

# Operációs rendszerek I.

## Folyamatok



Várkonyiné Kóczy Annamária

Professzor

Informatika Tanszék

[\(koczya@ujs.sk\)](mailto:koczya@ujs.sk)

[varkonyi-koczy@uni-obuda.hu](mailto:varkonyi-koczy@uni-obuda.hu)

Felhasznált irodalom:

- Kóczy-Kondorosi (szerk.): Operációs rendszerek mérnöki megközelítésben
- Tanenbaum: Modern Operating Systems 2nd. Ed.
- Silberschatz, Galvin, Gagne: Operating System Concepts



## 2. Folyamatok

- Bevezetés
- Folyamatkezelés multiprogramozott rendszerben
- Környezet váltás
- Folyamatleírók, I/O leírók
- Szálak
- Megszakítások

# Bevezetés

- A folyamat (process) a multiprogramozott OS rendszerek alapfogalma:  
végrehajtás alatt álló program.

# Folyamatok

Multitaskos rendszerben több folyamat van a rendszerben egyidejűleg.

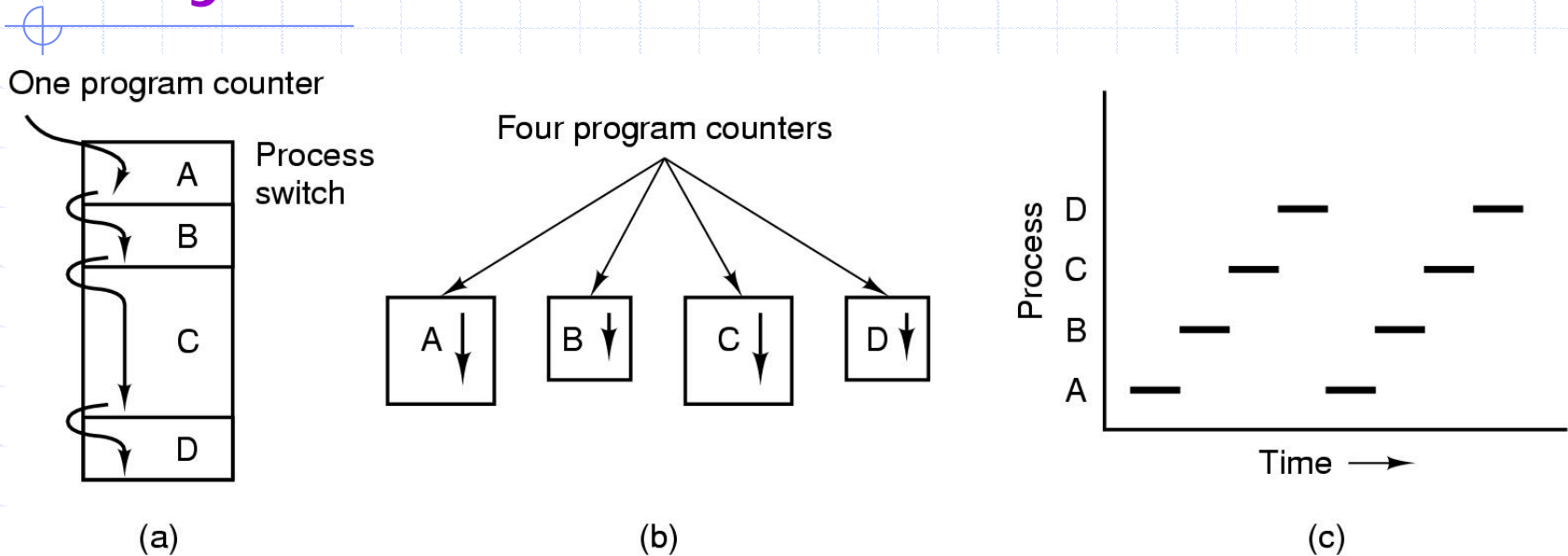
Elnevezések:

- munka (job) - batch rendszereknél
- feladat (task) - real time
- felhasználó - időosztásos rendszereknél
- Általános a *folyamat* név.

## 2.1. Folyamatkezelés multiprogramozott rendszerekben

- Folyamatok modellezése multiprogramozott környezetben
- Folyamatok állapotgráfja
- Az állapotok és állapotátmenetek
- Kibővített állapotgráf

# Folyamatok modellezése

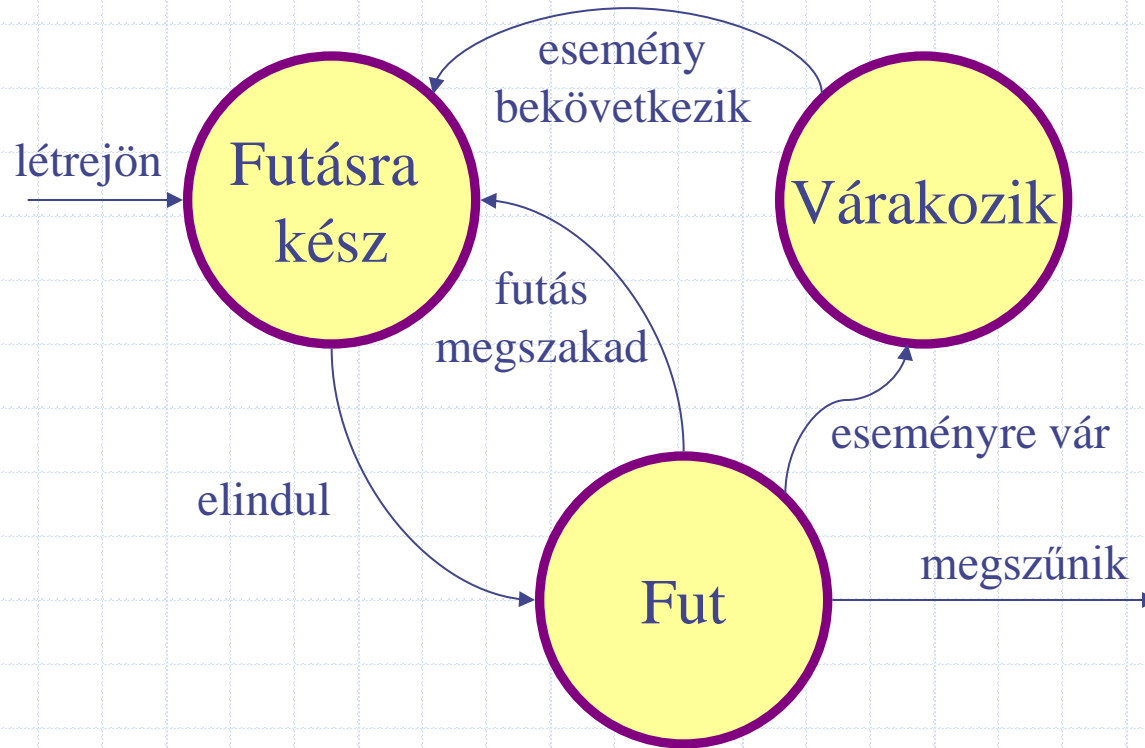


- a) Multiprogramozott rendszer (1 PC)
- b) Valódi párhuzamos rendszer (4 PC)
- c) Folyamatok időbeli eloszlása multiprogramozott rendszerben

# Folyamatok állapotai

- Több folyamat , 1 CPU → látszólagos párhuzamosság.
- CPU kitüntetett erőforrás, egy adott pillanatban csak 1 program hajtodik végre.
- Gráffal lehet ábrázolni. Állapotok:
  - **Fut:** A CPU a folyamathoz tartozó utasításokat hajtja végre, CPU-nként egyetlen ilyen folyamat lehet.
  - **Várakozik, blokkolt:** A folyamat várakozni kényszerül, működését csak valamilyen esemény bekövetkezésekor tudja folytatni. Több ilyen is lehet a rendszerben.
  - **Futásra kész:** Minden feltétel adott, a CPU éppen foglalt. Több ilyen is lehet a rendszerben.

