

# OPERÁCIÓS RENDSZEREK 1. – BEVEZETÉS

SOÓS SÁNDOR

Nyugat-magyarországi Egyetem  
Simonyi Károly Műszaki, Faanyagtudományi és  
Művészeti Kar  
Informatikai és Gazdasági Intézet  
E-mail: soossandor@inf.nyme.hu

2015. ŐSZ

## Tartalomjegyzék.

## Tartalomjegyzék

<b>1. Alapfogalmak</b>	<b>3</b>
1.1. Az operációs rendszer fogalma . . . . .	3
1.2. Az operációs rendszer feladata . . . . .	7
<b>2. Az operációs rendszerek története</b>	<b>8</b>
2.1. 0. generáció . . . . .	9
2.2. 1. generáció . . . . .	9
2.3. 2. generáció . . . . .	10
2.4. 3. generáció . . . . .	12
2.5. 4. generáció . . . . .	15
2.6. 5. generáció . . . . .	18
<b>3. Befejezés</b>	<b>18</b>
3.1. Emlékeztető kérdések . . . . .	18

## Bevezetés

### A félév rendje

#### Bevezetés.

- A tárgy célja:
  1. Ebben a félévben csak előadás – Operációs rendszerek 1. (OPRE1)
    - Az operációs rendszer fogalma
    - Az operációs rendszerek kialakulása, fejlődéstörténete
    - A multiprogramozott operációs rendszerek jellemzői
    - Biztonsági kérdések
    - A rendszermag (kernel) legfontosabb feladatai
    - Hálózati elosztott rendszerek
  2. Jövő félévben csak gyakorlat – Operációs rendszerek 2. (OPRE2)
    - Windows és Linux operációs rendszerek használata, üzemeltetése

## **A félév rendje.**

- Előadás
  - Minden héten hétfőn 10:00-12:30, GT021
  - 3x45 perc, egy 15 perces szünetet tartunk
  - Az előadásokon való részvétel kötelező, katalógus
  - Legfeljebb 2 hiányzás lehetséges orvosi igazolással
  - 1 zárthelyi dolgozat
  - Szóbeli vizsga
- Az aláírás feltétele:
  - Részvétel az előadásokon, legfeljebb 2 kivétellel
  - A zárthelyi megírása legalább elégséges eredménnyel
- A félévi jegy:
  - Zárthelyi: 50%
  - Szóbeli vizsga: 50%
- Mit kell tudni a vizsgán és a zárthelyin?
  - Ami az előadáson elhangzott
  - Ami a kiadott előadásvázlatokban szerepel
- Javasolt módszer:
  - Előadásvázlat kinyomtatva + saját jegyzetelés
  - Az előadások végén emlékeztető kérdések!
  - Az előadáson notebook nem használható!
  - Mindig szabad kérdezni!

## **Mit adunk ezért?.**

- A két félév alatt 3-3 kredit!
- Elméleti tudás, gyakorlati tapasztalat
- Jobban értjük, mi történik a számítógépben

## Információ források

### Információ források.

- Könyvek:
  - Kóczy Annamária-Kondorosi Károly: **Operációs rendszerek mérnöki megközelítésben** (Panem, Budapest, 2000.) – LINK
  - A. S. Tanenbaum, A. S. Woodhull: **Operációs rendszerek, Tervezés és implementáció, 2. kiadás** (Panem - Pearson-Prentice-Hall, Budapest, 2007.)
  - Dr. Fazekas Gábor: **Operációs rendszerek** – LINK
- Előadásjegyzetek:
  - A tárgy honlapján: INGA portál
  - Google Classroom
  - Kétféle PDF fájl minden előadáshoz
- Internet:
  - „Google a barátunk”

## Hibakereső pályázat

### Hibakereső pályázat.

- Ha valaki hibát talál a weben lévő előadásban, azt elküldheti nekem e-mailben az előadás kezdetéig
- Minden hiba első beküldője a hiba súlyától és jellegétől függően pontokat kap, amit a félév végén beszámítunk a zárthelyi értékelésébe
- A hiba lehet egyszerű elírás, vagy tartalmi hiba egyaránt. Mindkét esetben meg kell indokolni, hogy miért hibás, és ha lehetséges meg kell adni a javítást
- Sikeres nyomozást kívánok!

## 1. Alapfogalmak

### 1.1. Az operációs rendszer fogalma

Az operációs rendszer fogalma.

- Mit nevezünk operációs rendszernek?
- Miért van szükség operációs rendszerre?
- Milyen a jó operációs rendszer?
- Milyen operációs rendszereket ismerünk?
- Mit csinál az operációs rendszer?
- Létezik számítógép operációs rendszer nélkül?

