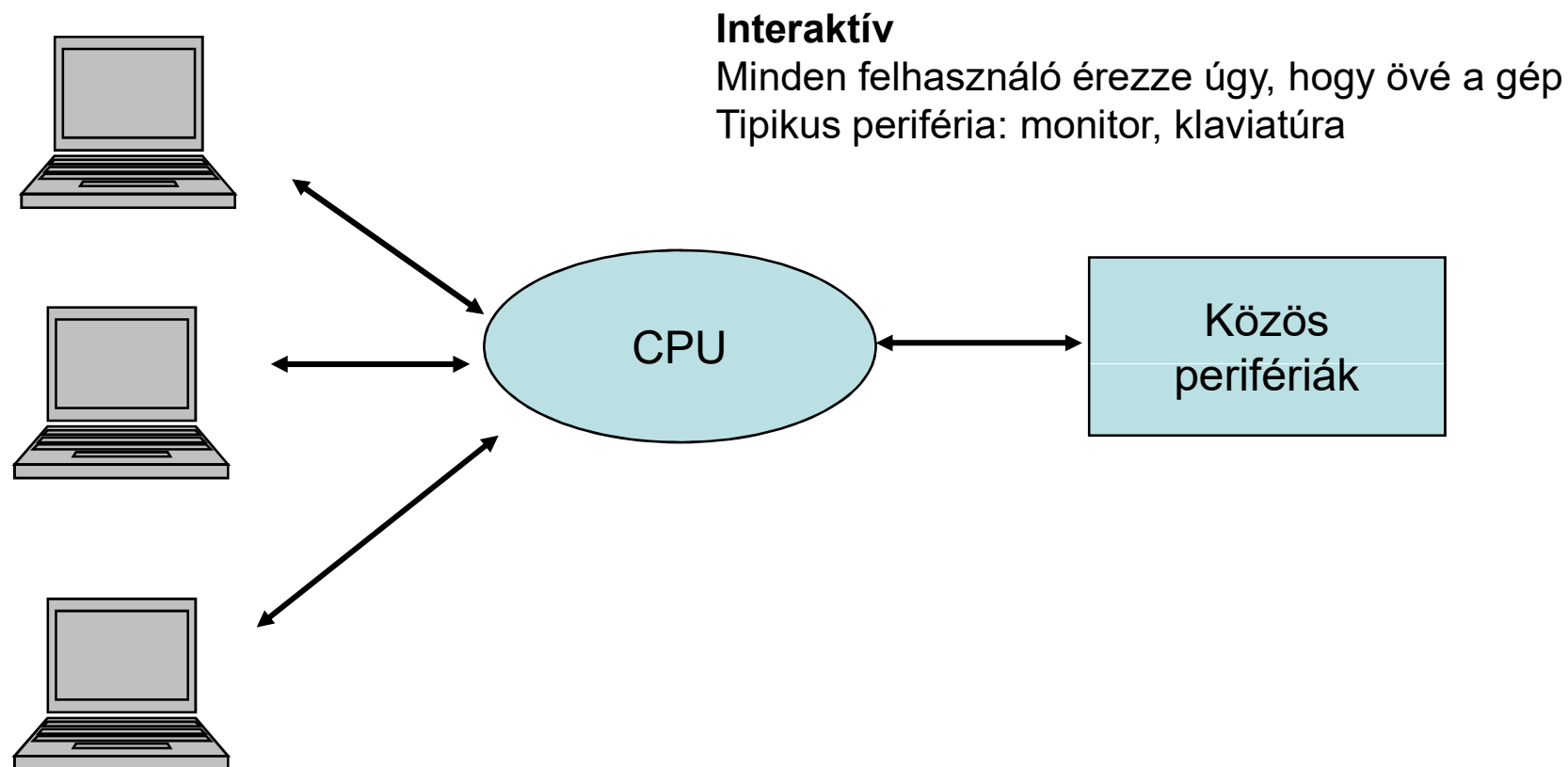
Holtpont kezelés

Védelmi kérdések

JOB $\leftrightarrow$ JOB

JOB $\rightarrow$ operációs rendszer

Időosztásos (time sharing) rendszerek



Valós idejű (real-time) rendszerek

A válaszidő kritikus

Kemény valós idejű (hard real-time): adott időn belül biztosan válaszol

Lágy valós idejű (soft real-time): adott időn belül nem 1, de nagy valószínűséggel válaszol

