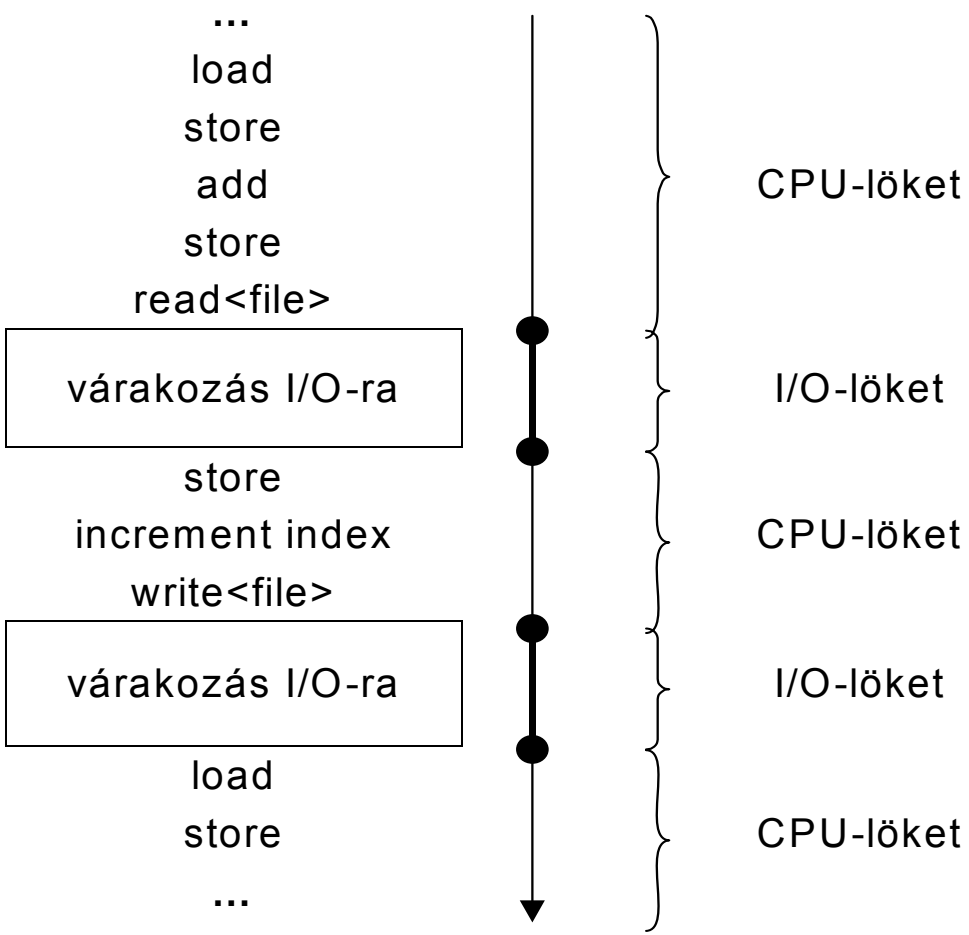
Operációs rendszerek

Folyamatok modellezése az operációs rendszerekben

CPU löket és I/O löket egy folyamaton belül



A löketek

- A folyamatok tipikus szerkezete:
  - processzor löket (egy processzor által végrehajtott utasítássorozat) után,
  - I/O löket, és ezek ciklikusan ismétlődnek.
- Az I/O művelet IT-s vagy DMA-s ezért a processzor addig más folyamatot futtathat. Ezáltal újabb I/O kérés lehetséges. Periféria verseny jön létre ezért ütemezés szükséges.
- CPU- ill. I/O-intenzív folyamatok egyensúlyban tartása.