Ezekről a kérdésekről fog szólni ez a tantárgy!

### **Hogyan működik egy mai számítógép?.**

- Mit kapunk, ha veszünk egy számítógépet?
- Hogyan vesszük használatba?
  - Kibontjuk a dobozt
  - Mi történik, ha most bekapcsoljuk a gépet?
  - Működik most a számítógép?
  - Tudjuk használni valamire?
  - Mi kell ahhoz, hogy használni tudjuk?
- Hardver – Szoftver
- Milyen szoftverekre van szükségünk?

### **Miért van szükség az operációs rendszerre?.**

Két nézőpont: felhasználó – programozó

#### **1. A felhasználó szempontjából**

- a hardver kezelése nagyon bonyolult:
  - csak egyszerű parancsokat ért meg: pörgesd fel a lemezt, vidd az olvasófejet egy adott szektor fölé, olvass be x bájtot stb.
  - mit szeretne a felhasználó: játszd le a behelyezett lemezen lévő filmet
  - ezt teszi lehetővé az operációs rendszer
- sokféle hardvereszközt kell kezelni:

- fájlok tárolására szolgáló eszközök: merevlemez, hajlékonylemez, CD, DVD, pendrive, memóriakártya, hálózati meghajtók
- különböző fajta nyomtatók: mátrix, tintasugaras, lézer,
- mi lenne ha ezeket különbözőképpen kellene kezelni?

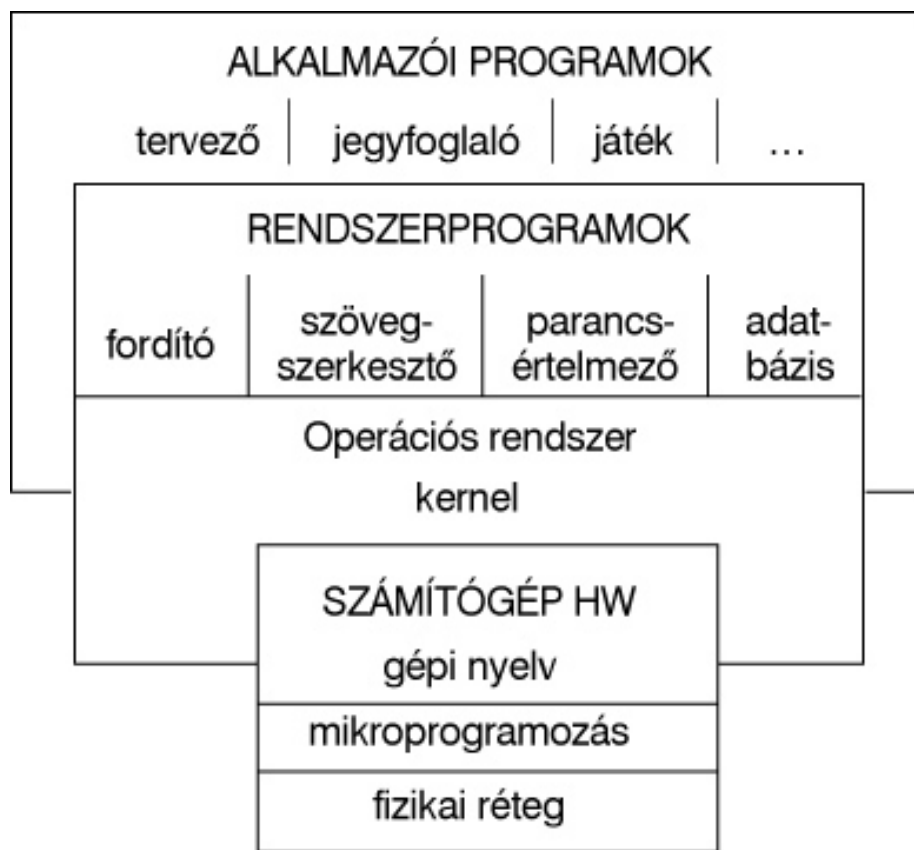
## 2. A programozó szempontjából

- ugyanezek a szempontok a programozó szempontjából is felmerülnek
- egyszer mindenképpen meg kell írni ezeket a hardverkezelő programrészleteket, de nem lenne jó, ha
  - ... minden programozónak értenie kellene az alacsonyszintű programozáshoz
  - ... minden programozónak ismernie kellene a különböző gyártók eszközeinek programozását
  - ... egy új eszköz megjelenésekor újra kellene írni minden programot
  - ... minden programozó hozzáférne a hardvereszközök alacsonyszintű kezeléséhez, nagyon veszélyes lenne!
  - ... ha az egyes programozóknak kellene összehangolniuk az eszközök működését és osztozkodniuk a hardver erőforrásokon