Tipikusan irányító rendszerek, célrendszerek

Beágyazott rendszerek

Feladatra specializált

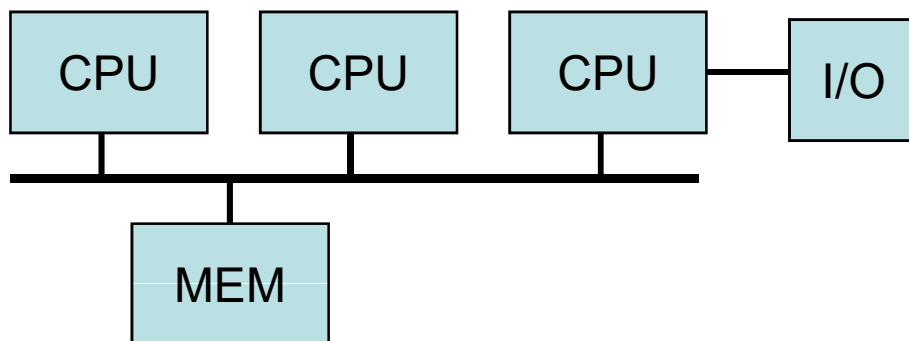
Környezetbe ágyazott

ipari berendezés, háztartási berendezés, orvosi eszköz, jármű, ...

