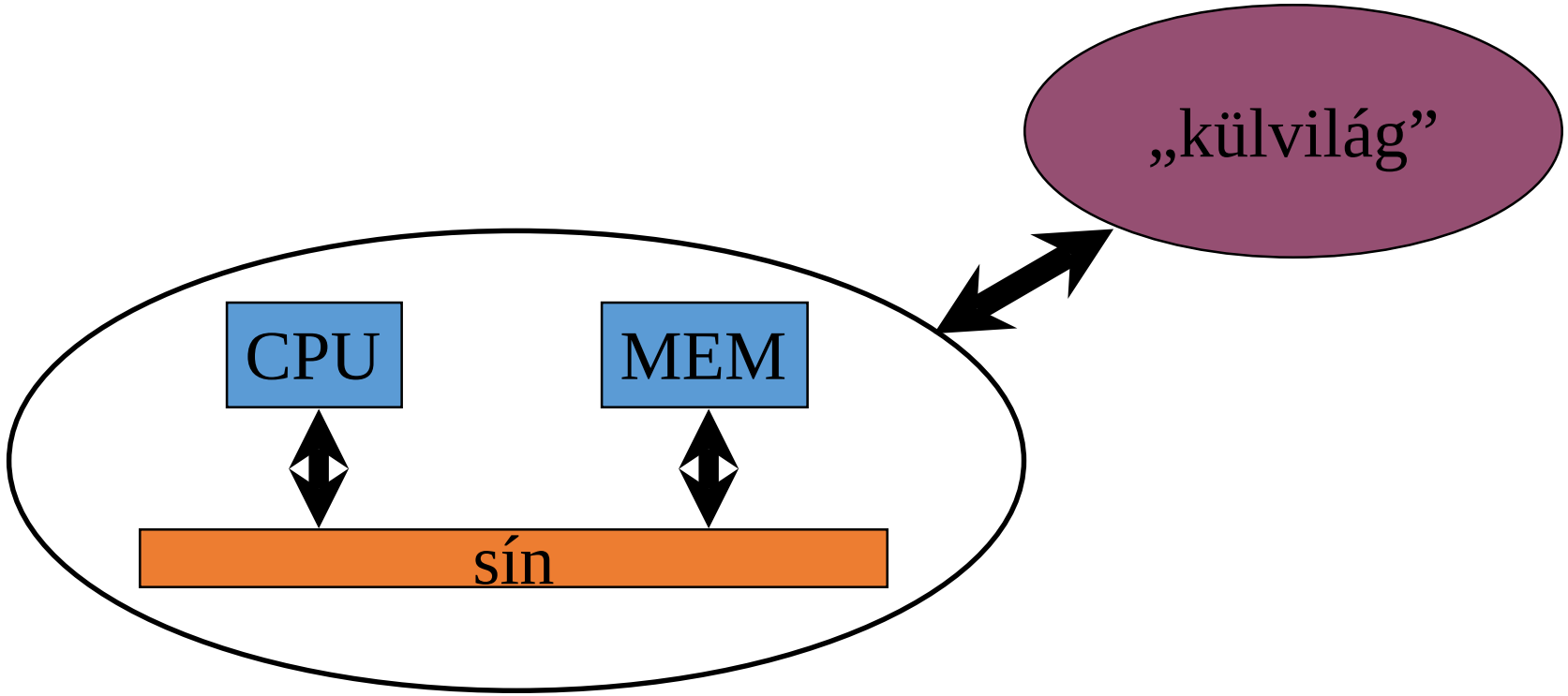
Operációs rendszerek 4.
I/O rendszerek

Tibi V 2019 IT alapok

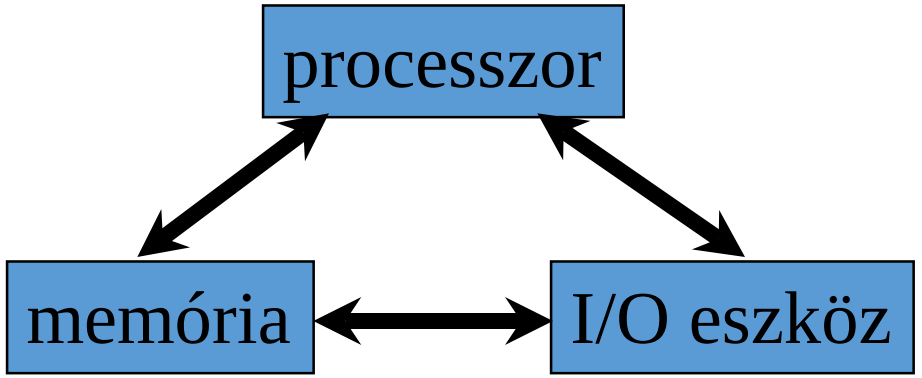
Kapcsolat a „külvilággal”

A kimeneti és bemeneti egység feladata: információ csere a CPU vagy a központi memória és a külvilág között...



A ki/bemeneti egység feladata

- Kapcsolatok kezelése
- Adatátvitel



ki/bemeneti eszköz
input/output device
I/O eszköz
periféria

Célok és problémák...

Célok :

- eszközök minél gyorsabb, „jobb” kiszolgálása...
- a CPU minél kisebb leterhelése...

Problémák :

- eszközök (CPU - perifériák) eltérő sebessége...
- eszközök különbözősége (kezelési mód)...