## A megoldás.

- Minden részfeladatot ott oldjunk meg, ahol az a legmegfelelőbb:
  - a hardvereszközök kezelését írja meg az eszköz gyártója, merevlemezgyártó, CD olvasó gyártója, nyomtatógyártó, stb.
  - mindenki tartson be bizonyos megállapodásokat, szabványokat
  - mindenki felhasználhatja a többi gyártó szabványos megoldásait
  - az eszközök együttműködését biztosító programokat írja meg egyetlen szoftvergyártó (operációs rendszer)
  - a többi szoftvergyártó használja az operációs rendszer szolgáltatásait, ne foglalkozzon az alatta lévő megoldásokkal
- Ezt nevezzük réteges felépítésnek
- Ezt a megoldást még nagyon sokszor fogjuk látni különböző informatika rendszerekben, a legkülönbözőbb tantárgyakban
- Minden rendszer ilyen?

## A számítógép réteges felépítése.



1. ábra. A számítógép réteges felépítése

### Az operációs rendszer fogalma.

- A réteges felépítés középső rétegének (RENDSZERPROGRAMOK) egy részét nevezzük operációs rendszernek
- A határok nem pontosan definiáltak
- Az egyértelmű, hogy a rendszermag (kernel) mindig az operációs rendszer része
- A kernel irányítja a számítógép működését
- Innen ered az elnevezés:
  - **Operating System**, magyarul **Működtető rendszer**
  - Az Operációs rendszer egy tükörfordítás, nem igazán szerencsés
  - A későbbiekben röviden **OS**-nek is fogjuk nevezni az angol elnevezés rövidítéseként
- Van, aki az operációs rendszer részének tekinti a gép használatát megkönnyítő segédprogramokat is
- Ebben a tárgyban elsősorban a kernellel fogunk foglalkozni

## 1.2. Az operációs rendszer feladata

### Az operációs rendszer feladata.

Az operációs rendszer alapvetően két feladatot lát el:

1. A **felhasználó** számára kényelmes és hatékony **munkavégzést** biztosít a számítógépen
2. Biztosítja a számítógép **erőforrásainak** hatékony, **biztonságos** és **igazságos** használatát (irányítás, koordinálás)

A **dőlt**, **félkövér** betűvel írt fogalmak további elemzést igényelnek. Erről (is) fog szólni ez a félév.

## Virtualizáció

### A virtuális gép koncepció.

- Napjainkban az informatika egyik legdivatosabb fogalma a virtualizáció
- Nagyon sok területen használjuk, sokféle értelemben
- Gyakran fogunk találkozni ezzel a különböző órákon és a gyakorlatban is



- **Most:** Az operációs rendszer is létrehoz egy virtuális gépet
  - A valódi számítógépen (hardver) fut
  - A felhasználó számára egy másik (virtuális) számítógépet mutat
    - \* merevlemez, pendrive, CD, DVD eszközök helyett C:, D:, E: meghajtók
    - \* VGA, TFT monitor, projektor helyett különböző felbontású kijelzők
    - \* különböző nyomtatók helyett nyomtatásvezérlő
    - \* stb.

## 2. Az operációs rendszerek története

### Az operációs rendszerek története.

- Az operációs rendszerek története szorosan összekapcsolódik az általuk működtetett számítógépek történetével
- A fejlődés vizsgálata segít megérteni az operációs rendszerek lényegét
- Speciális szempontból vizsgáljuk a fejlődést (részletesebben: Számítógép architektúrák)
- Az operációs rendszerek fejlesztői minden korban arra törekedtek, hogy az adott kor színvonalán minél jobban ki tudjuk használni a (drága) hardver eszközöket
- Amikor egy adott szinten sikerült hatékony megoldást találni, akkor továbbléptek a következő célok megoldása felé
- A fő szempont a hatékonyság növelése
- Mit értünk hatékonyság alatt?