Valós idejűség, biztonságkritikusság

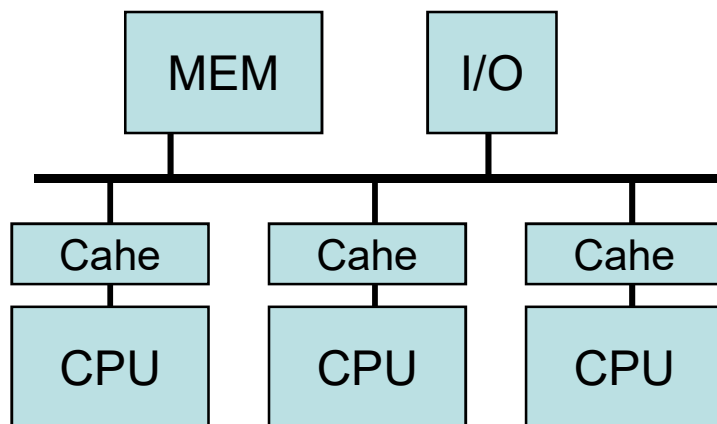
Multiprocesszoros rendszerek

Aszimmetrikus



A processzorok nem egyenértékűek feladathoz rendelvek

Szimmetrikus



A processzorok egyenértékűek

Operációs rendszer struktúrák

➤ Monolitikus

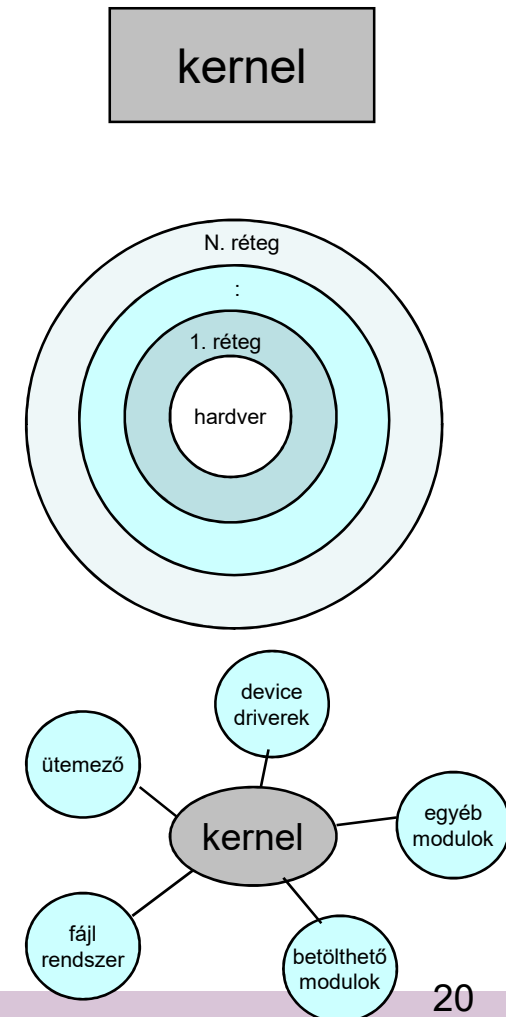
- Egyetlen program
- Jó teljesítmény
- Érzékeny a hibákra

➤ Rétegzett

- 0. réteg a hardver, N. réteg a felhasználói interfész
- Rugalmas, bővíthető
- Jól definiált interfészek a rétegek között
- Egy réteg csak a közvetlenül alatta és felette lévő réteggel tart kapcsolatot
- Magas rétegszám esetén lassú lehet

➤ Moduláris

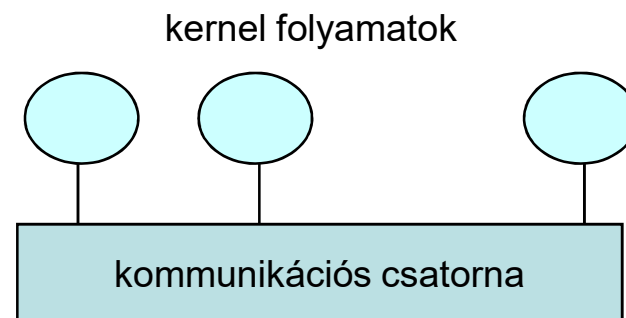
- Objektum-orientált megközelítés
- Minden objektum különálló
- A többiekkel ismert interfészen keresztül kommunikál
- Betölthető csak akkor, ha szükség van rá



Operációs rendszer struktúrák

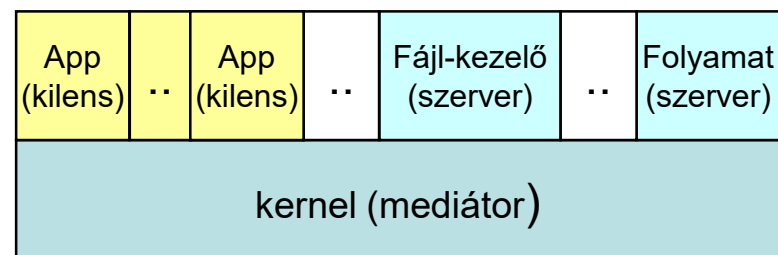
➤ Mikrokernel

- Ami csak lehet a „felhasználói” térben
- Kommunikáció üzenet továbbítással
- Védett módban csak a legszükségesebbek
- Előnyök
 - Könnyű bővíthetőség
 - Könnyű áthelyezhetőség más hardverre
 - Megbízható (keves kód fut kernel módban)
- Hátrány
 - A kommunikáció lassú



➤ Kliens-szerver

- Absztrakt modell
- Felhasználói programok: kliens
- Rendszer programok: szerver
- A kernel biztosítja az összeköttetést a szereplők között



