

„Ha beérjük annyival, hogy elátkozzuk vagy dicsőítjük a technikát, akkor sohasem jutunk el lényegének a megragadásához.”

Martin Heidegger

MIKROVEZÉRLŐS RENDSZERFEJLESZTÉS

ChibiOS/RT

Beágyazott operációs rendszer

Zsupányi Krisztián

Digitális be és kimenetek kezelése



STARTING STARTING STARTING STARTING

BEÁGYAZOTT OPERÁCIÓS RENDSZEREK



EMBEDDED SYSTEM

Mitől működik egy autó vagy egy háztartási gép?



Egy **beágyazott rendszer** olyan speciális számítógép, amelyet egy adott feladat vagy feladatkör ellátására terveztek.

Nem általános célú, mint a PC, hanem egy nagyobb rendszer részeként működik, sokszor észrevétlenül a felhasználó számára.

Korlátozottabb erőforrások: általában kisebb processzor, kevés RAM/flash memória.

Mi működteti? Egy beágyazott operációs rendszer.

MI A KONCEPCIÓ?

Leggyakrabban komplexek ezek a rendszerek.

Felbontás kisebb egységekre

Esemény

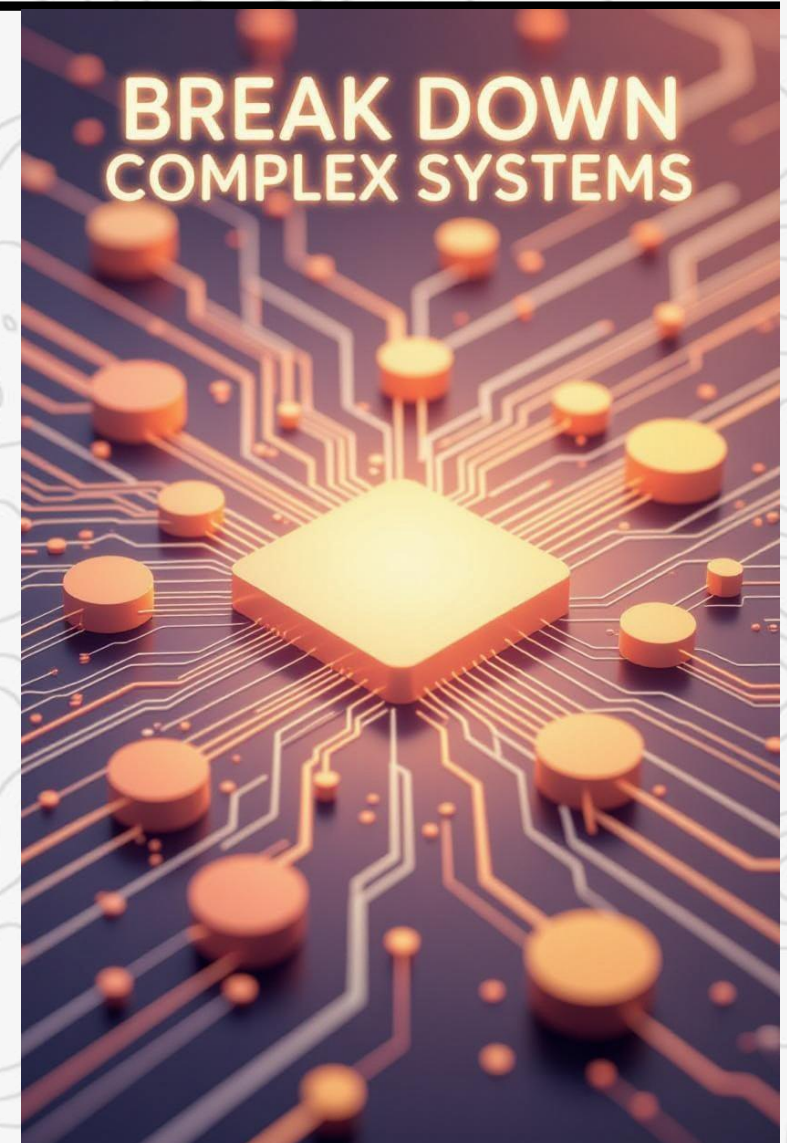


Reakció

Szoftver, hardver stb...