### Hatékonyság.

$$hatekonysag = \frac{eredmeny}{raforditas}$$

- Hogyan növelhető a hatékonyság?
  1. Az eredmény javításával: pontosabb, gyorsabb
  2. A ráfordítás csökkentésével: kevesebb erőforrás használata (emberi munka, gépidő)

- Az operációs rendszerek fejlesztésekor mindkettőn dolgozunk folyamatosan
- A hatékonyság mérése:
  - Processzor kihasználtság
  - Áteresztőképesség:  $\frac{munka(job)}{ido(ora)}$

## 2.1. 0. generáció

### Őskor, 0. generáció.

- Az első digitális számítógépeket Charles Babbage (1791-1871) tervezte differenciál gép és analitikai gép néven
- Nem sikerült megépítenie működőképes állapotban, de meglepően hasonló elveket használt mint a mai számítógépek
- Az eredeti tervek alapján, korabeli alkatrészekből sikerült megépíteni 1989-91-ben
- Ő alkalmazta az első programozót Ada Lovelace személyében (Lord Byron lánya), róla nevezték el az Ada programozási nyelvet
- Bővebben: <http://hu.wikipedia.org/wiki/Babbage>
- Ezeknek a gépeknek még nem volt operációs rendszere

## 2.2. 1. generáció

### 1. generáció (1945-1955): vákuumcsövek és kapcsolótáblák.

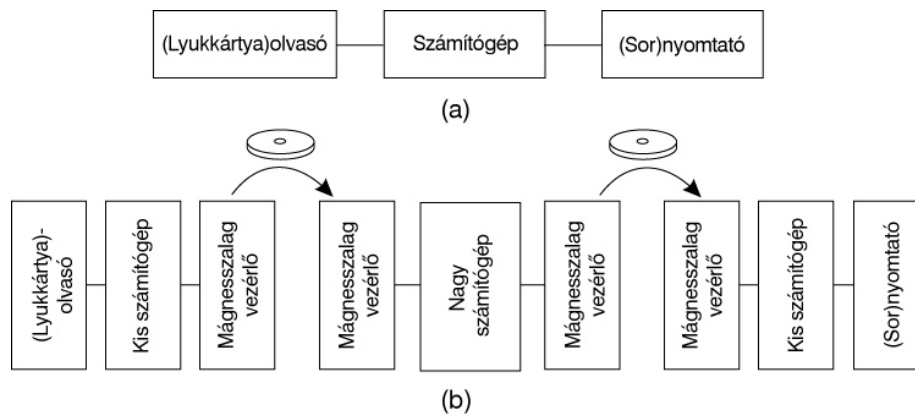
- II. világháború, rejtjelezés, kódfejtés
- teljes mérnöksapat kezelte a számítógépet, mindent ez a csapat csinált
- gyakori meghibásodás
- gépi kód, kapcsolótáblák
- programozási nyelvek nem voltak, még assembly sem
- a programozás is kapcsolótáblákkal történt
- az 1950-es évek elejétől lyukkártyák
- nem volt operációs rendszer

## 2.3. 2. generáció

### 2. generáció (1955-1965): tranzisztorok és kötegelt rendszerek.

- a tranzisztorok megjelenésével megnőtt a számítógépek megbízhatósága
- a számítógép hasznos munkát tudott végezni
- kezelőszemélyzet: tervezők, gyártók, kezelők, programozók, karbantartók
- nagyszámítógépek, mainframe-ek, légkondicionált gépterem
- A munkafolyamat:
  - programozó megírja a programot papíron, FORTRAN, vagy assembly nyelven
  - lyukkártyára lyukasztják
  - a kártyacsomagot beviszik a fogadó terembe, átadják a kezelőknek
  - a kezelő lefuttatja a programot, kinyomtatja az eredményt
  - a kiviteli teremben a programozó megkapja a kinyomtatott eredményt (hibalistát)
- Idővesztés: ha egy feladathoz szükség van valamilyen eszközre, pl. FORTRAN fordító, akkor a kezelő betölti azt is a gépbe
- Cél: minél jobban használjuk ki a rendelkezésre álló gépidőt
- Megoldás: **kötegelt feldolgozás, batch feldolgozás**
  - a fogadóteremben összeválogatják a hasonló feladatokat (job), pl. a FORTRAN fordítót igénylőket
  - ezeket egymás után futtatják
  - csak egyszer kell betölteni a FORTRAN fordítót
- További hatékonyságnövelő megoldások:
  - mágnesszalag használata
  - beállítunk egy vagy két olcsó számítógépet a periféria műveletekre
  - online helyett offline működés
  - egyszerű monitor program (resident monitor) (az első operációs rendszer)
  - jobvezérlő program
  - memória felosztása: monitor memóriaterület – felhasználói terület
- Új munkafolyamat:

- a fogadó oldalra beállítunk egy kisebb számítógépet, ami beolvassa a lyukkártyákat és mágnesszalagra írja azokat
  - a mágnesszalagot az operátorok átviszik a nagy számítógéphez
  - a monitor program beolvassa a mágnesszalagot és feldolgozza a rajta lévő feladatokat a jobvezérlő programok alapján
  - a programok eredményeit felírja egy másik mágnesszalagra
  - az operátorok kiviszik a mágnesszalagot a kimeneti teremben lévő harmadik számítógéphez
  - a kimeneti számítógép kinyomtatja a programok eredményeit
- Eredmény:
    - a drága központi számítógép ideje nagy részében a feladatokkal foglalkozhat
    - a lassú periféria műveleteket olcsóbb számítógépekkel végeztetjük
    - az operátorok munkájának egy részét (programok betöltése, elindítása) az állandóan a memóriában lévő monitor program végzi, ez is gyorsítja a központi számítógép munkáját, mert nem kell várni az operátorra
  - A mai számítógépekben használjuk ezeket a módszereket?
  - Hol használjuk ezeket a módszereket a mai számítógépekben?
    - Nyomtatás:
      - \* Fájl menü, Nyomtatás parancs
      - \* Dolgozhatunk tovább
      - \* A program a háttérben elküldi a nyomtatnivalót a nyomtatásvezérlőnek, és az elvégzi a nyomtatást
      - \* A nyomtatóban is van egy processzor, ami elvégzi a nyomtatást
    - Nagyfelbontású 2D/3D grafikus megjelenítés:
      - \* a mai korszerű grafikus kártyákon van processzor és memória
      - \* a CPU elküldi a kirajzolandó kép adatait a grafikus kártyának és dolgozik tovább
      - \* a grafikus kártyán lévő processzor rajzolja ki a képet
    - Batch fájlok DOS és Windows alatt (\*.BAT), script fájlok Linux alatt
      - \* Egy szöveg fájlban leírjuk azokat a parancsokat, amelyeket egymás után végre akarunk hajtani
      - \* A batch/script fájl elindításakor az operációs rendszer sorban végrehajtja a benne szereplő parancsokat
      - \* Használhatunk vezérlőszerveket (elágazás, ciklus), lásd Programozási alapok 1.



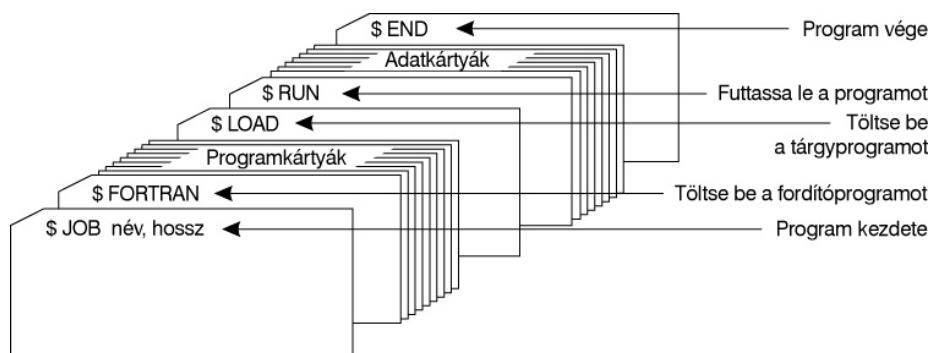
2. ábra. On-line és off-line perifériás műveletek