Operációs rendszer struktúrák

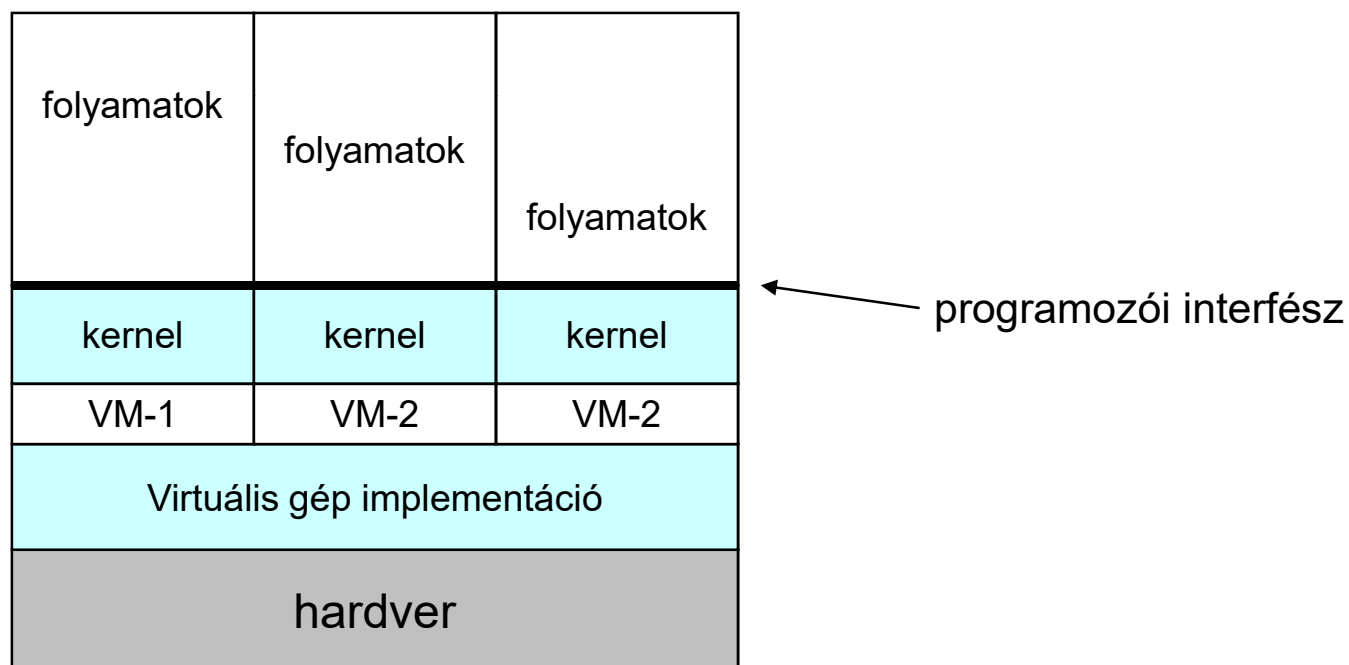
➤ Virtuális gép

A hardver és az operációs rendszer egységként kezelése

Mindenki saját hardvert lát

Jó virtualizációhoz hardver támogatás szükséges

Az időzítés problémás lehet – lassabb mint a valódi gép



Multiprogramozás

Program → Folyamat (Process)

Szekvenciális végrehajtás

Operációs rendszer funkció elérése: **rendszerhívással**
(privilegium szint váltás!)