# Folyamatok állapotai

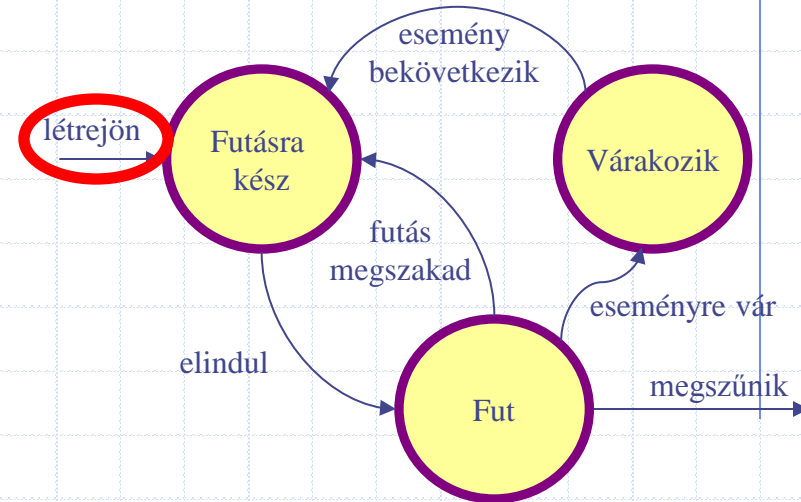


## Folyamatok állapot-átmeneti gráfja

# Folyamatok állapotátmenetei

- **Folyamat létrehozása**
- **Folyamat befejeződése**
- **Eseményre vár**
- **Esemény bekövetkezik**
- **Folyamat elindul**
- **Futás megszakad**

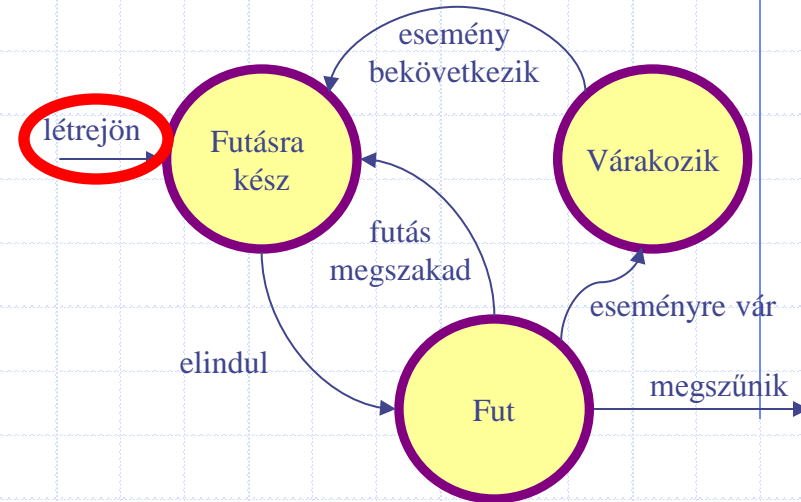
# Folyamat létrehozása 1.



Mikor jön létre egy folyamat?

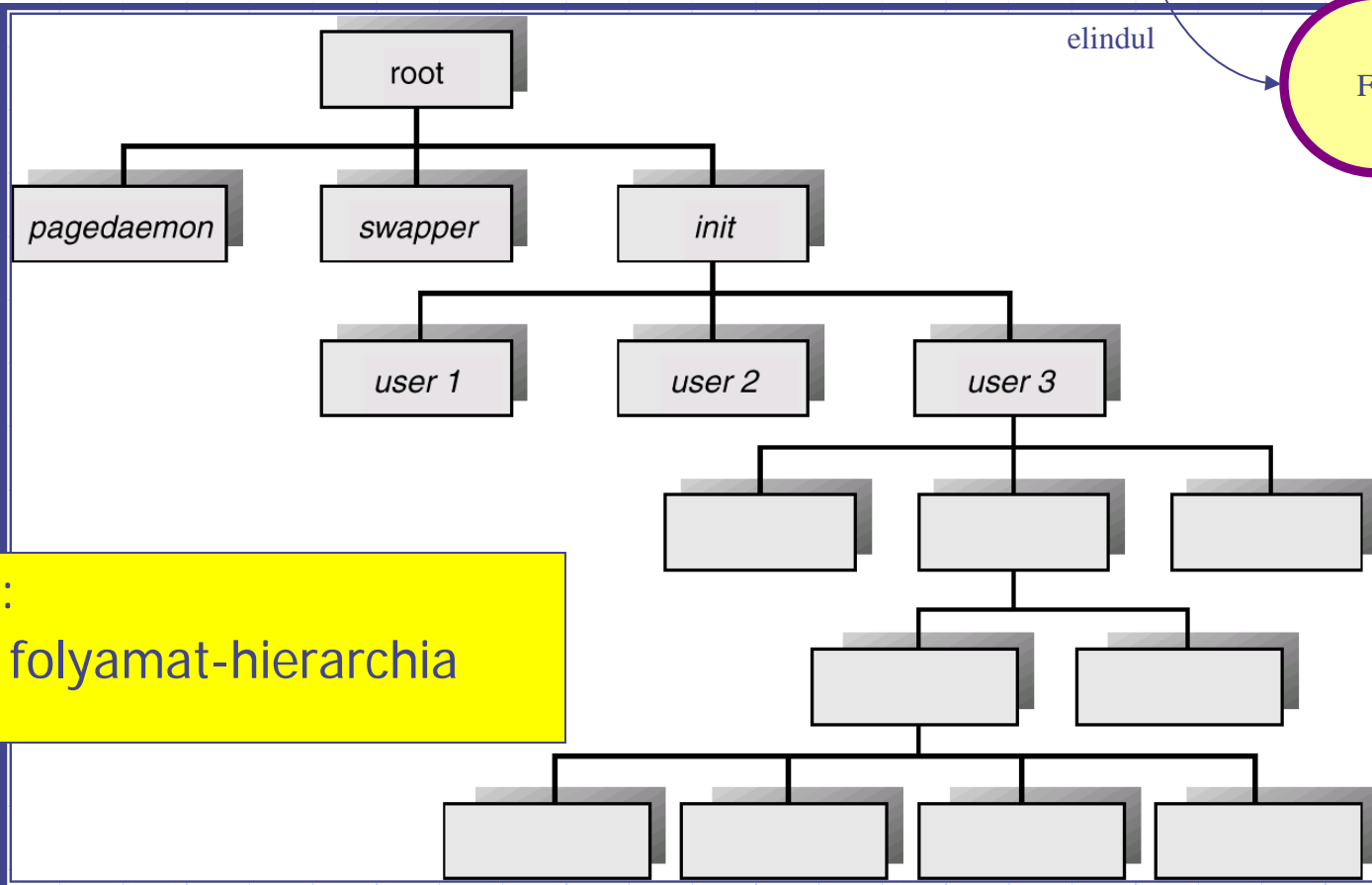
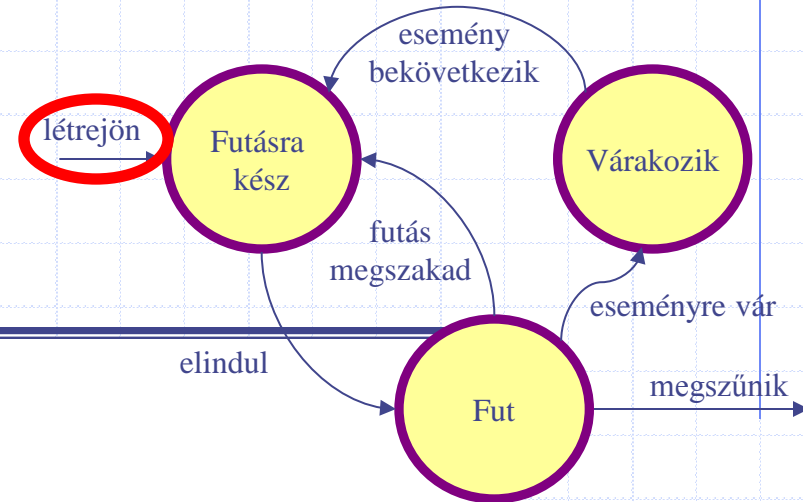
- Rendszer elindításakor (boot)
  - Pl. felhasználói interfészek, démonok
- Folyamatot létrehozó rendszerhívás hatására (pl. fork)
  - Egy folyamat gyermek folyamatot hozhat létre
    - Hierarchia (UNIX): „process group”
    - Minden folyamat egyenlő (Windows)
- Felhasználó
  - Program indítása
- Új batch-job indítása
  - Köteget rendszereknél OS dönti el, mikor kezdhet futni egy új munka