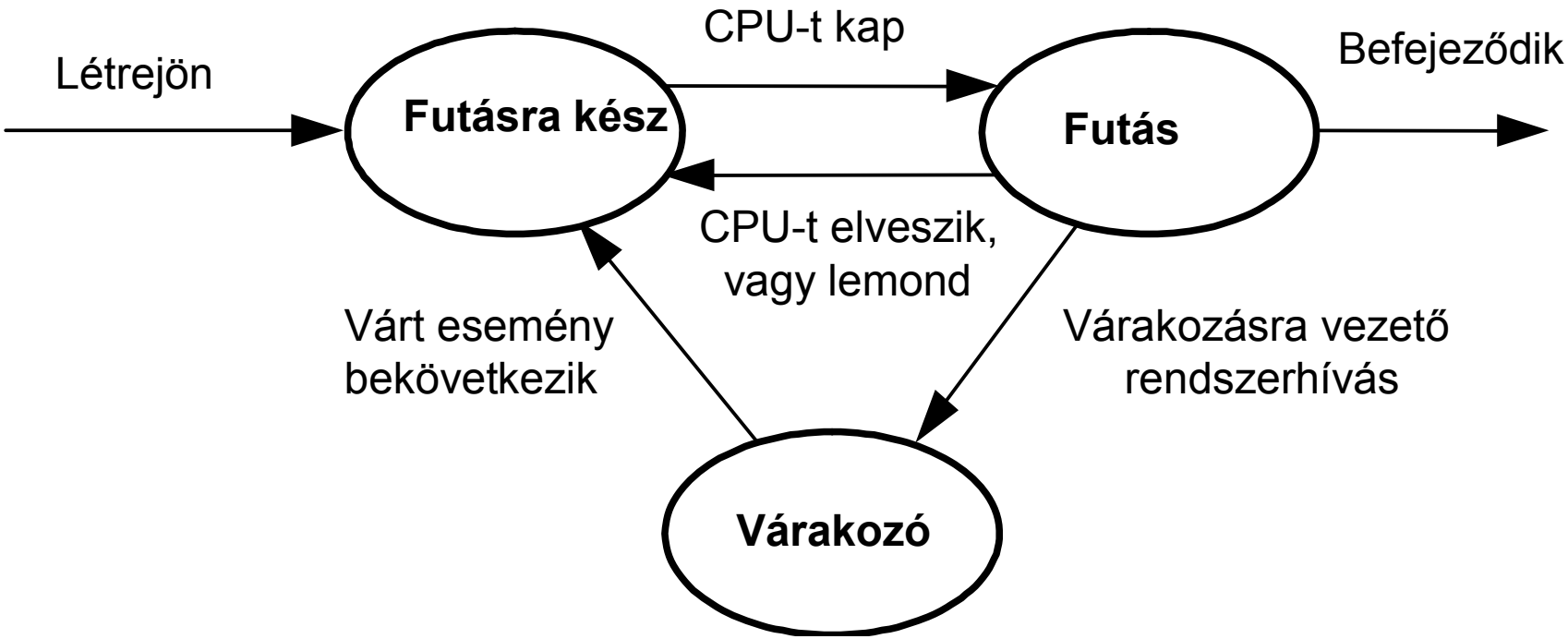
Folyamatkezelés

- Állapotmodell, - grafikus ábrázolása az állapot-diagram -, amelyik a folyamatok végrehajtásának menetére koncentrál.
- Tömegkiszolgálási ún. sorállási modell, - grafikus ábrázolása a sorállási diagram -, amelyik a folyamatok erőforrás-használatára koncentrál.

Folyamatok állapotmodellje

- Egy folyamat végrehajtásának dinamikáját írja le, egy hozzárendelt állapotjelzővel és az állapotátmeneti gráffal.
- Egy folyamat létrejöttekor:
  - fizikai memóriát kap, ahová az indításhoz szükséges kódrészlet betöltődik,
  - létrejönnek a változói,
  - erőforrásokat és indítási paramétereket kap,
  - az OPR nyilvántartásba veszi és futásra kész állapotba helyezi (állapotjelző beállítása).

Folyamatok állapotátmeneti diagramja



Folyamatok állapotmodellje

- Állapotjelzők:
  - futásra kész (ready): minden erőforrást birtokol az adott szakaszban, kivéve a CPU-t,
  - futó (running): 1 processzoros rendszerben az éppen futó folyamat, aktuális művelete,
  - várakozó (waiting): valamilyen feltétel teljesülésére váró folyamat (pl. I/O kiszolgálásra vár).