Tipikus rendszerhívások

I/O művelet kezdeményezése

Erőforrás igénylés

Várakozás más program jelzésére/üzenetére

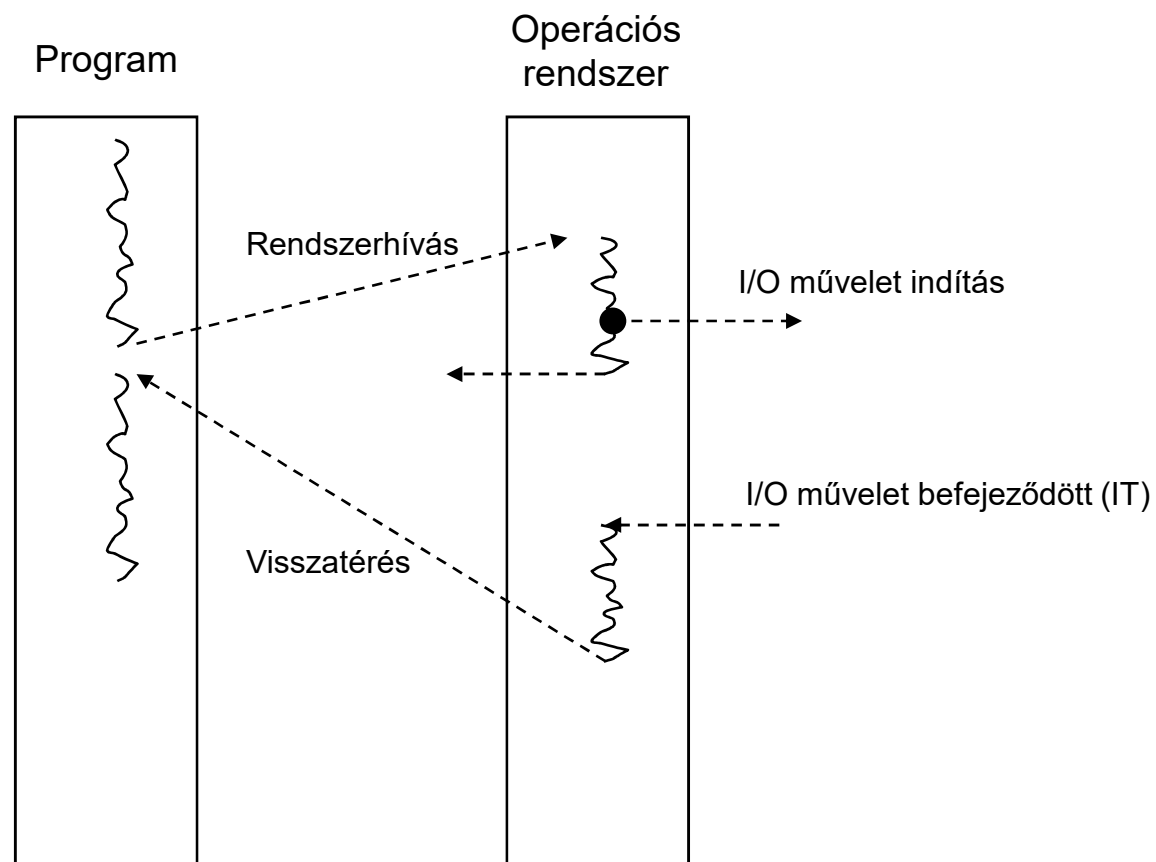
Adott idejű (vagy adott időpontig tartó) várakozás

Újraütemezés kérése (lemondás a processzorról)

Program befejezése (az erőforrások felszabadítása)