### On-line és off-line perifériás műveletek.

(a) On-line perifériák

(b) Off-line perifériák

### Munka vezérlő program – OS Job Control.



3. ábra. Munka vezérlő program – OS Job Control

## 2.4. 3. generáció

### 3. generáció (1965-1980): integrált áramkörök.

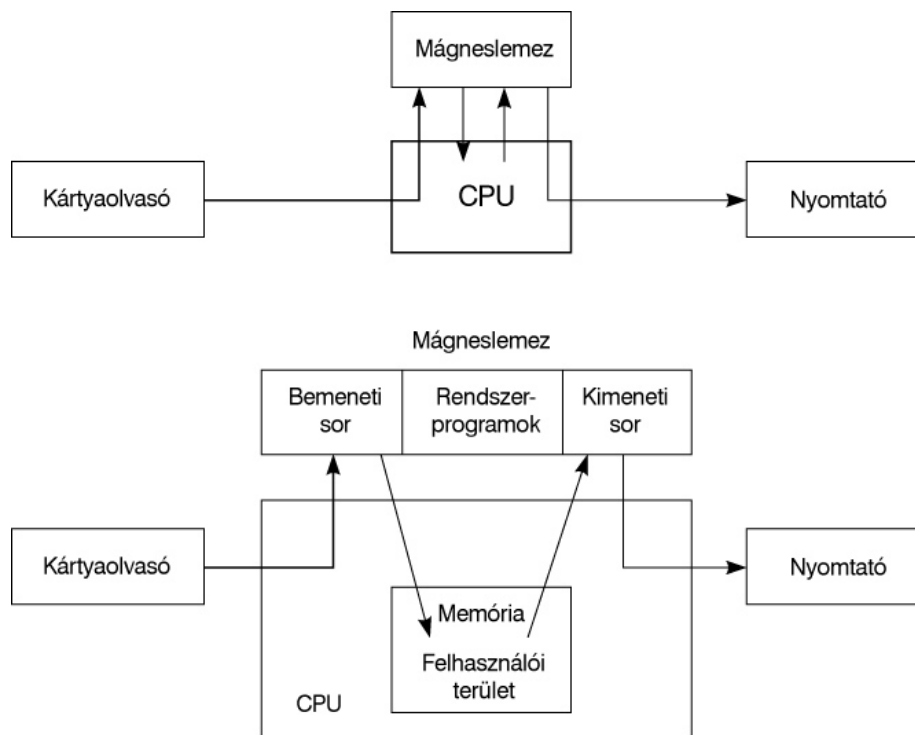
- A 2. generációban alkalmazott módszerek megoldottak egy fontos problémát: megnövelték a drága számítógépek kihasználtságát, ezzel lecsökkentették a fajlagos költségeket

- Viszont születtek egy új problémát: a számítóközpontokban kétféle számítógép típus jelent meg
  1. nagy teljesítményű tudományos komputer
  2. kisebb teljesítményű perifériakezelő számítógépek
- Ezeket eltérő módon kellett programozni, több, eltérő tudású programozóra, fejlesztőeszközre volt szükség
- Ez megnövelte a költségeket
- A számítógépgyártók erre a problémára kerestek megoldást
- Az IBM válasza: System/360 gépcsalád
  - a család különböző méretű (memória, gyorsaság, perifériák, ár, teljesítmény) számítógépekből állt
  - de ezek egymással kompatibilisek voltak, azaz bármelyik gépre írt program fut az összes többin is (legalábbis elvileg)
  - ez sok előnnyel járt, de sokkal bonyolultabb operációs rendszerre volt szükség
  - Új perifériák: mágnesdob, mágneslemez
    - \* nagykapacitású
    - \* gyors
    - \* véletlen hozzáférésű (random access): a mágnesszalaggal ellentétben nem csak sorban egymás után lehet elérni a tárolt adatokat, hanem tetszőlegesen (így működnek a mai merevlemezek és a legtöbb háttértár, kivéve a szalagos egység – streamer)
- Új technikák
  - A véletlen elérésű tárolók két új technikai megoldás kidolgozását tették lehetővé:
    - \* spooling technika
    - \* multiprogramozás
  - Ezeket ma is folyamatosan használjuk
- Spooling technika
  - simultaneous peripheral operation on-line – on-line szimultán periféria (háttértárolás)
  - a számítógép háttérben el tudja végezni periféria műveleteket
  - nincsen szükség külön számítógépre a perifériák kezeléséhez
- Multiprogramozás

- A spooling technika segítségével a számítógép képes a háttérben végrehajtani a periféria műveleteket, eközben a feladatok folyhatnak tovább
  - Ha azonban egyetlen feladat fut a gépen, akkor a periféria műveletek alatt a számítógép várakozni kényszerül, ezzel lecsökken a számítógép kihasználtsága
  - A mágneslemezek lehetővé teszik, hogy a processzor egyszerre több feladat adataihoz férjen hozzá és kiírja azok eredményét
  - Amikor egy feladat várakozni kényszerül, akkor a processzor elindít egy másik feladatot
  - Ehhez arra is szükség van, hogy a memóriát továbbtagoljuk, a monitor és az egyes feladatok által használt területekre
  - Ezek a memóriaterületek egymástól függetlenek, az egyes programok nem férhetnek hozzá a másik program által használt memóriaterülethez
- Új operációs rendszer (OS/360)
    - Ezek az új technikák új, a korábbiaknál sokkal nagyobb méretű, összetett operációs rendszert igényeltek
    - Az IBM ezt OS/360-nak nevezte el
    - A nagy méretű kód nagyon sok hibát eredményezett, a sok javítás pedig a régi hibák kijavítása mellett újabb hibákat okozott, a hibák száma körülbelül állandó volt
    - Ennek ellenére ezek a rendszerek elég jól kiszolgálták a felhasználók igényeit
    - A programozók azonban elkezdtek hiányolni a géppel való közvetlen kapcsolatot, ami utoljára az 1. generációs gépeknél volt meg
    - Erre az igényre adott választ az időosztás (timesharing) technika
  - Időosztás (timesharing)
    - Minden felhasználó kap egy on-line terminált (képernyő + billentyűzet), amivel közvetlenül kapcsolódik a számítógéphez
    - Parancsokat tud begépelni a billentyűzeten, a választ a képernyőn látja
    - Amikor a processzor felszabadul az éppen ellátott feladat alól, akkor kiválasztja a soron következőt és végrehajtja, az eredményt visszaküldi a felhasználó képernyőjére
    - Amikor egy felhasználó gondolkodik, vagy éppen gépeli a parancsot, akkor nem terheli a számítógépet, így egyetlen gép sok felhasználót tud interaktívan kiszolgálni, miközben egy nagy kötegelt feladaton is dolgozik