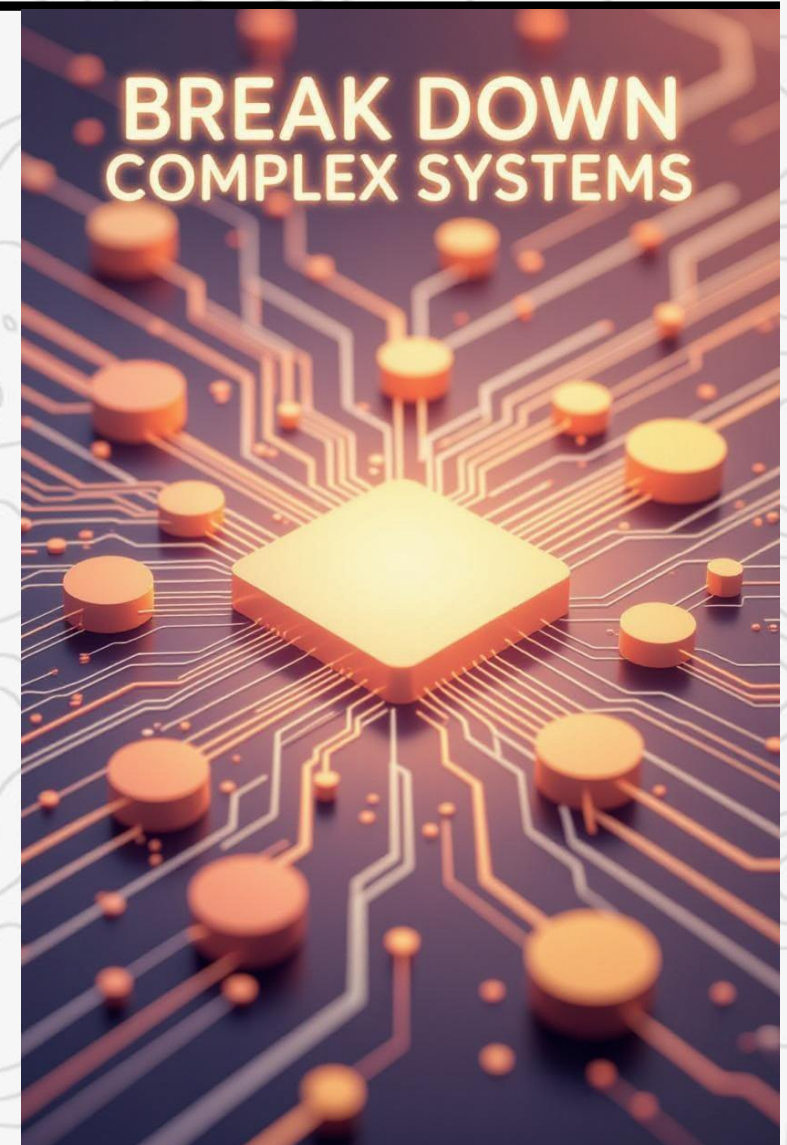
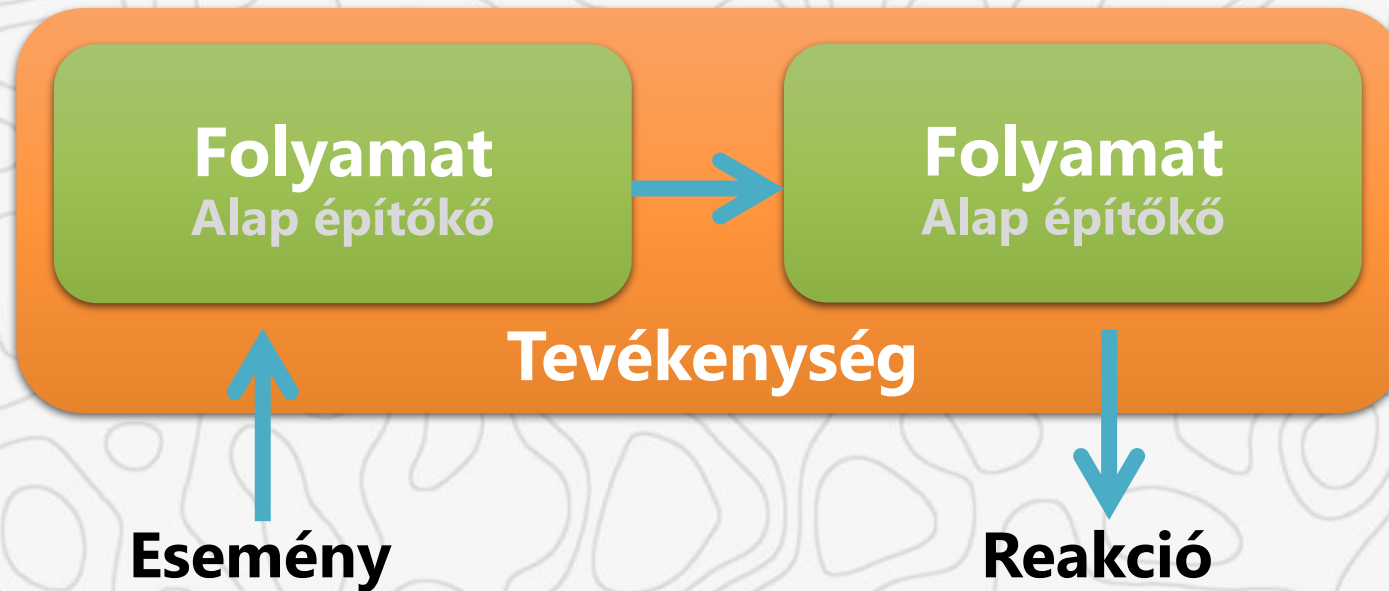
A feladat végre hajtása megadott időn belül teljesül