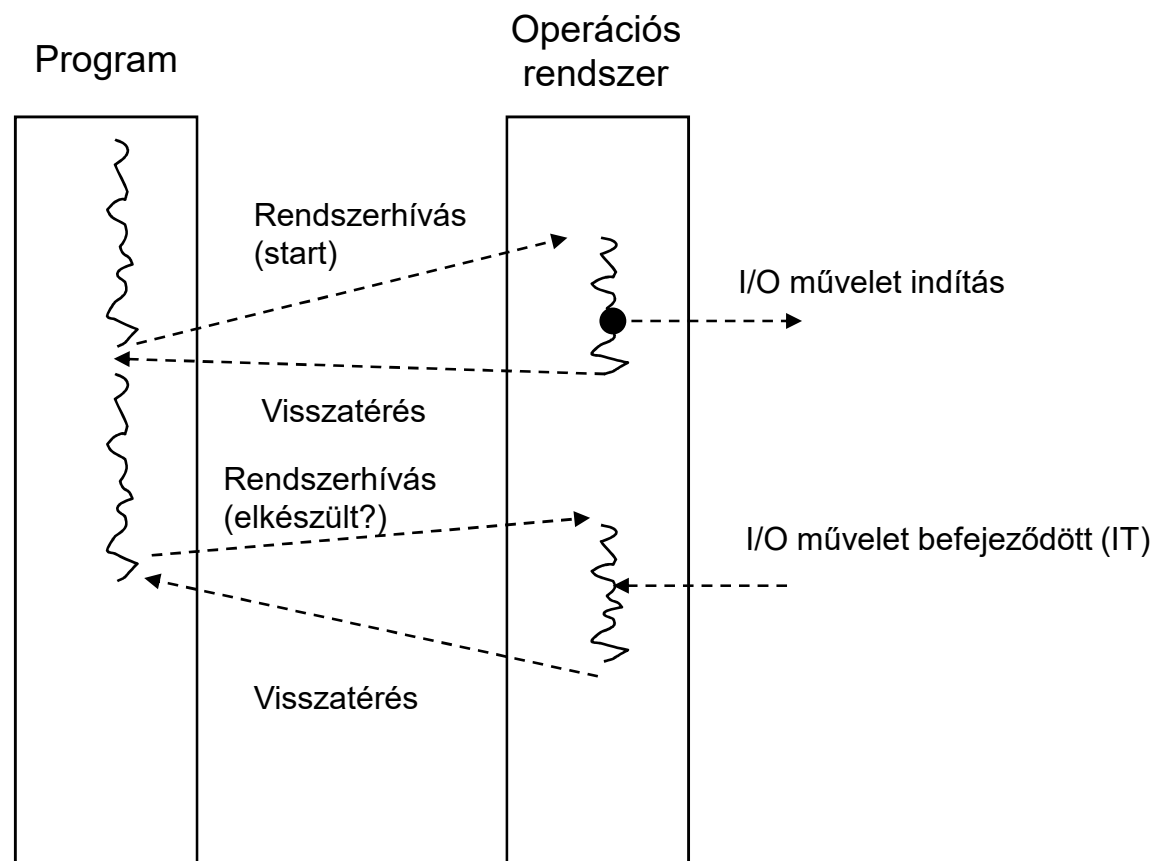
Rendszerhívások

Szinkron



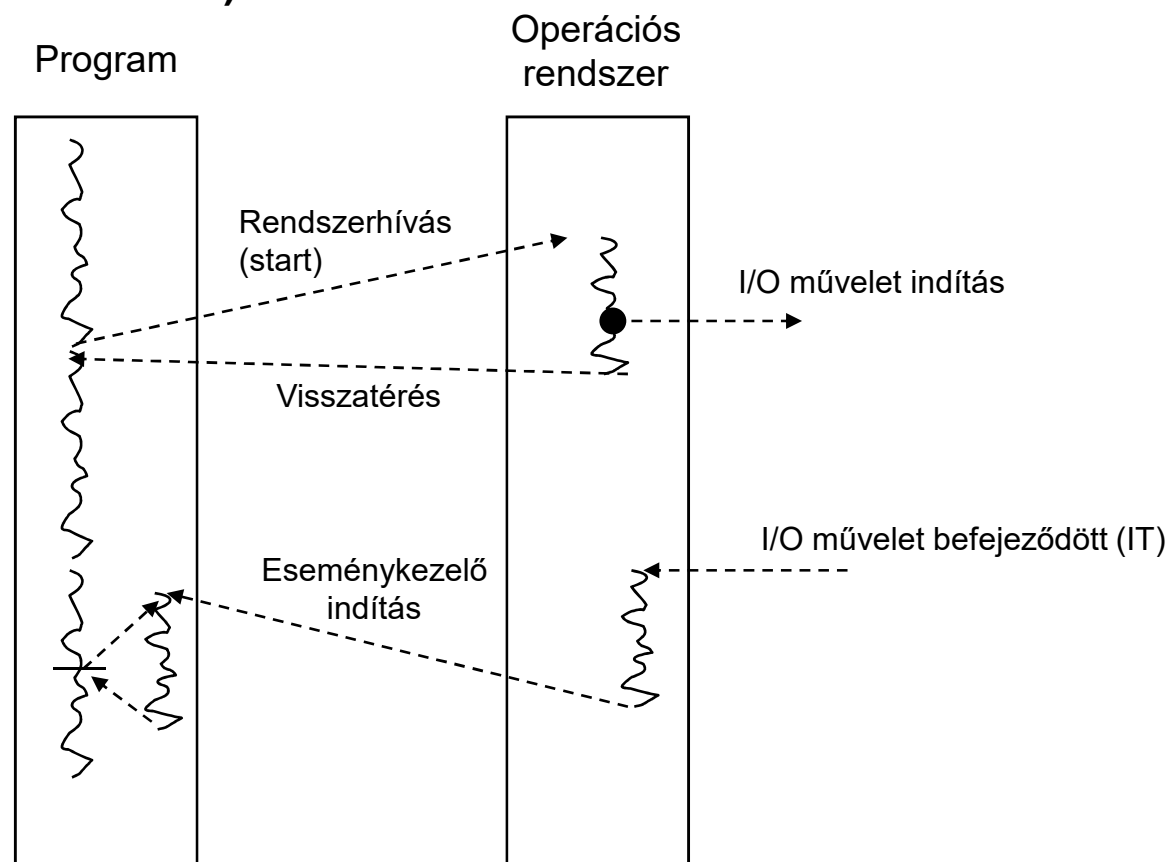
Rendszerhívások

Aszinkron



Rendszerhívások

Aszinkron (eseménykezeléssel)



Megszakítások

Aszinkron esemény a rendszerből vagy a külvilágból

A megszakítást hardver jelzi

Az operációs rendszer jelzésen (signal) keresztül értesítheti a felhasználói programot

Tipikus források

- I/O eszköz

 - I/O művelet vége (normálisan vagy hibával)

 - Karakteres eszköz (karakterenként)

 - Blokkos átvitelű eszköz (blokkátvitel végén – DMA)

- Időzítő

 - Periodikus jelzés (pl. 100 ms)

 - Időzítő lejárt

Trap (kivétel)

Hibás művelet amelyet a hardver észrevesz

- Nullával osztás

- Érvénytelen utasítás

- Érvénytelen cím

- Sín időzítés hiba (WATCH DOG)

Az aktuális műveletet nem lehet befejezni

Az operációs rendszer lehetséges válaszai

- Ismétlés (ha csak tranziens hiba)

- Folyamat megszakítás (abort)

- Hibaüzenet küldés az operátornak

- Diagnosztika indítása

- (Kék halál ☺)

Szándékos trap

- Emuláció (pl. lebegőpontos műveletek megvalósítása szoftverrel)

- Töréspont (program nyomkövetés)



Operációs rendszerek

Megszakítás (IT) és kivétel (Trap) kezelés különbsége

IT

Elfogadása két utasítás között lehetséges, az aktuális utasítás még befejeződik
Az IT rutin lefutása után (RETURN) a következő utasítással folytatódik a végrehajtás

Trap

Bekövetkezésekor azonnal elfogadásra kerül

A továbblépéshez

- Közbenső állapotokat menteni kell és tudni kell a félbeszakadt utasítást folytatni az állapot visszatöltése után
- Utasítás-rollback és a félbeszakadt utasítás újrafuttatása

Rendszer indulás/megállás

Betöltés (boot)

- Bootstrap loader
- Boot block
- Firmware

Leállítás

- Állapotok mentése

Hibernálás

- Memória kép mentés a gyors visszaállításhoz

Vészleállítás (tápfeszültség kimaradás)

- A legszükségesebb állapotinformációk mentése