- Egy felhasználó viszont képes hosszú időre lefoglalni a teljes gépet egy sokáig futó parancs kiadásával
- Ezek a feladatok tovább bonyolították az operációs rendszert
- A legismertebb ilyen operációs rendszer a MULTICS (MULTIplexed Information and Computing Service)
- Az utolsó nagy MULTICS rendszert 2000. októberében állították le a Kanadai Védelmi Minisztériumban

### Spooling technika.



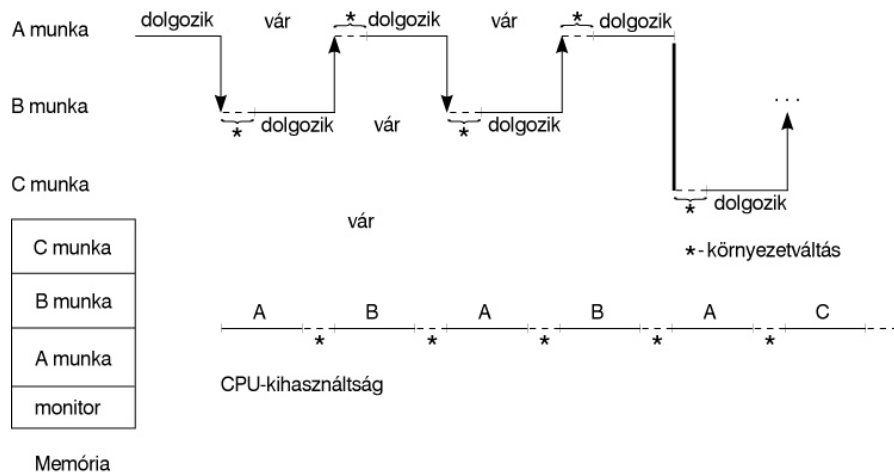
4. ábra. Spooling technika

### A multiprogramozás alapelve.

#### 2.5. 4. generáció

4. generáció (1980-napjainkig): személyi számítógépek.





5. ábra. A multiprogramozás alapelve

- A hardvereszközök árának rohamos csökkenésével alapvetően új irányt vet a számítástechnika fejlődése:
  1. Eddig a felhasználó osztozott a számítógépen sok más felhasználóval
  2. Ugyanakkor sok szakember állt rendelkezésre a számítógép üzemeltetéséhez
- Amikor megjelent a személyi számítógép (Personal Computer, PC), mindkét irányból jelentős változás következett be:
  1. A felhasználó egyedül birtokolja a számítógépet, nem osztozik rajta senkivel
  2. A felhasználónak egyedül kell tudnia kezelni a számítógépet, ő az operátor, a kezelő, a programozó és a felhasználó egy személyben
- Ehhez gyökeresen új megoldásokra, és új operációs rendszerekre volt szükség
- Megjelent egy új, eddig ismeretlen szempont az operációs rendszerek fejlesztésében, a felhasználóbarátság
  - nem elég jónak lennie az operációs rendszernek
  - az egyszerű felhasználónak meg kell értenie, tudnia kell kezelni, sőt meg kell szeretnie az operációs rendszert, különben másikat választ, vagy távoltartja magát a számítógéptől
  - kialakult a piaci verseny fogalma
  - A személyi számítógépeken működő operációs rendszerek: CP/M, DOS (DR-DOS, MS-DOS, Apple DOS), Mac OS, Windows, Linux