I/O rendszerek feladatai

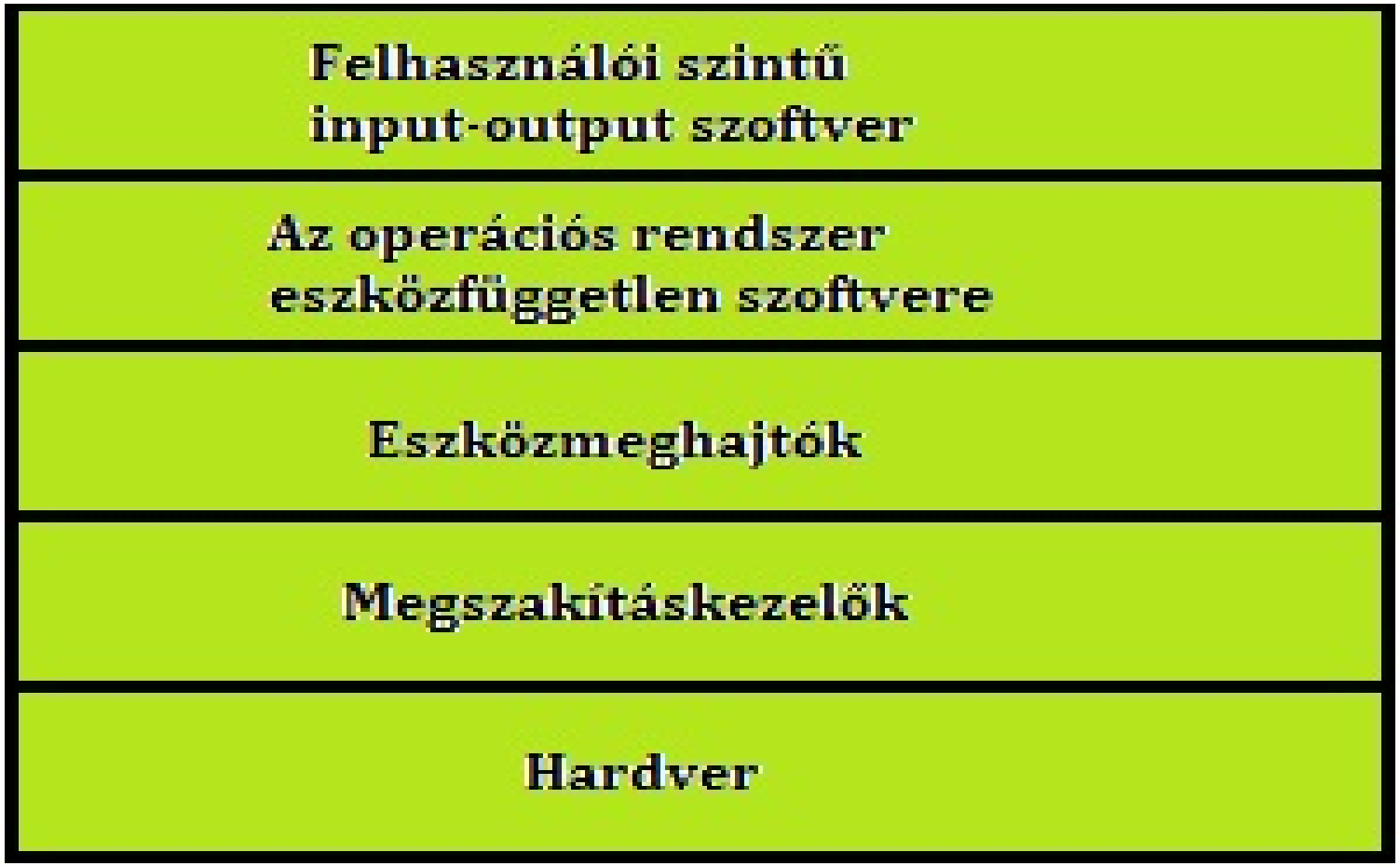
- Az operációs. fő feladatai közül az egyik, az input-output eszközök (azaz a beviteli-kiviteli eszközök, rövidebben I/O eszközök) vezérlése.
- Az input-output eszközök felé az operációs parancsokat kell kiadnia, és kezelnie kell a megszakítási kéréseket és a felmerülő hibákat is.
- Továbbá az operációs gondoskodnia kell az input-output eszközök és a rendszer többi része közötti kapcsolódási felületről, amelynek egyszerűnek és könnyen használhatónak kell lennie.

Az input-output rendszerek alapelvei

- Az input-output rendszerek alapelvei a következők:
- az eszközfüggetlenség,
- az egységes névhasználat,
- a hibakezelés,
- a szinkron (blokkolásos) és az aszinkron (megszakítás vezérelt) átviteli módszerek,
- a pufferelés,
- az eszközök megosztott és az ezzel ellentétes, monopol módú használata.

Az I/O rendszerek rétegei

- Az input-output rendszerek négy rétegbe vannak szerveződve.



Az eszközfüggetlenség

- Az input-output szoftverek tervezésénél az eszközfüggetlenség egy kulcsfogalom. Ez azt jelenti, hogy legyen lehetőség olyan program írására, amely hozzáfér bármely input-output eszközhöz anélkül, hogy előzetesen az adott input-output eszközt meg kellene adni. Például egy programnak képesnek kell lennie arra, hogy egy állományt inputként be tudjon olvasni bármilyen input-output eszközről anélkül, hogy a programot módosítani kellene a különböző input-output eszközök szerint.

Az egységes névhasználat és hibakezelés

- Az **egységes névhasználat** azt jelenti, hogy egy állomány vagy egy eszköz nevének egyszerűen egy jelsorozatnak vagy egy egész számnak kell lennie, függetlenül az adott input-output eszköztől.

- Másik kulcsfogalom az input-output szoftverek tervezésénél a **hibakezelés.** Ez azt jelenti, hogy a hibákat amennyire csak lehet, a hardver közeli szinten kellene kezelni. Ha a vezérlő olvasási hibát észlel, akkor megpróbálja a hibát javítani. Ha erre nem képes, akkor az eszközmeghajtónak kell kezelnie a hibát, esetlegesen akár a blokk olvasásának megismétlésével. A magasabb szoftverrétegeknek csak akkor szabadna tudomást szerezniük a hibákról, ha azok az alacsonyabb rétegekben nem kezelhetők. Sok esetben a hiba javítása tökéletesen elvégezhető az alacsony szinteken is anélkül, hogy a magasabb szintek bármit is érzékelhetnének belőle.

A szinkron (blokkolásos) és az aszinkron (megszakítás vezérelt) átviteli

- A szinkron (blokkolásos) és az aszinkron (megszakítás vezérelt) átviteli módszerek fogalma. A fizikai inputok és outputok legtöbbje aszinkron jellegű, ami azt jelenti, hogy a processzor beindítja az átvitelt, és más feladatot lát el a megszakítás megérkezéséig. A felhasználói programokat viszont sokkal egyszerűbb úgy írni, hogy ha az input-output műveletek blokkoltak.
- Ez azt jelenti, hogy a program automatikusan felfüggesztődik addig, amíg az adat a pufferben elérhetővé nem válik.

Pufferelés

- A pufferelés. Gyakran egy input-output eszközről jövő adat nem tárolható közvetlenül az adat végső célhelyén. Például amikor a számítógépes hálózatról egy csomag érkezik,addig nem tudja, hogy hova helyezze el, amíg nem tárolja és vizsgálja meg „valahol” az adott csomagot. Továbbá a szigorúan valós válaszidejű eszközök esetében, amilyenek például a digitális audioeszközök is, az adatot előre el kell helyezni egy kiviteli pufferbe, hogy ne hátráltasson egy üres puffer feltöltésének ideje; ebből a célból kerülni kell a puffer túlcsordulását. A pufferezés ráadásul tekintélyes mennyiségű másolással is jár, és gyakran jelentősen befolyásolja az input-output teljesítményét is!

Az eszközök megosztott és monopol módú használata

- A legutolsó fogalom, ami az input-output szoftverek tervezésének a világában szóba kerülhet még, az eszközök megosztott és az ezzel ellentétes, monopol módú használata. Néhány input-output eszközt sok felhasználó használja ugyanabban az időben. Ilyenek például a lemezegységek. Nem okoz problémát az, ha azonos időben több felhasználó is megnyit állományokat ugyanazon a lemezen. Ugyanakkor más eszközök, mint például a szalagos egységek esetében egyszerre csak egy felhasználó számára elérhetőek az adatok.
- A monopol módú eszközök bevezetése számos problémát okoz, mint például a holtpontok. Így ismét az operációs rendszerre marad az, hogy kezelje mind a megosztva, mind a monopol módon használt eszközöket úgy, hogy a felmerülő esetleges problémákat is elkerülje.

Az input-output eszközök

- Az input-output eszközöket két csoportra bonthatjuk, ezek a következők:
- blokkos eszközök és
- karakteres eszközök.