„Új” állapot, a folyamat megszületése

- Folyamat létrehozása: egy folyamatot általában egy már futó folyamat, "szülő" hozhat létre, emiatt a folyamatok között fennáll egy hierarchikus leszármazási reláció, amelyet az OPR-nek követnie kell:
  - erőforrások kijelölése a gyermek folyamat számára (vagy a szülő folyamattól, vagy az OPR-től),
  - paraméterek átadása a gyermek számára a szülőtől, amely paraméterek a gyermek futását szabályozzák, befolyásolják,
  - a szülő folyamat vagy párhuzamosan futhat a gyermek folyamattal, vagy be kell várnia annak futását, attól függően, hogy a "gyermek folyamat" eredményeire mikor és miképp van szüksége a szülőnek.

„Befejezett” állapot - A folyamat megszűnése

Ebbe az állapotba több okból juthat egy folyamat:

- A folyamat "önszántából" befejeződik, azaz annak utolsó utasítása is végrehajtódott.
- Egy folyamatot leállíthat az OPR, vagy az azt létrehozó szülő folyamat, esetleg valami másik folyamat. Ennek több oka lehet: pl. a folyamat hibás működése, vagy fölöslegessé válása, vagy mert több erőforrást használ, mint amennyit engedélyeztek számára.

A "befejezett folyamat" "elhalálozik": erőforrás igényei megszűnnek, a lekötött erőforrások visszaadódnak a folyamatot létrehozó folyamat (szülő vagy OPR) számára.

A befejezett folyamat megszűnésekor információt ad az őt létrehozó szülőnek a megszűnés tényéről, okáról.

Lehetséges állapotátmenetek

- Állapotátmenetek:
  - futásra készből futóba: az OPR ütemezője kiválasztja (ez minden olyan esetben kell, amikor az előző folyamat befejeződött vagy várakozó állapotba kerül),
  - futóból futásra készbe:
    - az OPR elveszi a folyamattól a CPU-t (preemptív ütemezés, pl. prioritási elven történő működtetés),
    - a folyamat mond le a futási jogáról (ütemezést kér, nem preemptív ütemezés, pl. együttműködő folyamatoknál az egyik engedi a másikat futni bizonyos pontoknál),

Lehetséges állapotátmenetek

- futóból várakozóba:
  - a folyamat olyan műveletet indított, amely CPU--tétlenséget okozna (pl. I/O műveletek),
  - meg kell várnia egy másik folyamattól érkező jelzést,
  - csak olyan erőforrás felhasználásával tudna "folytatódni", amelyet éppen egy másik folyamat használ.
- várakozóból futásra készbe:
  - a folyamat által várt esemény bekövetkezik (pl. I/O művelet vége).

- A gyakorlatban még további állapotok is léteznek: pl.: felfüggesztés, ”kilövés”.