- Kezdetben az emberek „megelégedtek” a saját személyi számítógéppel
- Hamarosan megjelent az igény, hogy a személyi számítógéppel kapcsolódni tudjunk más számítógépekhez, miért?
  - elektronikus kommunikáció
  - erőforrások megosztása: saját erőforrások megosztása másokkal, távoli erőforrások elérése
- Ezen igények kielégítésére megjelentek a számítógépes hálózatok
- A hálózatok megjelenése ismét előtérbe hozott eddig háttérbe szorított szempontokat: biztonság, felhasználók azonosítása, védelme egymástól, jogszolgáltatások kezelése stb.
- Ezeket az igényeket kétféle operációs rendszer típussal tudjuk kielégíteni:
  - hálózati operációs rendszerek:
    - \* több felhasználó használhatja ugyanazt a számítógépet
    - \* egy felhasználó használhat több különböző számítógépet távolról belépve a számítógépre
    - \* ezek az operációs rendszerek nem térnek el gyökeresen a korábbi egyfelhasználós rendszerektől, csak új funkciókat építettek beléjük az új feladatok kezelésére
  - elosztott operációs rendszerek:
    - \* Olyan operációs rendszer, ami a felhasználó felé egy hagyományos egyprocesszoros rendszert mutat, miközben több processzor, vagy több teljes számítógép van a rendszerben
    - \* A felhasználó nem tudja, hogy ténylegesen melyik gépen, vagy melyik processzoron fut a programja, vagy hol tárolódnak az adatai
    - \* Mindezt a rendszer dönti el valamilyen szempontból optimálisan
  - Úgy is megfogalmazhatjuk a különbséget, hogy a hálózati operációs rendszer felhasználója tudja, hogy éppen melyik számítógépen dolgozik, az elosztott rendszer használója nem tudja
  - Ezt úgy mondjuk, hogy az elosztott rendszer transzparens (átlátszó)

### **Valós idejű rendszerek.**

- A 4. generációs rendszerek speciális fajtája
- Olyan rendszerek, amelyekkel szemben a környezeti valós időskálához kötött követelményeket támasztunk

- (A hagyományos rendszerek saját időskálával dolgoznak, függetlenek a valós időtől)
- Például adott időpontban kell végrehajtani valamit, vagy megadott időn belül kell válaszolni egy eseményre
- Ilyenek az ipari folyamatfelügyelő, -irányító, orvosi rendszerek, járművek fedélzeti rendszerei
- Ehhez speciális hardver és szoftvermegoldásokra van szükség

## 2.6. 5. generáció

### 5. generáció (napjainkban): Internet.

- Nyílt rendszerek (Open systems)
- Egységesített, szabványosított rendszerek
- Szabványos protokollok, csatlakozási felületek
- ISO (International Standards Organization) – OSI (Open System Interconnection)
- Az operációs rendszerek is szabványosak
- Minden számítógép képes kommunikálni minden másikkal
- Ez nem volt mindig így
- Egy gépen működhet több különböző operációs rendszer egyidőben
- Virtualizáció

## 3. Befejezés

### 3.1. Emlékeztető kérdések

#### Emlékeztető kérdések.

1. Hány hiányzás engedélyezett a félév során?
2. Mi az aláírás feltétele?
3. Hol található meg az előadás az Interneten?
4. Miért van szükség az operációs rendszerre?
5. Mit jelent a réteges felépítés (architektúra)?

6. Mit jelent az OS rövidítés?
7. Mi az operációs rendszer feladata?
8. Mit jelent a virtuális gép?
9. Mit jelent a hatékonyság? Hogyan mérjük a számítógépek esetében?
10. Mi jellemzi az egyes számítógép generációkat?
11. Ki volt a világ első programozója? Miről híres ezen kívül?
12. Mit jelent a köteget feldolgozás (batch)?
13. Hogyan növelhetjük a processzor kihasználtságát?
14. Mi a különbség az on-line és az off-line perifériaműveletek között?
15. Hogy néz ki egy job vezérlő program?
16. Mit jelent az, hogy két számítógép kompatibilis egymással?
17. Mit jelent az, hogy egy tároló véletlen hozzáférésű? Mi az ellentéte ennek?
18. Mit jelent a spooling technika?
19. Mit jelent a multiprogramozás?
20. Mit jelent a timesharing?
21. Mit jelent a PC rövidítés?
22. Mit jelent a felhasználóbarátság?
23. Mire használjuk a számítógép hálózatokat?
24. Mi jellemzi a hálózati operációs rendszereket?
25. Mi jellemzi az elosztott operációs rendszereket?
26. Mit jelent a transzparencia?
27. Mit nevezünk valós idejű rendszernek?
28. Mit jelent a nyílt rendszer?

**Befejezés.**

Köszönöm a figyelmet!