A **blokkos eszközökön** olyan eszközöket értünk, amelyek az adott információt egy megadott méretű blokkokban tárolja, és a blokkok mindegyikét egy saját címmel látja el. A blokkos eszközök lényeges tulajdonsága, hogy minden egyes blokk írható és olvasható, a többi blokktól függetlenül. A leggyakrabban használt blokkos eszköz például **HDD, SSD**

A **karakteres eszközökön** olyan eszközöket értünk, amelyek vagy kibocsátják, vagy fogadják a karakterek sorozatát anélkül, hogy figyelembe vennének bármilyen blokkszerkezetet. Az ilyen eszközök nem címezhetők. Karakteres eszközöknek tekinthetőek a **nyomtatók**, a hálózati csatlakozási felületek, **az egerek**

I/O rendszerek fajtái

Programozott I/O

teljesen a CPU irányítja az I/O műveleteket

Közvetlen memória hozzáférés + **Megszakítás** (interrupt)

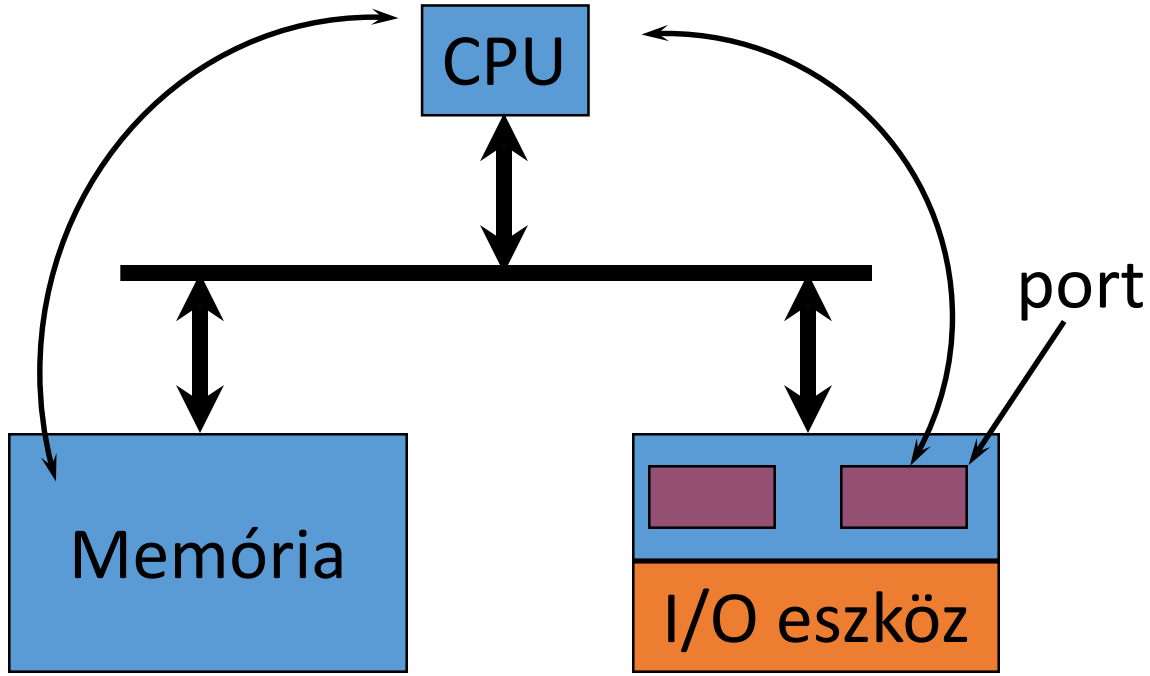
(DMA - Direct Memory Access)
CPU csak „elindítja”, aztán az I/O eszköz
a CPU „nélkül” végzi az átvitelt

I/O processzor

memóriához hozzáférés, CPU felfüggesztése,
I/O program végrehajtása

Programozott I/O

adatátvitel a CPU-n keresztül



- IO eszköz kiválasztása a címsínnel
- IO port: egy adott (memória) cím memóriára leképzett IO (memory mapped IO)

A programozott IO hátrányai

az IO átvitel sebességét attól függ, hogy a CPU milyen gyorsan tudja az IO eszközt vizsgálni és kiszolgálni...

a CPU (feleslegesen) sok időt „veszít” az az eszköz állapotának a vizsgálatával és az adatátvitellel...

ha több (vizsgálandó) eszköz van...

az adat a CPU-n halad keresztül, ahelyett hogy közvetlenül a memóriába jutna...

Megszakításos I/O

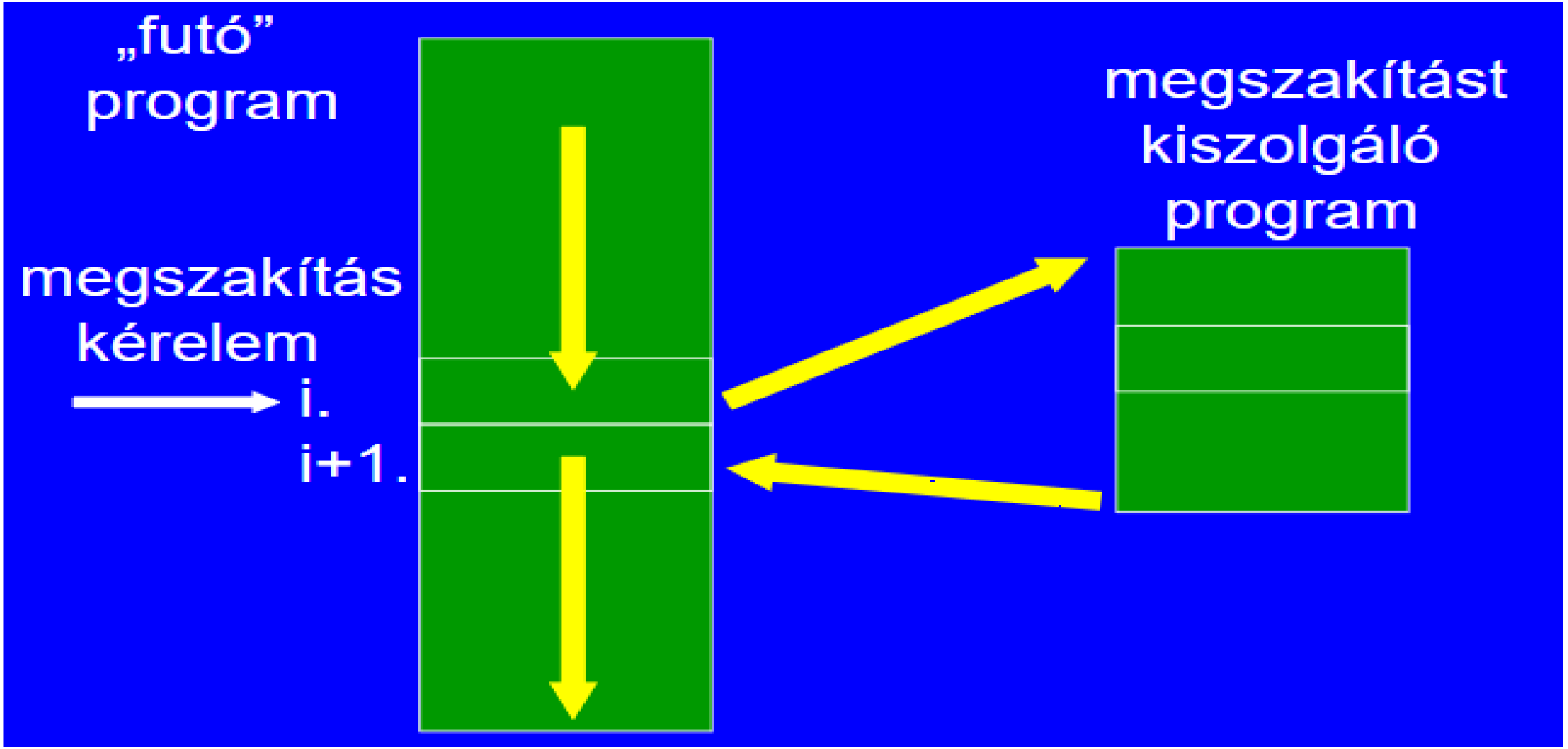
- az eszköz megszakítás kérelemmel jelez ha készen van... (CPU-nak nem kell várnia...)