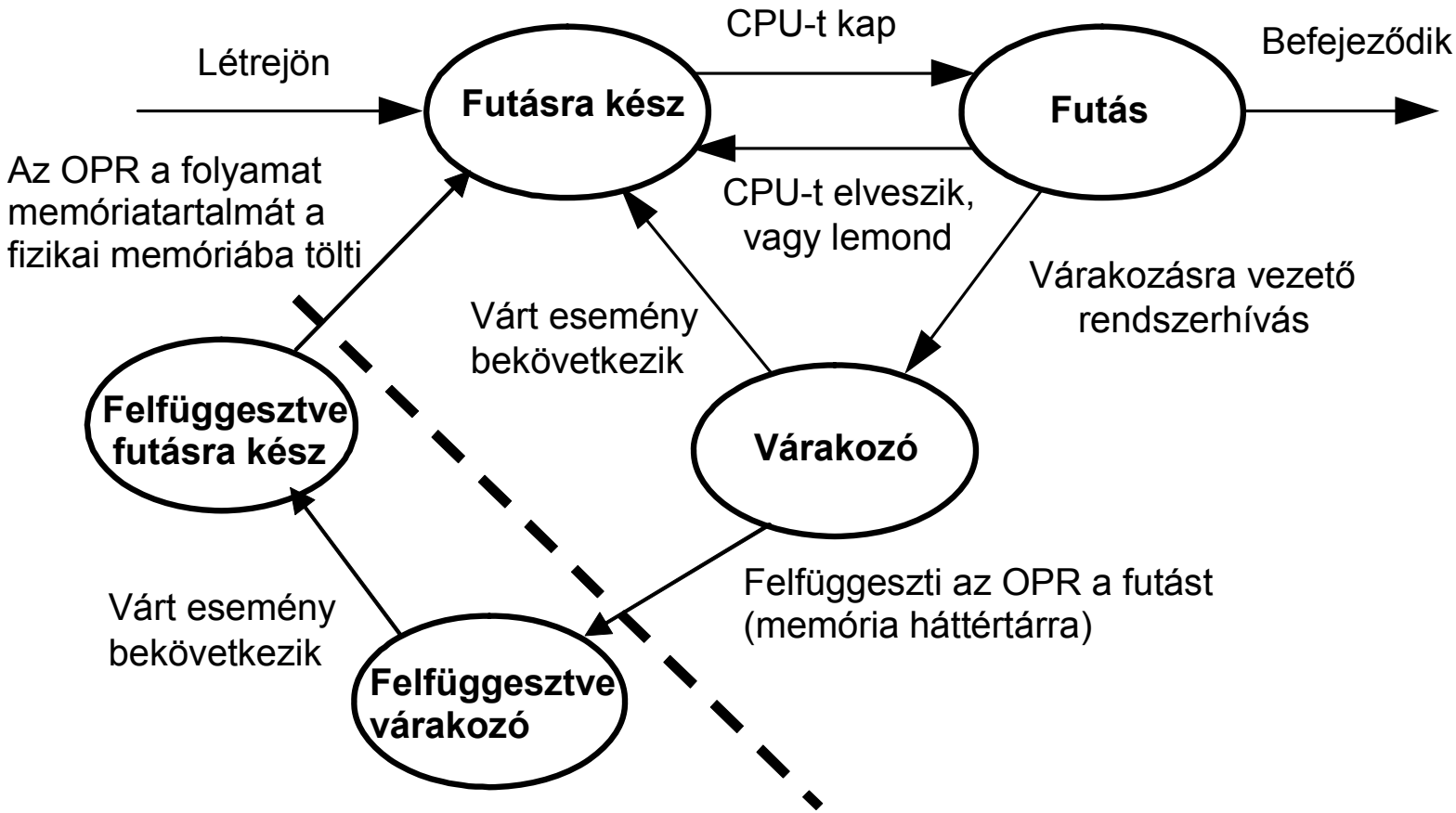
Gyakorlati esetek

- A szimultán élő folyamatok állapotlistájának bővülése egyre több munkával terheli az OPR-t, elsősorban az erőforrásokon való osztozás problémája miatt:
  - egy létező folyamatnak megszűntéig pl. véges mennyiségű memóriát kell biztosítani annak adatai, kódja, verme számára. Ha az erőforrások már kimerültek, azaz pl. nem lehet már elegendő memóriát kiosztani egy újabb, létrejövő folyamat számára, az OPR eredendően két dolgot tehetne:
    - megtagadja az új folyamat létrehozását,
    - meg kellene "ölnie" ("Process Termination") egy még élő folyamatot, hogy a felszabaduló memóriába be tudja illeszteni az új folyamatot.

Gyakorlati megvalósítás

- Ezért az OPR tartalmazza:
  - a felfüggesztés lehetőségét, amely a folyamat működésének időleges szüneteltetése.
    Így felszabadul(hat):
    - a memóriaterülete (akár tárcserével is),
    - és/vagy a foglalt, kritikus erőforrások csoportja,
  - az abortálás lehetőségét – a folyamat végső eltávolítása, általában rendellenes működéskor,
  - több várakozó állapotot létrehozása.

Folyamatok bővített állapotátmeneti diagramja



Környezetváltás

Folyamatok közti átkapcsolás: "Környezetváltás”.

- Szükséges a folyamat és a HW pillanatnyi állapotának az elmentése, annak érdekében, hogy a folyamat visszakapcsoláskor folytatható legyen ott, ahol a környezetváltáskor tartott.
- Lehetséges helyei az állapotátmenet diagramban:
  - a futó folyamat várakozni kényszerül,
  - a futó folyamat befejeződik.

Folyamatok sorállási modellje

- A folyamatok versengése a korlátos erőforrásokért.
- Addig áttekinthető amíg, a folyamatok szekvenciálisan használják az erőforrásokat (azaz ”elengedik” a birtokoltat mielőtt újat igényelnének).
- A folyamatoknak elegendő memória áll rendelkezésre.
- Valószínűségi változókkal és eloszlásfüggvényekkel számolja ki a terhelést (folyamatok belépési gyakorisága), és a kiszolgálást (erőforrások birtoklásának az ideje).