# Folyamat létrehozása 2.



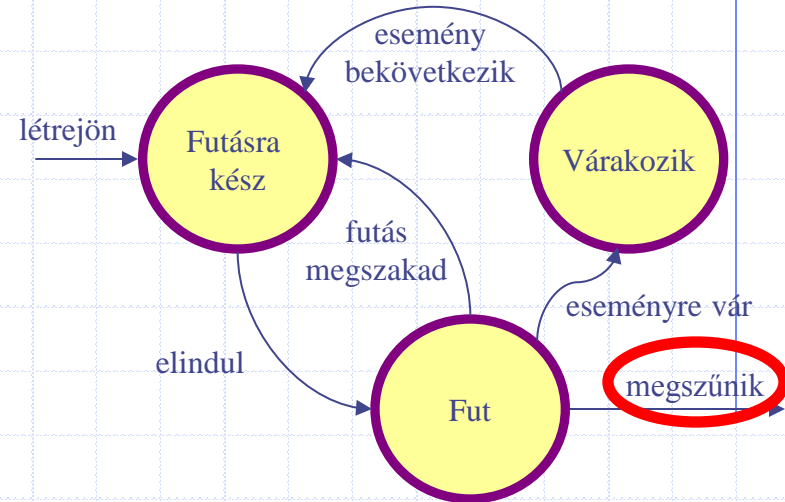
- Létrehozó folyamat: *szülő*, a létrejöttek a *gyerekek*.
- Leszármazási reláció, hierarchikus struktúra.
- Erőforrás kell neki
  - OS-től kapja
  - szülő erőforrásain osztozik
- Szülőtől paramétereket kap (befolyásolja a futását)
- Gyerek a szülővel párhuzamosan fut, vagy bevárja a gyerekének, gyerekeinek befejeződését.
- Szülő azonosítja a gyerekét.

# Folyamat létrehozása 3.



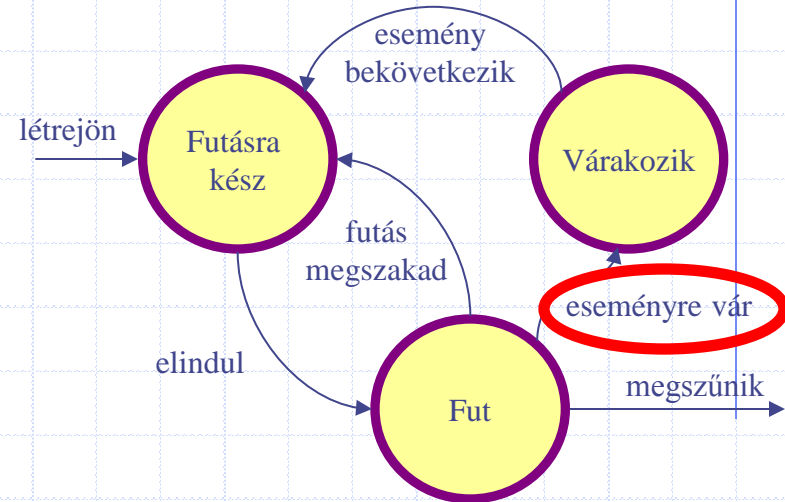
Példa:  
UNIX folyamat-hierarchia

# Folyamat befejeződése



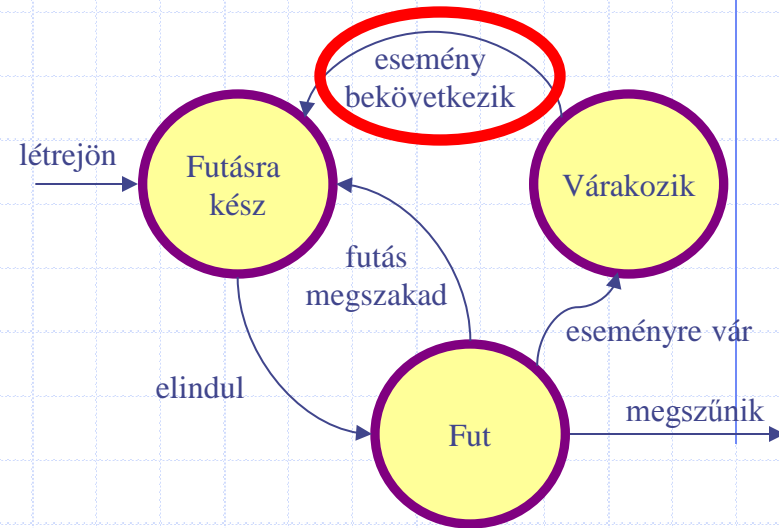
- Önszántából
  - végrehajtotta utolsó utasítását
  - hiba miatt leáll
- OS vagy rokon (általában a szülő) leállítja
  - Hibás utasítás, erőforrás használat túllépése
  - kill utasítás (másik folyamat)
- Erőforrások felszabadulnak
  - attól függően, hogy kitől kapta: felszabadul vagy szülőhöz kerül
- Szülőnek információt adhat vissza.