IRQ : A processzort az aktuális feldolgozási művelet átmeneti megszakítására és a megszakításkezelő eljárás elindítására utasító jel.

Az I/O eszközök a megszakításvezérlőnek jelzik a megszakításkérelmüket (IRQ – Interrupt Request), ami a megszakításkérelmet továbbadja a CPU-nak. Ezután a megszakításvezérlő újabb megszakításkérelmet nem továbbít addig, amíg nem nyugtázzák.)

- sok nagysebességű eszköznél (+blokkátvitel) használhatatlan...

A megszakítás



Megszakítási események kiváltója

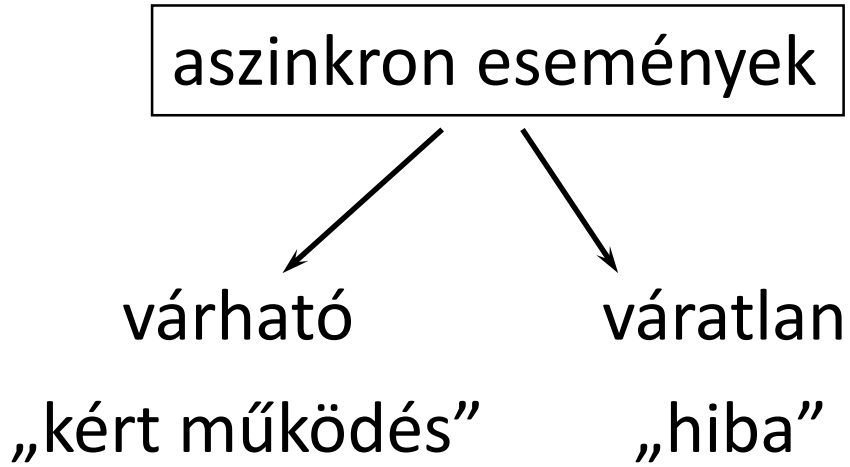
program

- végrehajtás közbeni hiba
- 0-val osztás
- túlcsordulás
- lapváltás

szinkron események

hardver

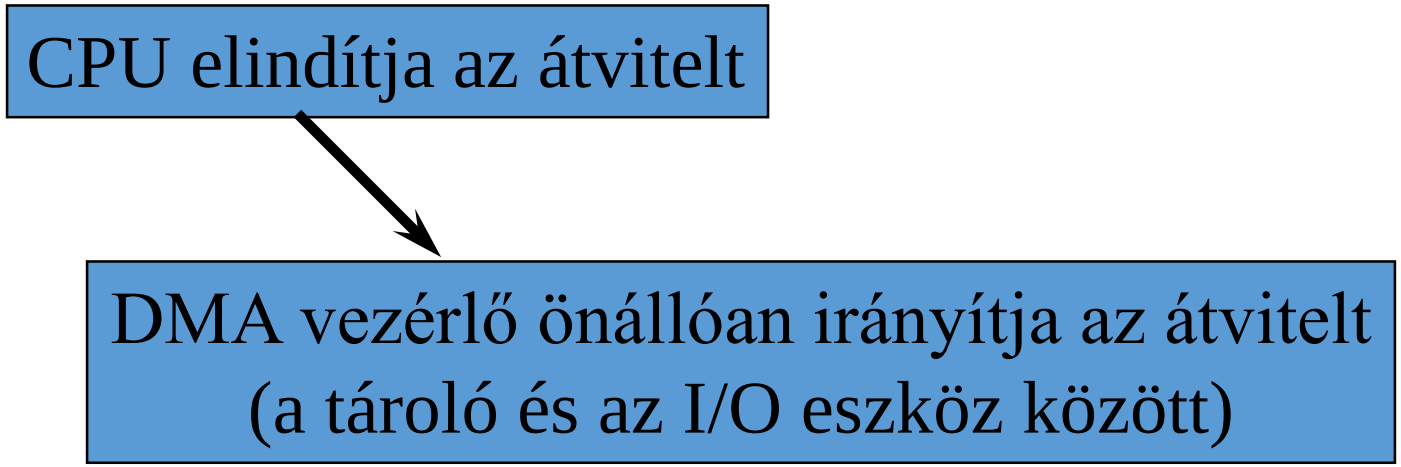
végrehajtás felfüggesztése a periféria működése közben



Megszakítás kezelés folyamata

- megszakítás engedélyezés
  - maszkolás
  - prioritás
  - megszakítható pont
- megszakítás analízis (kiszolgáló rutin ?)
  - kód - tárcím - utasítás
- állapotmentés
- kiszolgálás
- állapot visszaállítás

Közvetlen memória hozzáférés
(DMA - direct memory access)