Folyamatok sorállási modellje

- A valószínűségi modell alapján számolható pl.: átfutási idő, várakozási idő, sorhosszúság.

Az ütemező neve a futási gyakoriságra utal!

Ütemezési pontok:

- Hosszú távú ütemező:
  - az új folyamatok indításáról dönt, amikor egy job lefutott,
  - kiválasztási szempont, hogy CPU- vagy I/O--intenzív folyamatok vannak-e túlsúlyban (előzetes infók a folyamatokról).

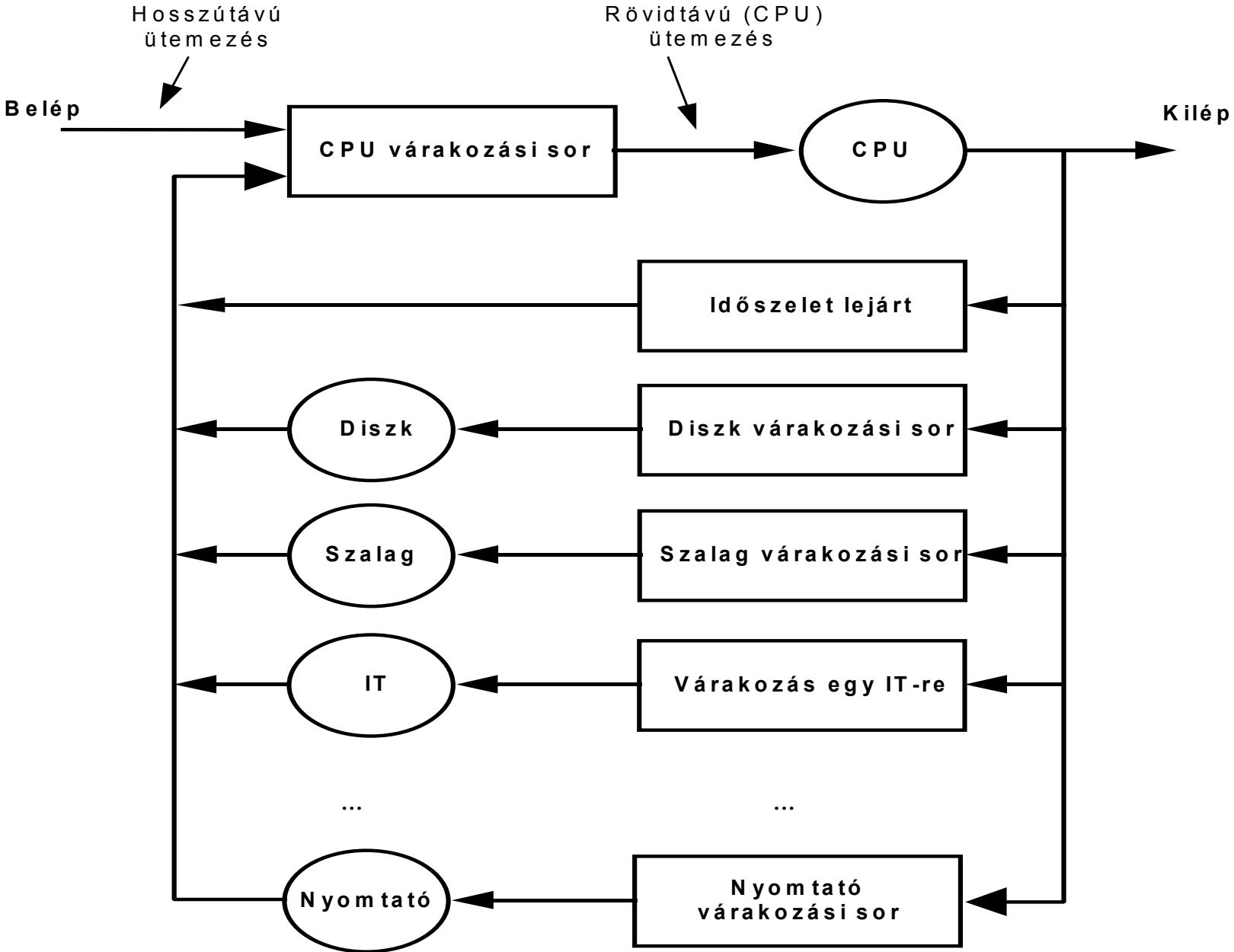
Ütemezési pontok folytatása

– Rövid távú vagy CPU ütemező:
  - minden processzor löket után lefut.
– Periféria ütemező:
  - leggyakrabban érkezési sorrendben (FIFO) történik a kiszolgálás.
– Középtávú ütemező:
  - akkor lép működésbe, ha a memória válik szűk keresztmetszetté, illetve ha ez a helyzet megszűnik,
  - memória hiány esetén, bizonyos folyamatokat felfüggeszt és a memória területüket háttértárra menti és a helyüket felszabadítja,

Középtávú ütemező folytatása

- ha újra van biztonságosan elégséges memória, akkor visszatölti őket,
- kiválasztási szempontok:
  - a CPU- és I/O-intenzív egyensúly megtartása,
  - hosszú várakozási idejű folyamatokat,
- a helye nem határozható meg a modellben, hiszen ez egy olyan esemény, ahol erőforrás kihasználás nem szekvenciális,
- így ebben a modellben nem is írható le.

Folyamatok sorállási modellje



Egy megvalósítási séma

- Az OPR a folyamatok kezeléséhez szükséges infókat egy speciális adatszerkezetben, az ún. folyamatleíróban (PCB: Process Control Block) tárolja. Ennek a tartalma:
  - a folyamat azonosítója,
  - a folyamat állapota,
  - a folyamat szülőjének és gyerekeinek azonosítója,
  - a folyamathoz tartozó összes tárterület leírása,
  - a folyamat által használt egyéb erőforrások leírása,
  - a regiszterek tartalma,
  - várakozás esetén, a várt esemény leírása,

Egy megvalósítási séma