# Eseményre vár



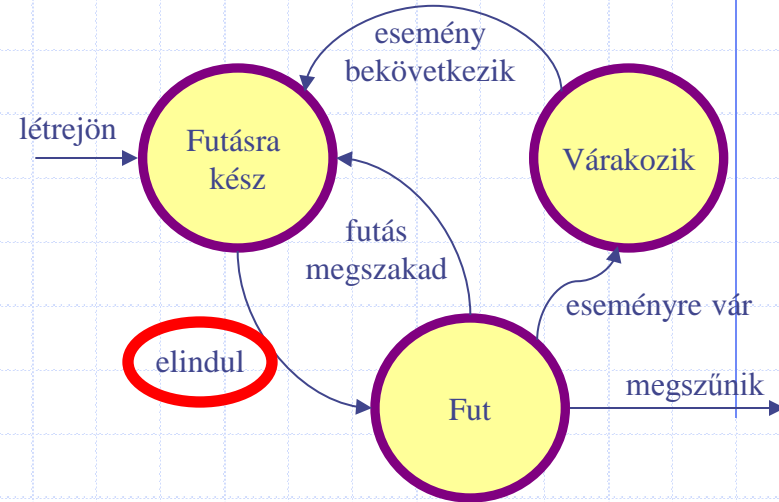
- Fut → várákozik
- Valamit kér, amire várnia kell
  - jelzés,
  - erőforrás, stb.
- OS feljegyzi, hogy ki mire vár.
- Több folyamat is várhat ugyanarra.

# Esemény bekövetkezik



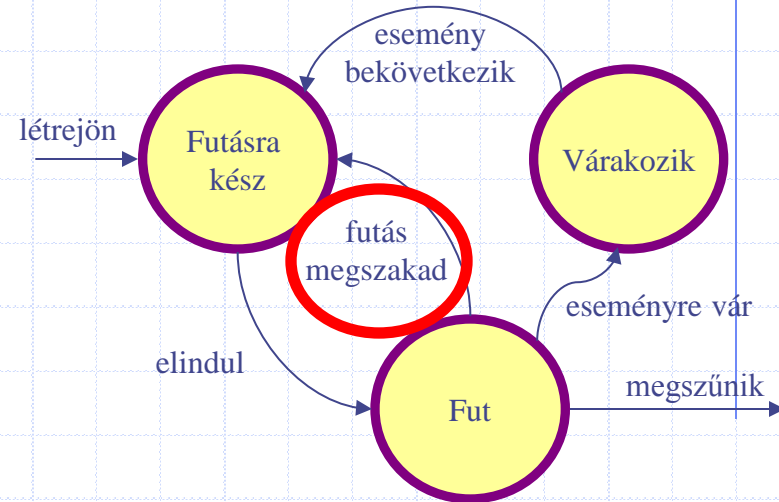
- Várákodik → futásra kész
- A várt esemény bekövetkezett
- Még nem fut!

# Folyamat elindul



- Futásra kész → fut
- ha a CPU felszabadul, *egy* folyamat futhat
- kiválasztás kritériumok alapján (**CPU ütemezés**)

# Futás megszakad

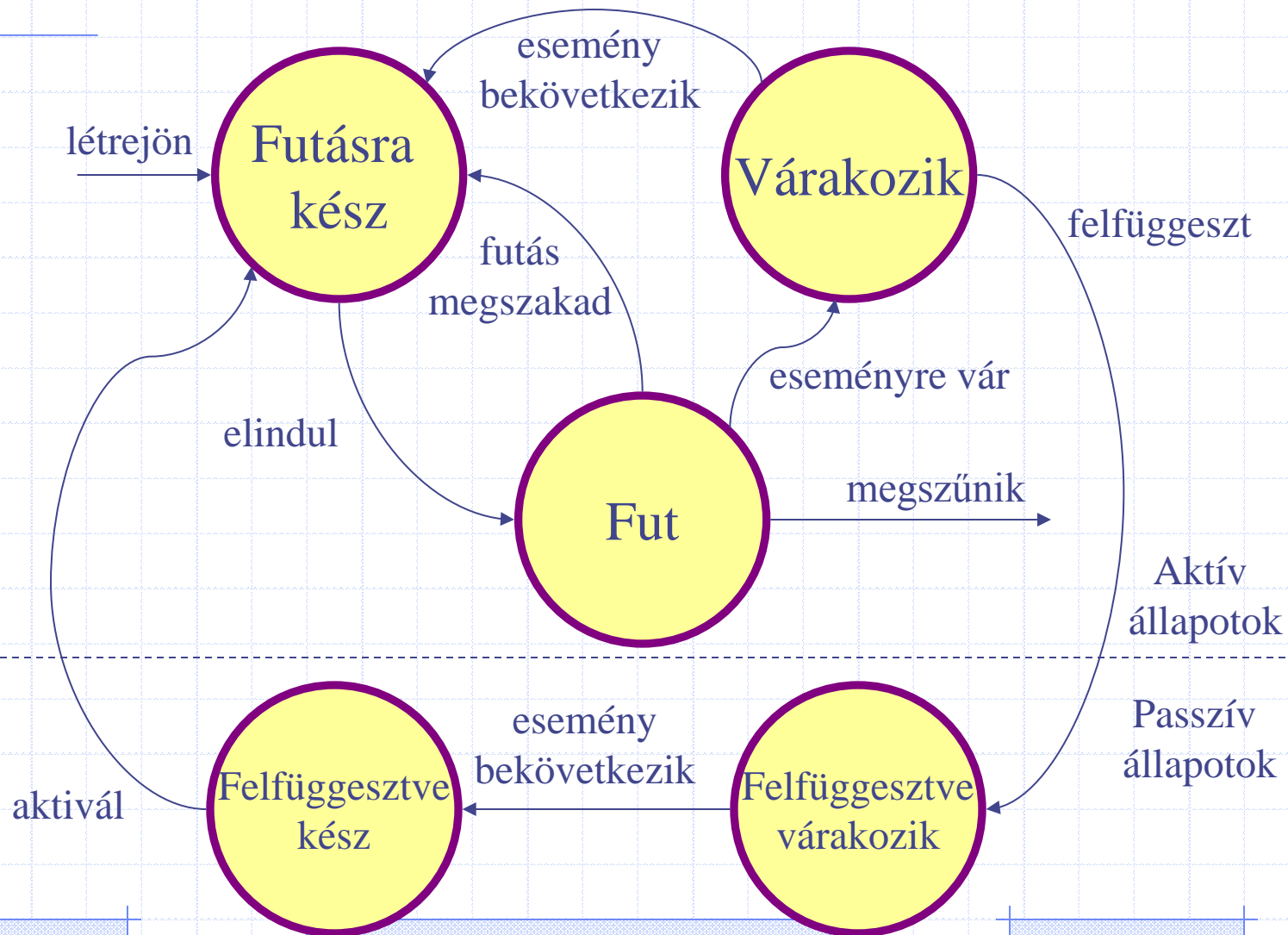


- Fut → futásra kész
- Önként lemond a CPU-ról.
  - Pl. kooperatív viselkedés: hosszú feladatok esetén újraütemezést kér.
- OS elveheti a CPU-t, még akkor is, ha a folyamat egyébként nem kényszerül várakozásra (*preemptív ütemezés*)
  - Pl. időosztásos rendszerek. Túl sok ideje futott. Óra megszakítás jelzi az eseményt.