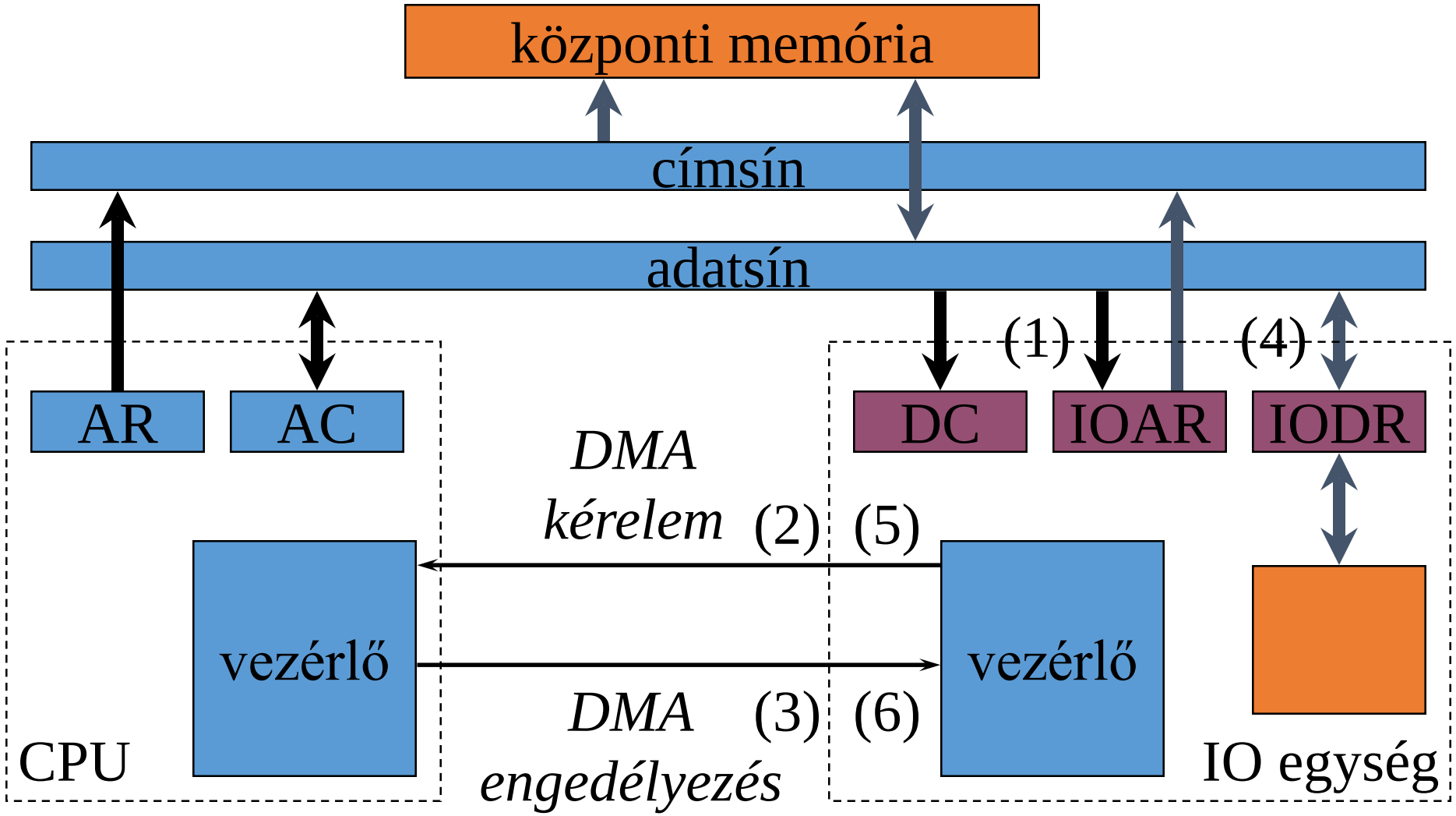


CPU - DMA közötti kapcsolat : megszakítások...

DMA : címsín, adatsín vezérlése...

nagysebességű eszközöknél...

Közvetlen memória hozzáférés

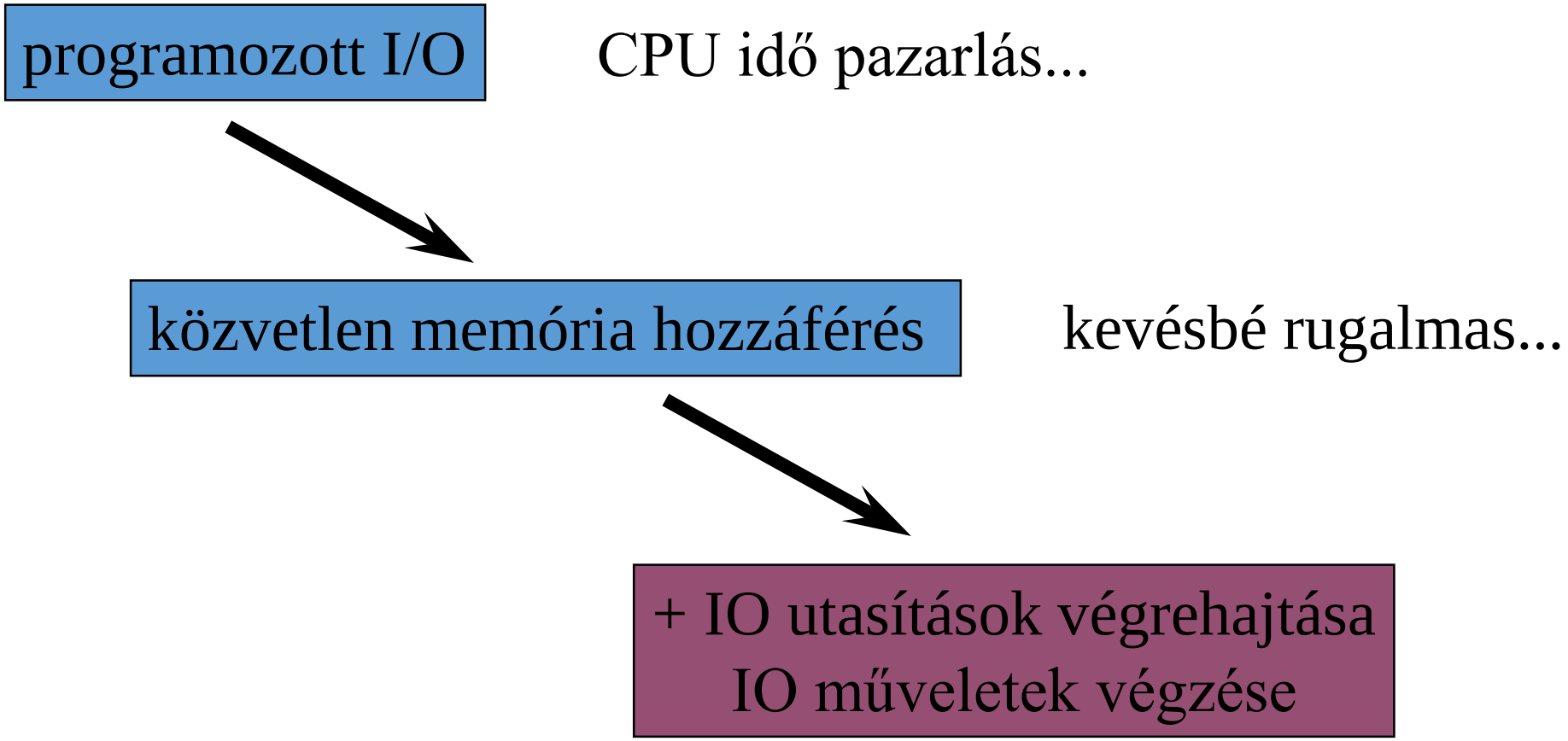


Közvetlen memória hozzáférés

Adatátviteli módok :

- Tömb átvitel (DMA block transfer)
- CiklusloPás (cycle stealing)

I/O processzor
(IOP)



I/O processzor (IOP)