  - ütemezéshez szükséges infók (pl. prioritás),
  - statisztikai infók.
- A sorállási modell ezen leírók láncolt listájaként valósul meg.
- Az I/O műveletek paraméterei egy másik leíróban az IOCB (Input-Output Control Block)-ben kerülnek rögzítésre. Ez tartalmazza a művelet végrehajtásához szükséges összes adatot (pl. írás/olvasás, tárterület cím, szektorcím).

Egy megvalósítási séma

- Az IOCB-k az I/O-műveletet kezdeményező folyamat PCB-jére fűződnek fel.
- Így az elindított I/O-műveletek és az azokra várakozó folyamatok összekapcsolása is megoldott.

I/O műveletek végrehajtása

- A folyamat kitölt egy IOCB-t (rendszerhívásokkal).
- I/O rendszerhívás történik (IOCB paraméterként átadódik).
- Erre az OPR:
  - az IOCB-t hozzáláncolja a folyamat PCB-jéhez,
  - ezt a PCB-t befűzi a periféria sorába,
  - ha a sor üres, indítási parancs a perifériának,
  - a folyamatot várakozó állapotba helyezi,
  - CPU-ütemezést hajt végre, majd környezetet vált,
  - visszatér, az új folyamatra.

I/O műveletek végrehajtása

- A CPU és a perifériák párhuzamosan működnek.
- Az átvitelek végét megszakítások (IT-k) jelzik, amelyeket az OPR kezel.

I/O megszakítás

- Az OPR ilyenkor:
  - az átvitel eredményére utaló jelzést ír a periféria várakozási sorának elején álló PCB-hez láncolt IOCB-be,
  - a sor elején álló PCB-t kifűzi a sorból, és átteszi a futásra kész sorba (a PCB-ben az állapotjelzőt átírja),
  - ha van még várakozó folyamat a periféria sorában, akkor a következő IOCB paramétereivel indítási parancsot ad a perifériának,
  - ha az ütemezés preemptív, CPU-ütemezést hajt végre,
  - visszatér.

I/O megszakítás

- Ezzel a szervezéssel az I/O-rendszerhívást kiadó folyamatnak, a rendszerhívást követő utasítása akkor hajtódik végre, ha már az átvitel befejeződött, és a CPU-ütemező őt választotta ki a futásra.