# Folyamatok kibővített állapotai

- Bővítés: az OS felfüggeszthet folyamatokat (középtávú CPU ütemezés)
  - a rendszer túl van terhelve (sok program vetélkedik a futás jogáért, vagy a tár túlzottan megtelt, stb.)
  - vészhelyzet esetén
  - felhasználó kezdeményezésére
- A felfüggesztett folyamatok erőforrásaikat elvesztik (de a rendszer számon tartja őket, később folytatódnak)
- Két új állapot:
  - felfüggesztve vár
  - felfüggesztve kész

# Folyamatok kibővített állapot-átmeneti gráfja



# Új állapotátmenetek

- **Felfüggeszt**
  - OS felfüggeszti (futásra kész vagy várakozik állapotból)
  - erőforrásokat elveszi (pl. memória), néhányat megtarthat (pl. nyomtató)
- **Aktivál**
  - Erőforrásokat visszaad
- **Felfüggesztve várakozik → felfüggesztve futásra kész**
  - Esemény bekövetkezik, de CPU-t nem kaphat
- A felfüggesztve vár → várakozik átmenetnek nincs értelme (tovább várakozna, de lekötné az erőforrásokat)

## 2.2. Környezetváltás

- Környezet váltás (kontext switch):
  - a futó folyamat elhagyja a futó állapotot,
  - egy futásra kész pedig elindul.
- Az átkapcsolás zökkenőmentes legyen: állapotjelzőket meg kell őrizni
  1. Folyamat állapota.
  2. Végrehajtó gép állapota.
- Multiprogramozásnál fontos a gyors átkapcsolás (hatékonyság).

# A folyamat állapota

Állapotváltozók:

- programkód
- változók aktuális értéke
- verem tartalma
- hol tart a program végrehajtása (programszámláló)
  - ezt a végrehajtó gép tartalmazza!
- Tárban több folyamat van, így a folyamatok állapotjelzői megmaradnak a tártartalom megőrzésével. Csak a *végrehajtó gép* állapotát kell menteni.

# A végrehajtó gép állapota

- A rendszer rétegszerkezete miatt több szintű. A legegyszerűbb eset, amikor a folyamat kódja csak gépi utasításokat tartalmaz: hardver-szoftver határfelület, legtöbbször az OS alapszolgáltatásait is tartalmazó virtuális gép határfelülete.
  - CPU regiszterek
  - OS változók
    - rendszertáblák,
    - memóriakezelési információk,
    - periféria hozzárendelések, stb.
- A végrehajtó gép (OS) a folyamat környezete, a gép állapotjelzőinek összességét kontextusnak nevezzük.

## 2.3. Folyamatleírók, I/O leírók

- Speciális adatszerkezetek
- Az OS-nek a folyamatok és az I/O egységek kezeléséhez szükséges adatait tárolja
  - Process Control Block, PCB
  - Input Output Control Block, IOCB

# Folyamatleíró blokk

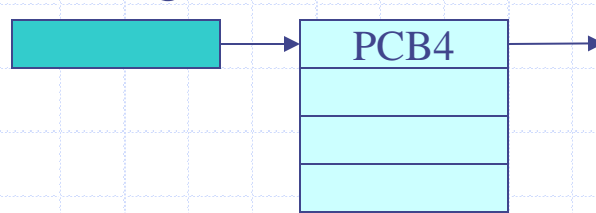
## Process Control Block, PCB:

- folyamat azonosítója
- szülők, gyerekek azonosítója
- folyamat állapota
- folyamathoz tartozó összes tárterület leírása (mutatók, virtuális tárkezeléshez tartozó adatok, cím transzformáció)
- a folyamat által használt egyéb erőforrások leírása (pl. nyitott állományok)
- regiszterek tartalma
- várakozó folyamatoknál: várt esemény leírása
- ütemezéshez információk (prioritás, várakozási idő)
- statisztikák

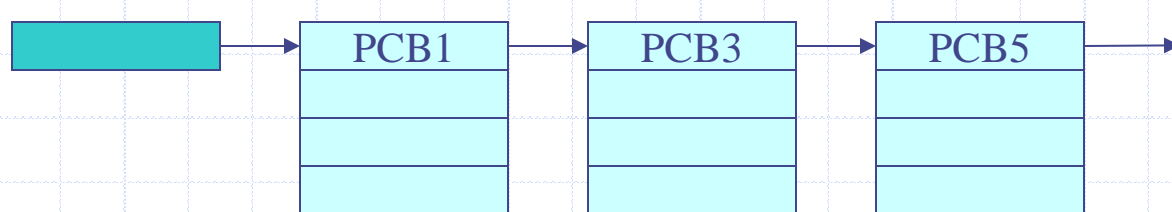
# Folyamatleírók kezelése

Láncolt listák:

- Futó folyamat



- Futásra kész folyamatok



# I/O műveletek leírása

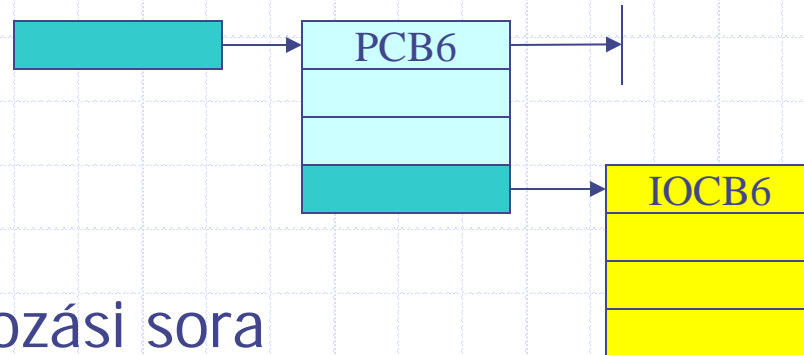
Input Output Control Block, IOCB:

- Az I/O művelet végrehajtásához tartozó minden adat:
  - művelet kijelölése (írás, olvasás, ...)
  - tárterület címe (ahonnan, ahova a művelet végrehajtandó)
  - I/O készülék egyéb adatai (mágneslemez szektorcíme, stb.)
  - átvihető adatmennyiség
  - állapotjelző
  - stb...

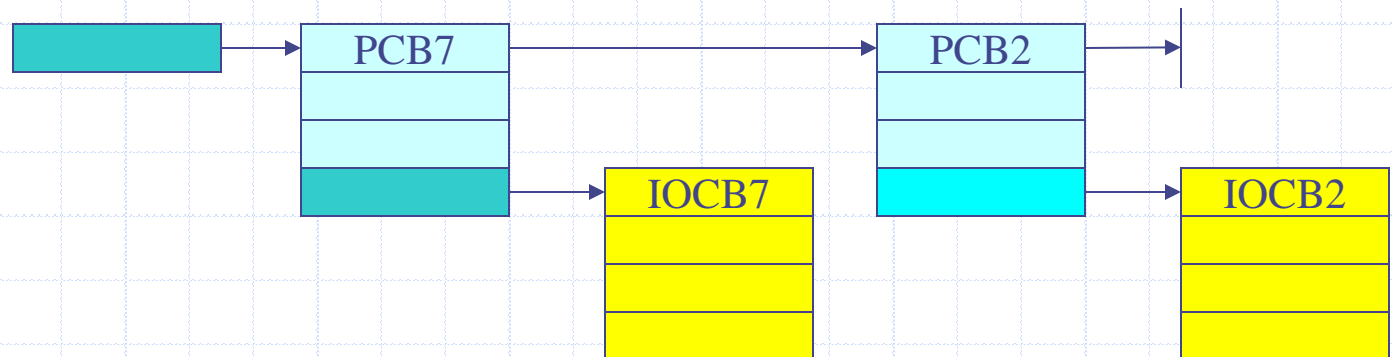
# I/O leírók kezelése

Az IOCB a PCB-hez fűződik:

- 1. periféria várakozási sora



- 2. periféria várakozási sora



# I/O műveletek végrehajtása

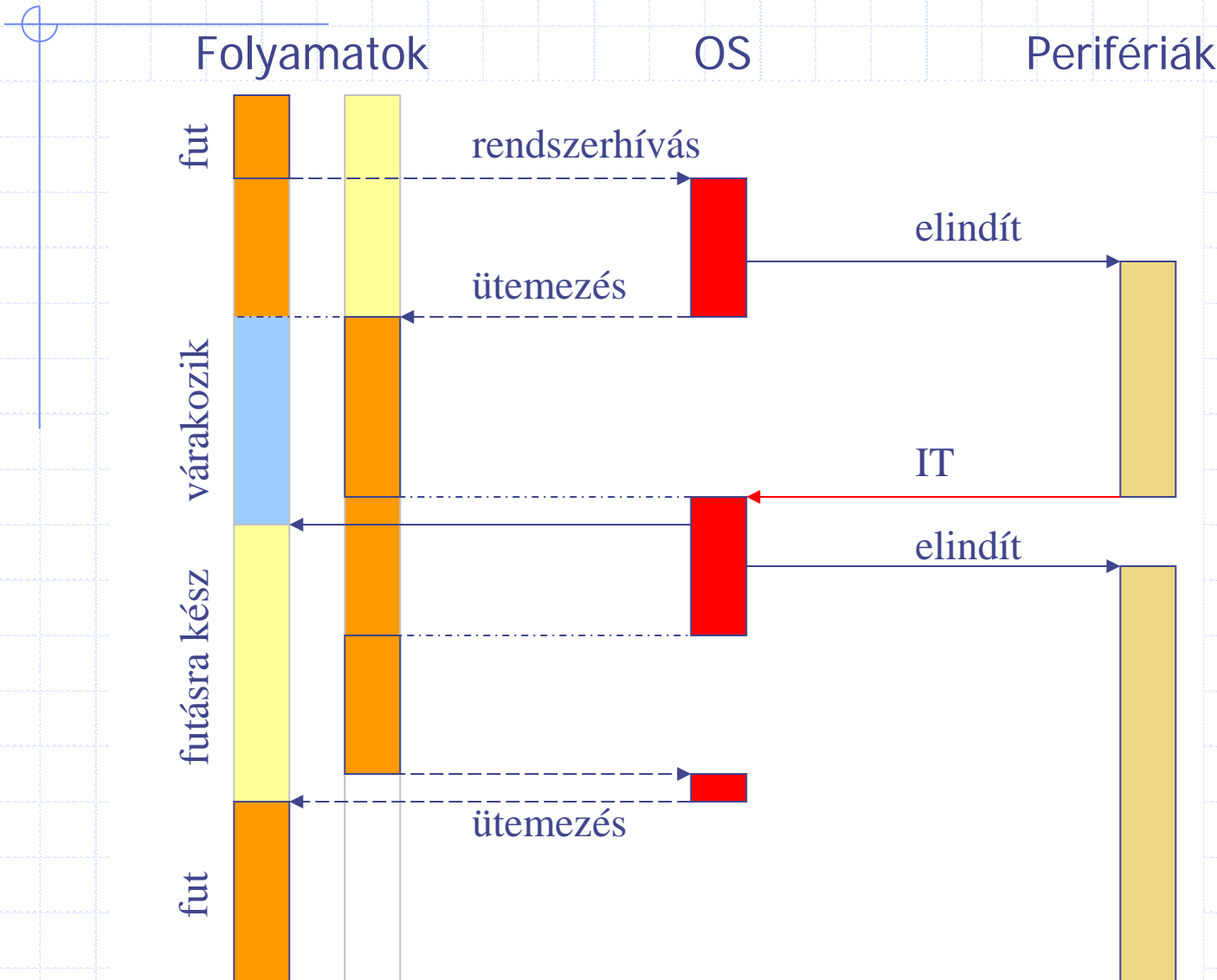
## Folyamat:

- Kitölti az IOCB-t
- Rendszerhívás, paraméter az IOCB
- Láncolja az IOCB-t a PCB-hez
- PCB-t befűzi a periféria várakozási sorába
- Ha a sor üres, indítja a perifériát az IOCB paramétereivel
- Folyamatot várakozó állapotba teszi
- Újraütemez és visszatér (másik folyamatra)

## Periféria

- Feladat végeztével megszakítást okoz
- IOCB-be a végrehajtás eredményére utaló jelzést ír (helyes/helytelen)
- PCB-t a futásra kész állapotba helyezi
- Ha van még várakozó a perifériára, akkor a perifériát indítja
- Újraütemez
- visszatér

# I/O műveletek végrehajtása



## 2.4. Szál (thread)

- Folyamatokhoz hasonló fogalom
- Nincs saját memória, saját erőforrás
  - csak a *regiszterek* és a *verem* sajátja (+állapot)
  - *minden más* közös a folyamatával
- Előny:
  - Gyors váltást tesz lehetővé
  - Osztott erőforrások (memória!)
- *Állapot-átmeneti gráfja megegyezik a folyamatoknál tárgyalttal.*

# Szálak és folyamatok 1.

Minden folyamatnak van:

Címtér

Globális változók

Megnyitott fájlok

Gyermekek folyamatok

Várt események listája

...

Minden szálnak van:

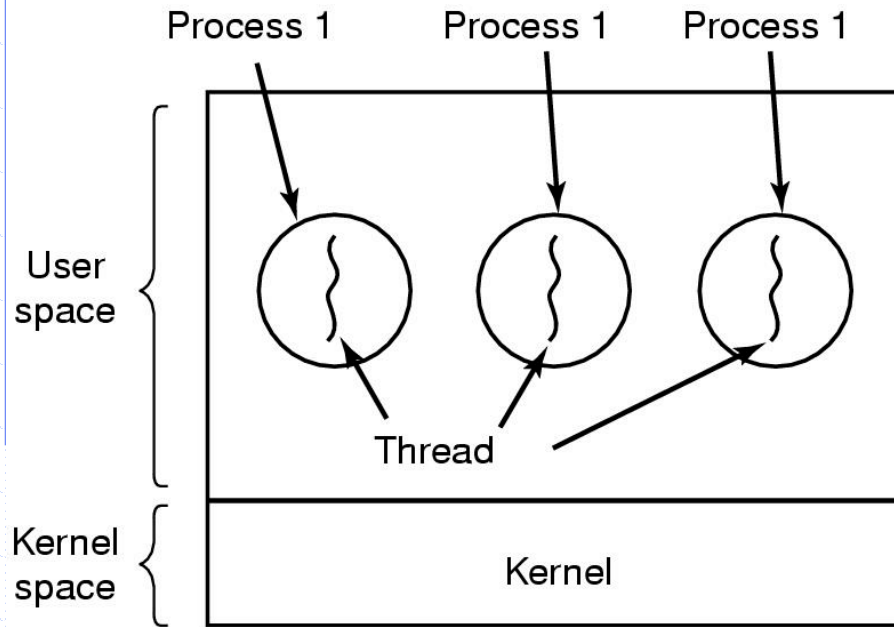
Programszámláló

Regiszterek

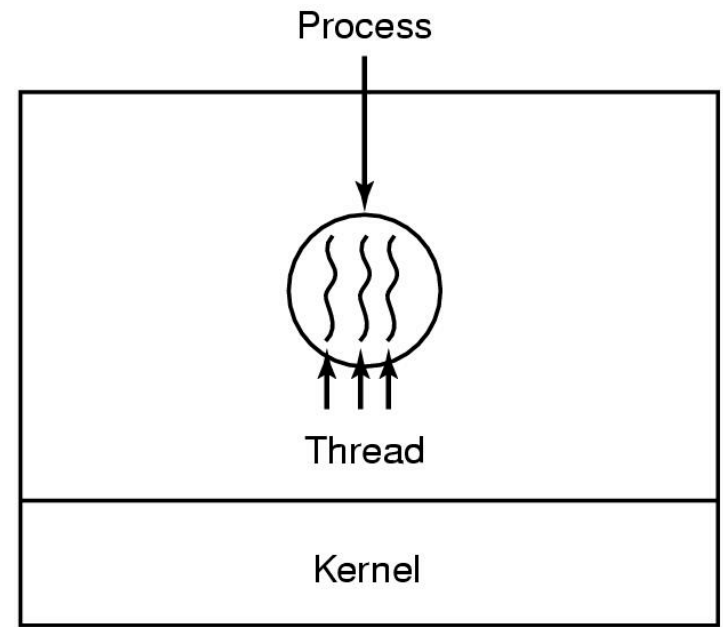
Stack

Állapot

# Szálak és folyamatok 2.



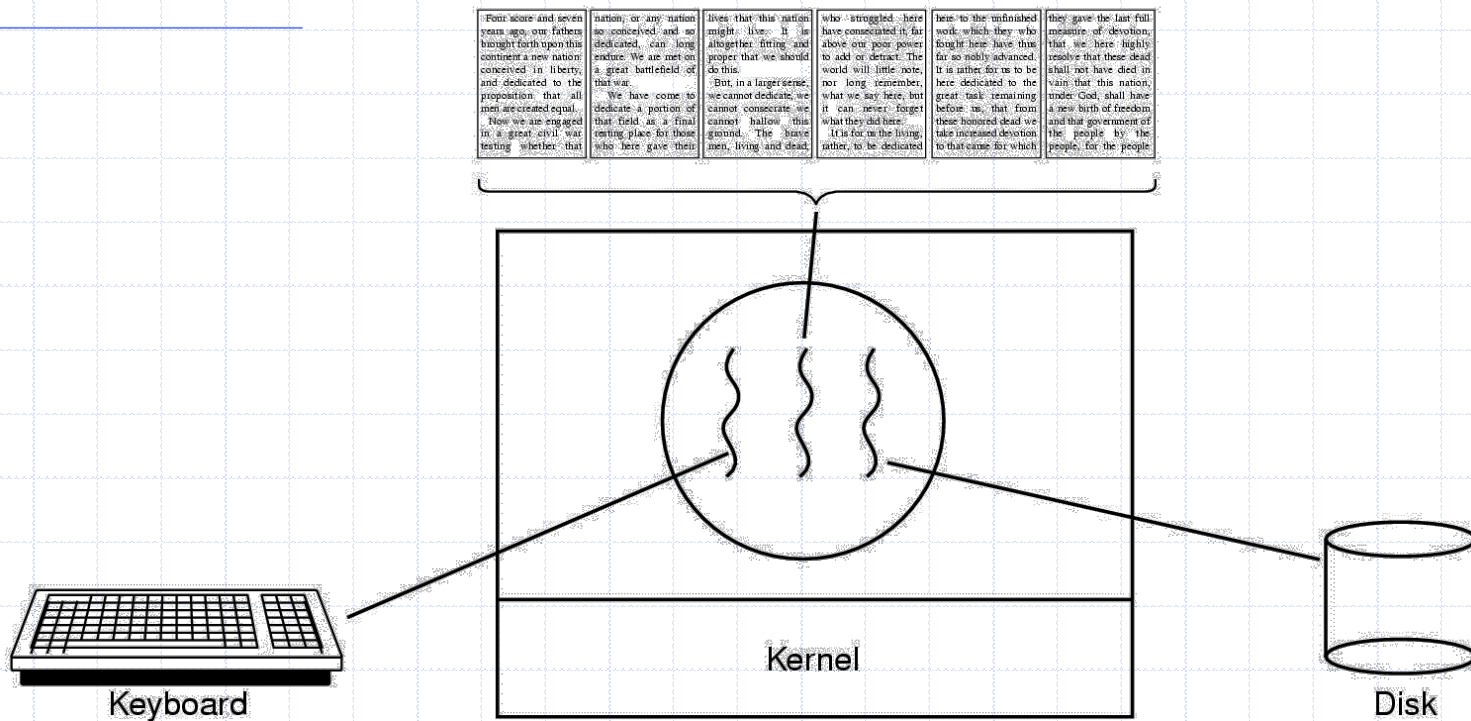
(a)



(b)

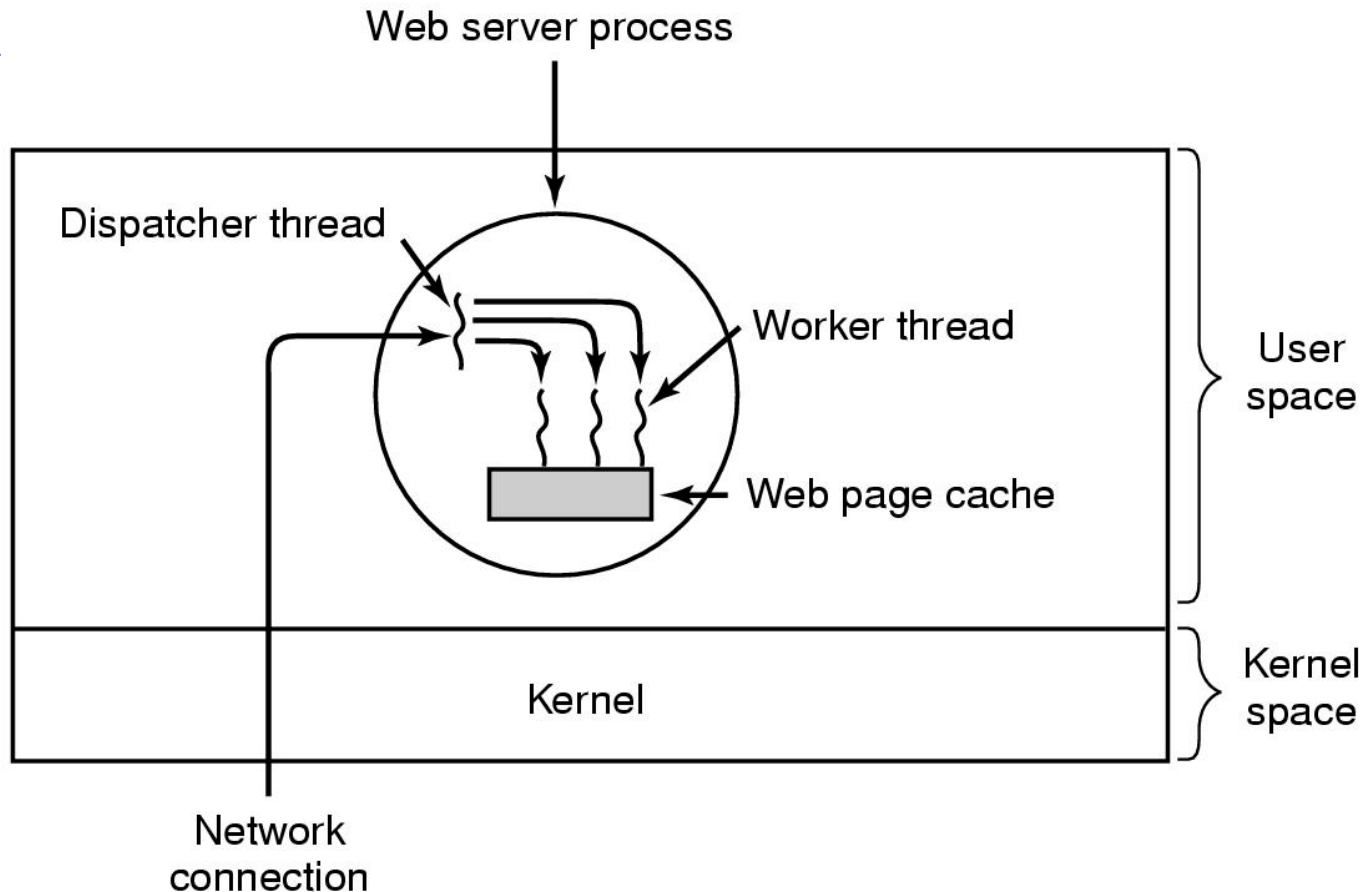
- a) Három folyamat egy-egy szállal
- b) Egy folyamat három szállal