IOP: korlátozott (speciális IO) utasításkészletű feldolgozó egység

PPU: Peripherial processing unit

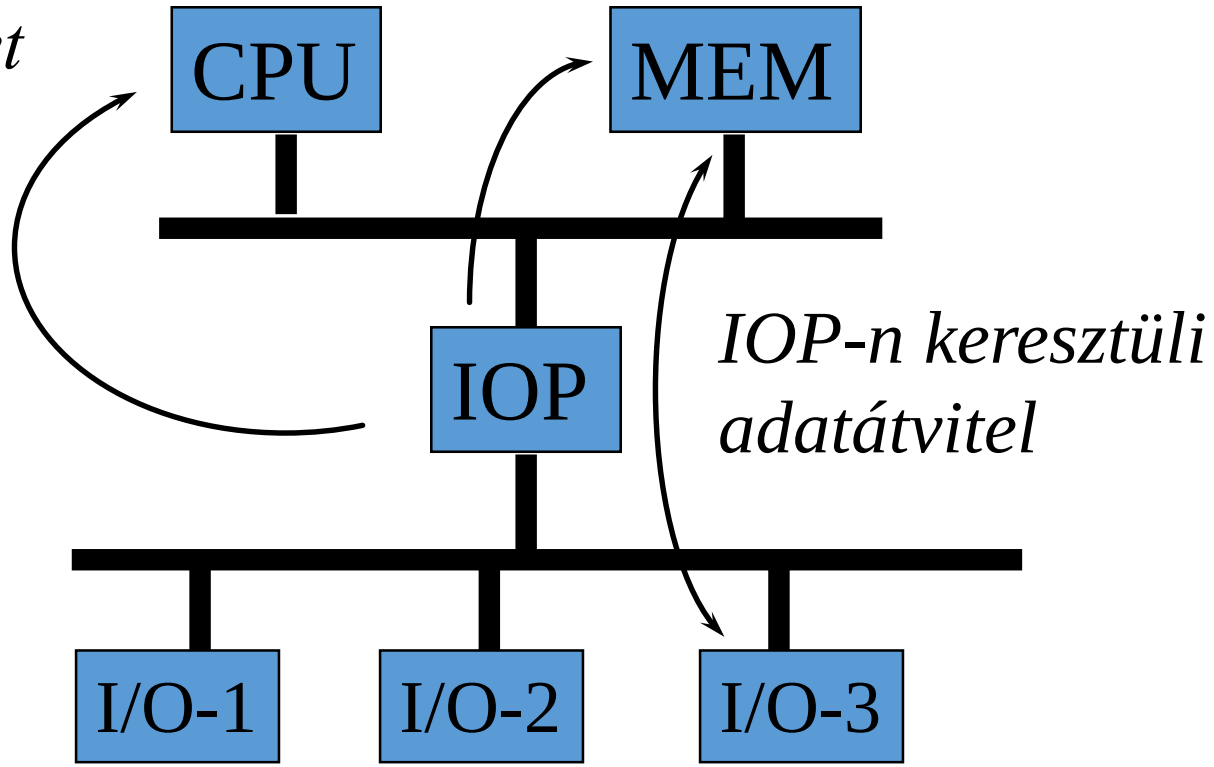
IOP: kommunikációs kapcsolat („csatorna - channel”)
a központi memória és az IO eszközök között

I/O processzor elvi működés

CPU által elindított IO művelet

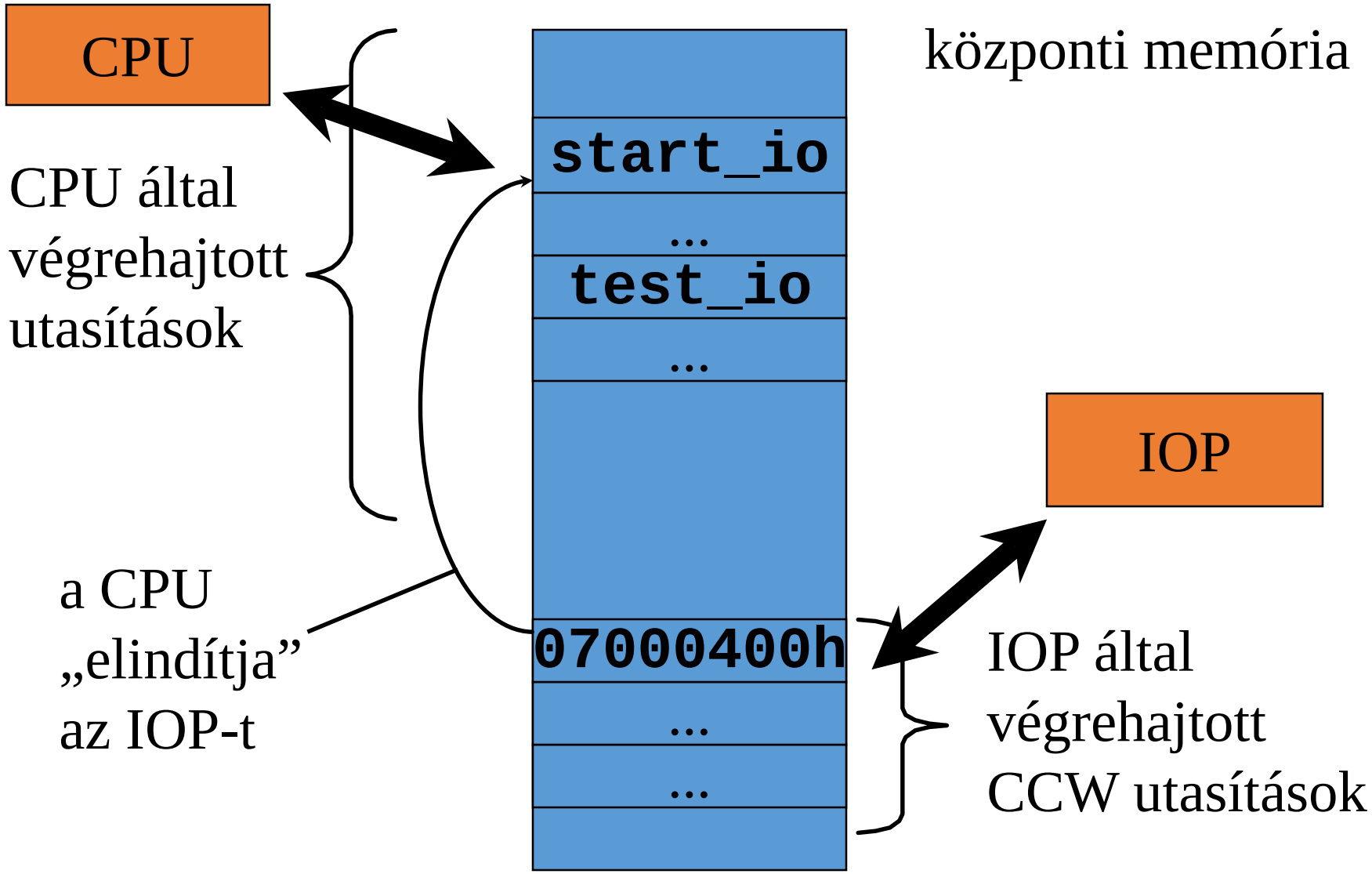
START_IO
STOP_IO
TEST_IO

IOP által végrehajtott utasítások (CCW)

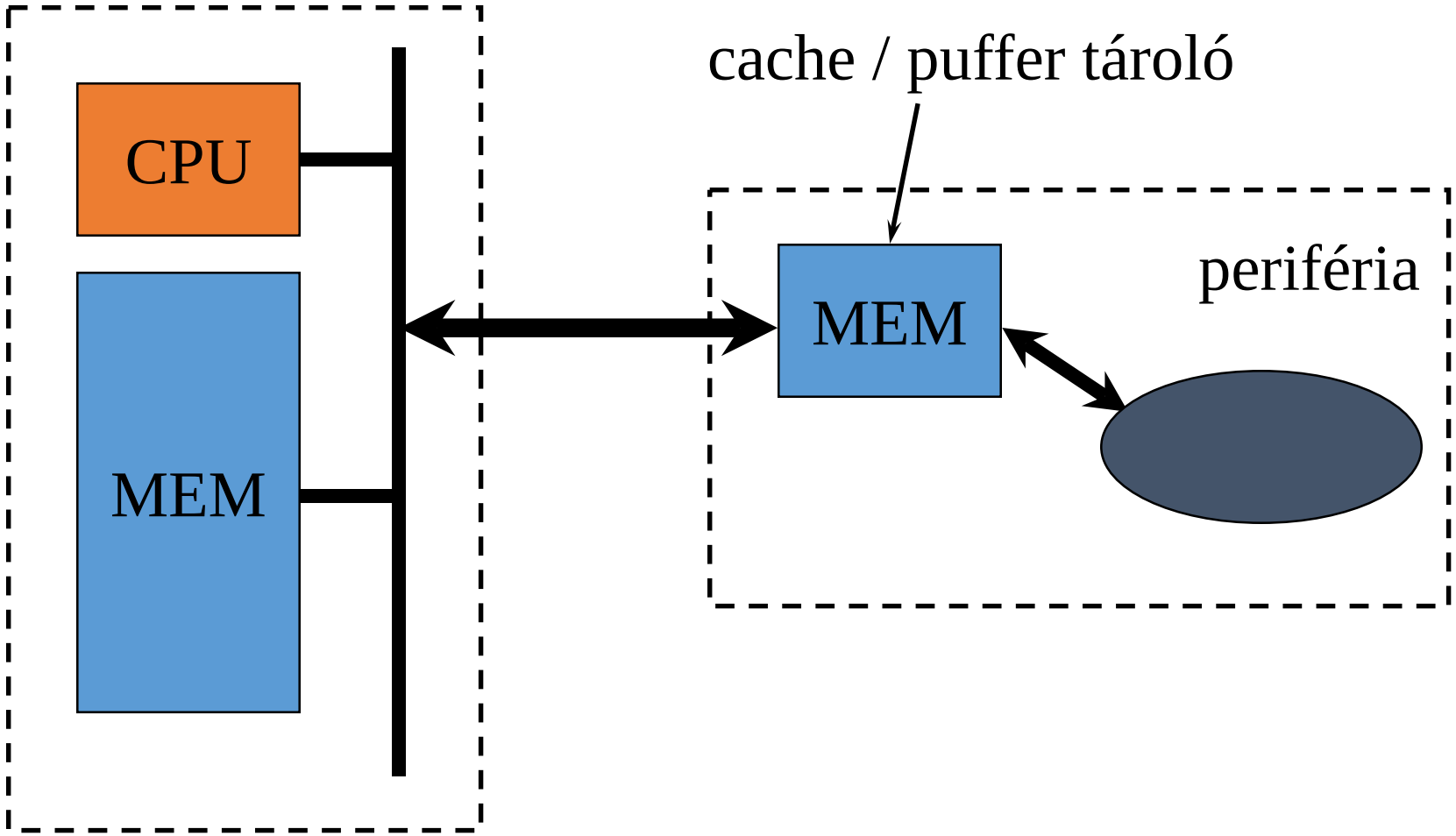


IOP-n keresztüli adatátvitel

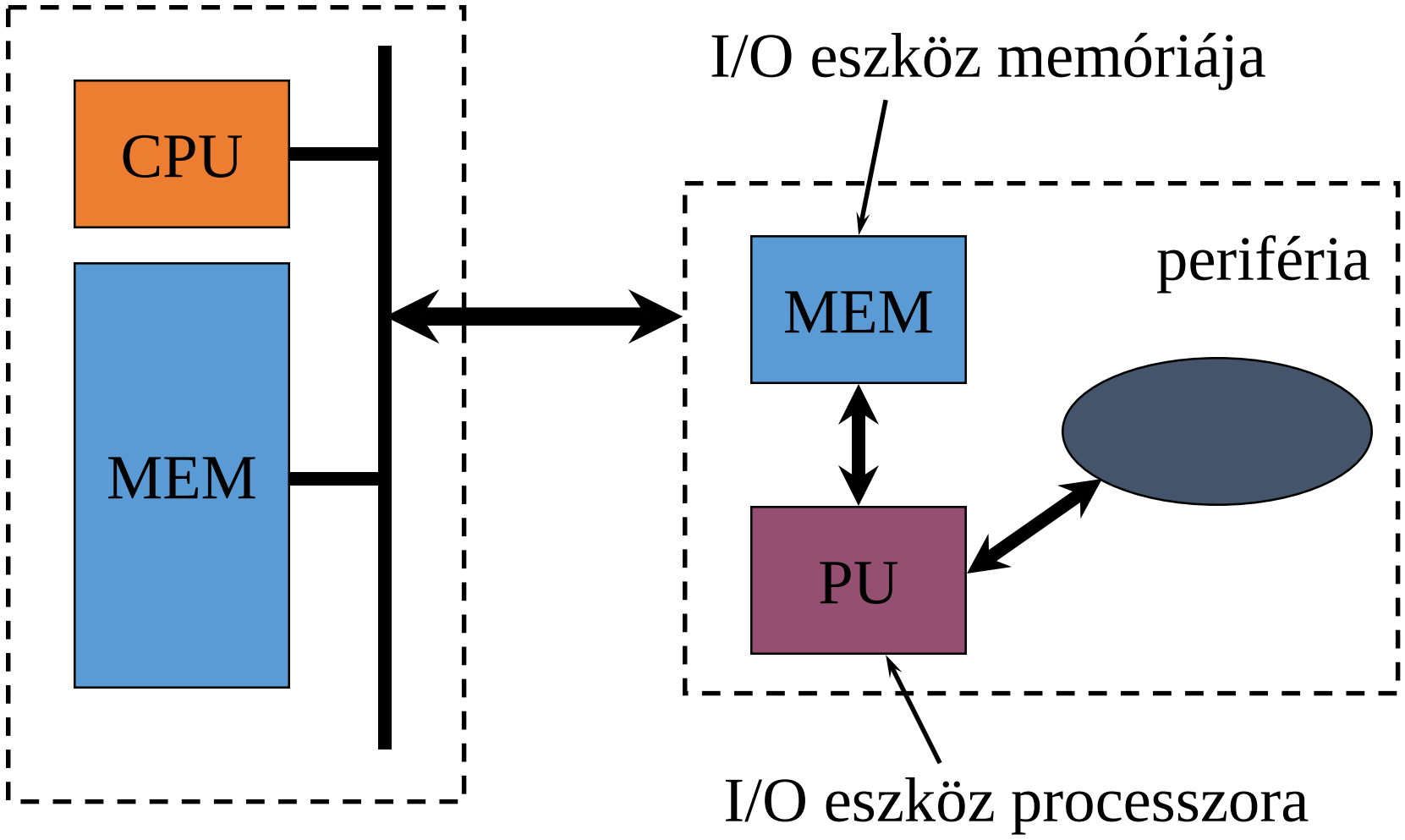
CPU és IOP utasítások a memóriában



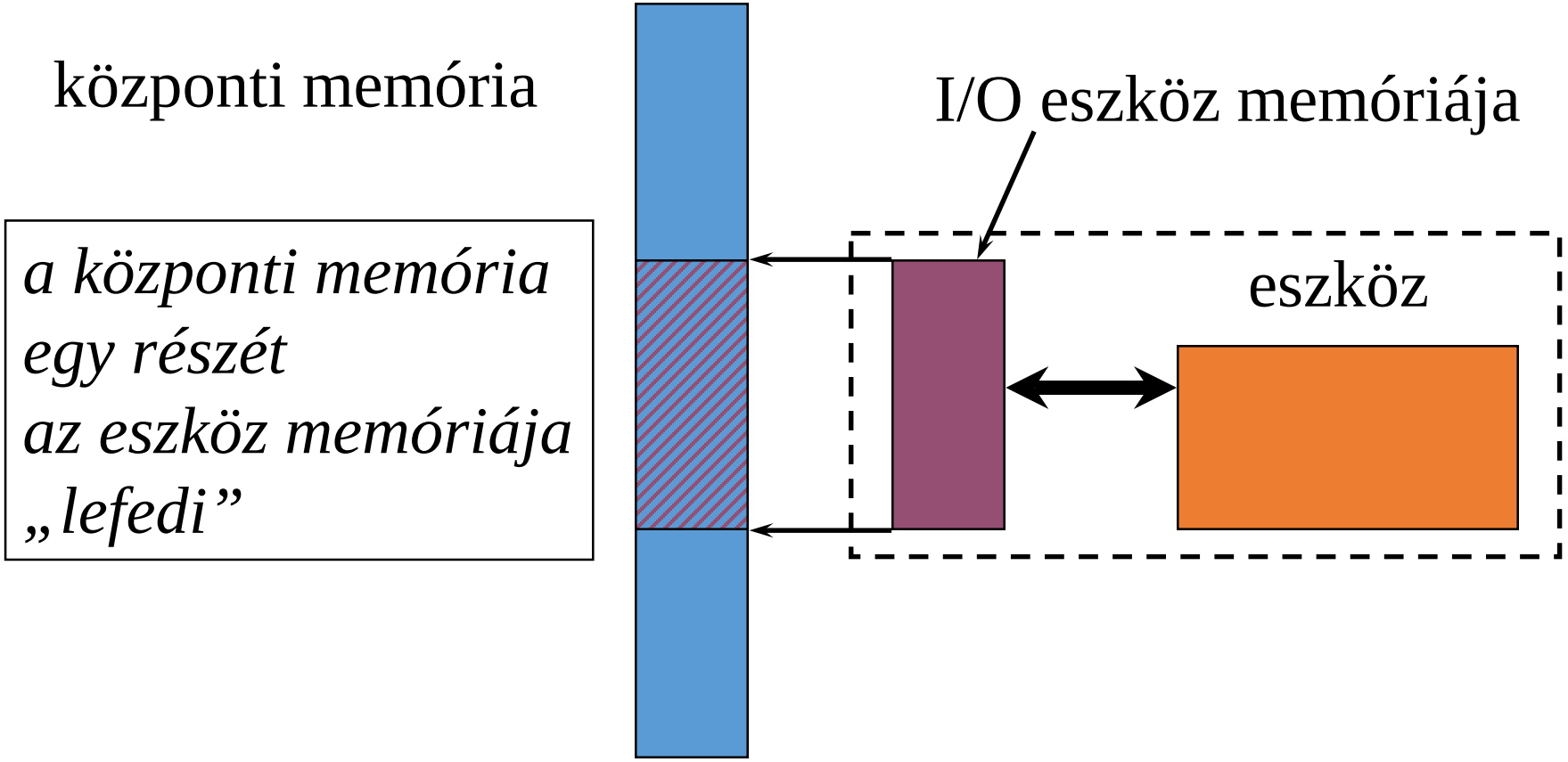
egyéb lehetőség(ek)…
(periféria cache/puffer tároló)



egyéb lehetőség(ek)…
(„periféria processzor”)



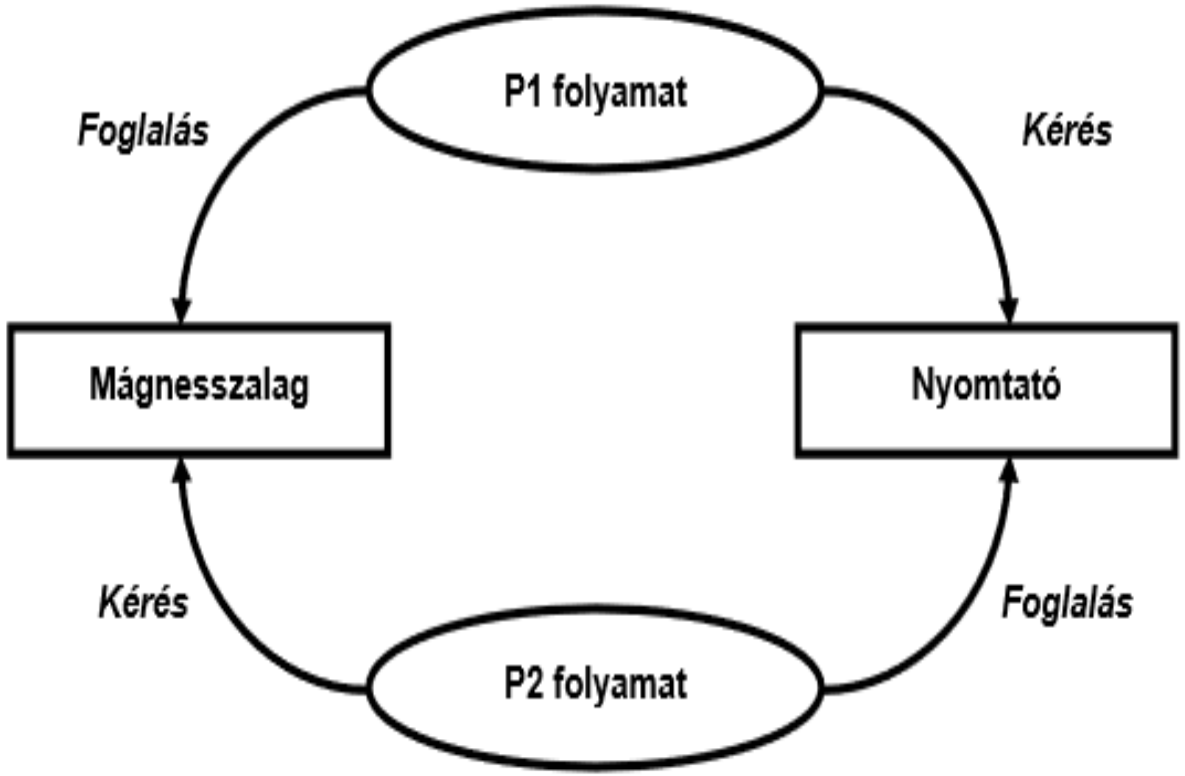
egyéb lehetőség(ek)…
(„memória lefedés”)



A holtpont fogalma - deadlock

- Azt mondjuk, hogy a folyamatok egy részhalmaza holtpontban van, ha a folyamatok halmazának minden egyes eleme valamelyik másik halmazbeli folyamat által kiváltható eseményre várakozik.
- Tehát hétköznapi nyelven fogalmazva: akkor fordulhat elő néhány folyamat között (azaz a folyamatok egy halmazában), ha ez a néhány folyamat mindegyike használatba vesz valamilyen mennyiségű erőforrást és közben ezen a folyamatok mindegyike várakozik is.
- Akkor is holtpont alakul ki, ha ezek a folyamatok nem kimondottan egy megadott várakoznak, hanem például egymás végeredményére. Ezt a kommunikációban létrejövő paradox állapotot is nevezzük.
- Szemléltető példaként szolgálhat a holtpont definíciójára az, hogy nemcsak a számítástechnikában és az ehhez kapcsolható tudományágakban „idézhetünk elő” holtpontot, hanem egyéb más területeken is megjelenhet ez a paradoxon, mint például a törvényhozásban.

Holtpont


P1 folyamat
Foglalás
Kérés
Mágnesszalag
Nyomtató
Kérés
Foglalás
P2 folyamat