# Példa: szövegszerkesztő



- Billentyűzet-kezelő szál
- Tördelő szál
- Automatikus mentés szál

# Példa: web-szerver 1.



- *Dispatcher* szál: fogadja a kérést és kiosztja a munkát
- *Worker* szál: teljesíti a kérést

# Példa: web-szerver 2.

```
while (TRUE) {  
  get_next_request(&buf);  
  handoff_work(&buf);  
}
```

(a)

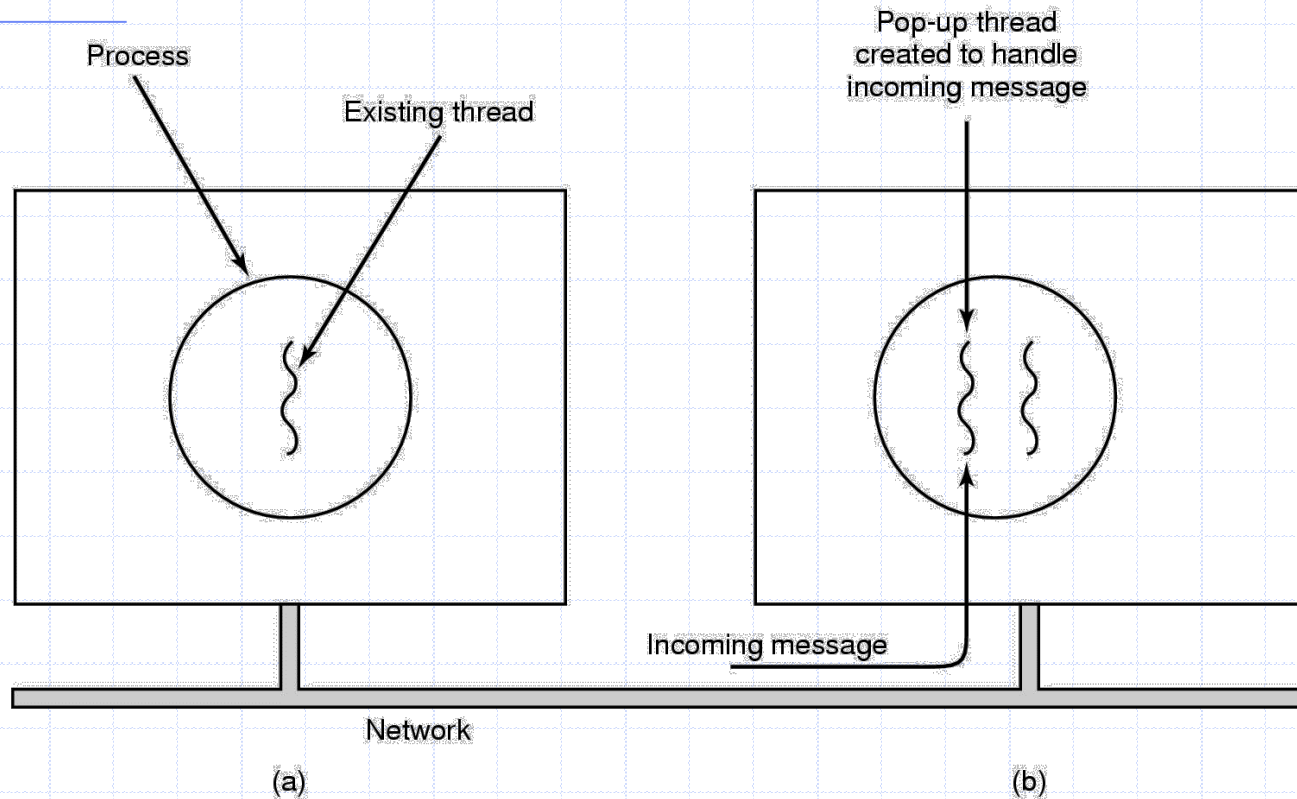
```
while (TRUE) {  
  wait_for_work(&buf)  
  look_for_page_in_cache(&buf, &page);  
  if (page_not_in_cache(&page)  
      read_page_from_disk(&buf, &page);  
  return_page(&page);  
}
```

(b)

a) *Dispatcher* szál

b) *Worker* szál

# Felbukkanó (pop-up) szálak



- a) Üzenet megérkezése előtt
- b) Üzenet megérkezése után

## 2.5. Megszakítások

- Interrupt
- Osztályai:
  - perifériák (berendezések állapotának változásai, átvitel befejezése)
  - belső hardver (timer)
  - utasítás végrehajtási hiba (pl. 0-val való osztás, memória hiba, virtuális tárkezelésben laphiba, stb.)
  - hardver hiba (pl. tápfeszültség)
  - szoftver megszakítás (trap), speciális gépi utasítás pl. rendszerhívás

# Prioritás

- Nem minden megszakítás azonos súlyú
  - pl. egy magasabb megszakíthat egy alacsonyabb prioritású kiszolgálást
- Megszakítások letilthatók / engedélyezhetők
  - csak korlátozott ideig hiszen pl. a perifériáknál az átvitelkor adatok veszhetnek el).

# Megszakítások kezelése

1. Nincs környezet váltás.
  - Megszakítás hatására OS rutin indul el, folyamat felfüggesztve, csak a kiszolgáló rutin által használt regisztereket mentjük el (ezek általában rövid kis programok).
2. Van környezet váltás (ritkábban, lassabb).
  - A megszakítás elindít egy arra váró folyamatot. A folyamat várakozóból rögtön futni fog (elkerüli a futásra kész állapotot).

# Megszakításkezelés lépései

- A futó folyamat megszakad, vezérlés az OS-nek
- Megszakított folyamat állapotmentése (1. vagy 2. módban)
  - pl. regiszterek mentése (van HW támogatás)
- Vezérlést a kiszolgáló rutin kapja meg
- Befejezés után állapot visszaállítás (1. vagy 2. mód)
- Megszakított - vagy egy másik - folyamat folytatja a futását.

# Megszakításkezelés támogatása

- Megszakításkezelést hardver támogatja
- A megszakított folyamat folytatásához ad segítséget
  - automatikus regiszter mentés, visszatöltés, folyamatok állapotainak megváltoztatása.