Beágyazott operációs rendszerek



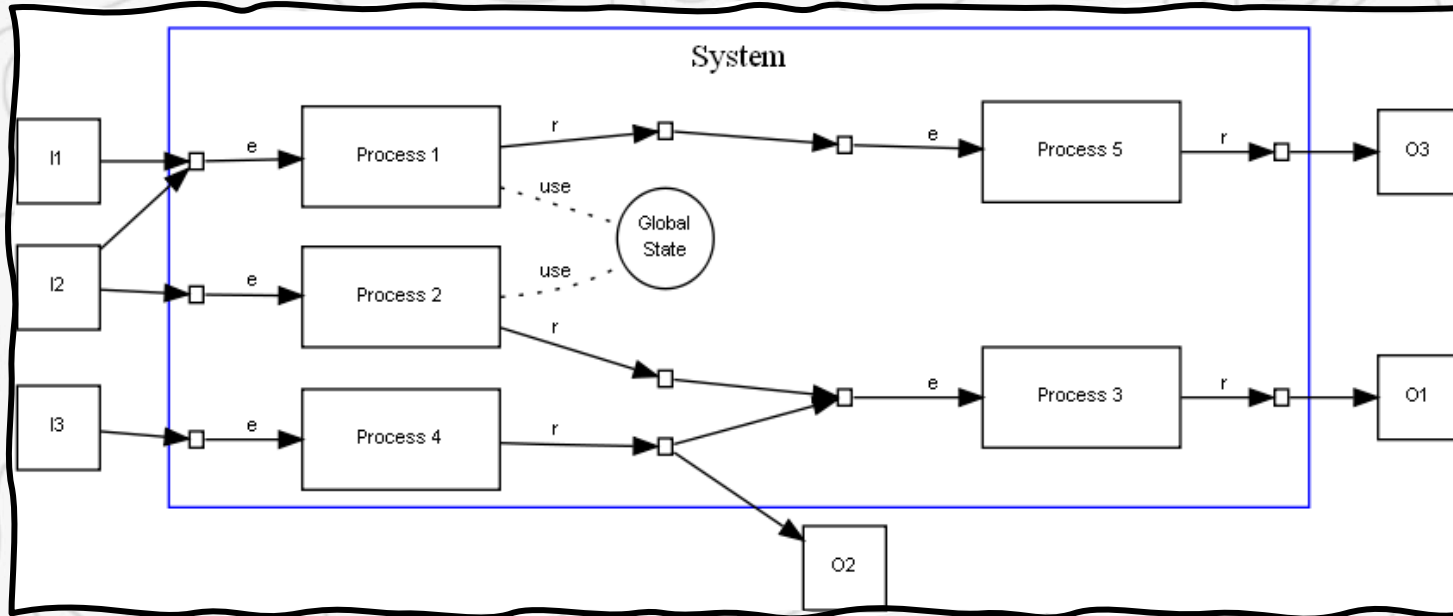
MI A KONCEPCIÓ?

Több folyamat láncba rendezve, tevékenységet hoz létre



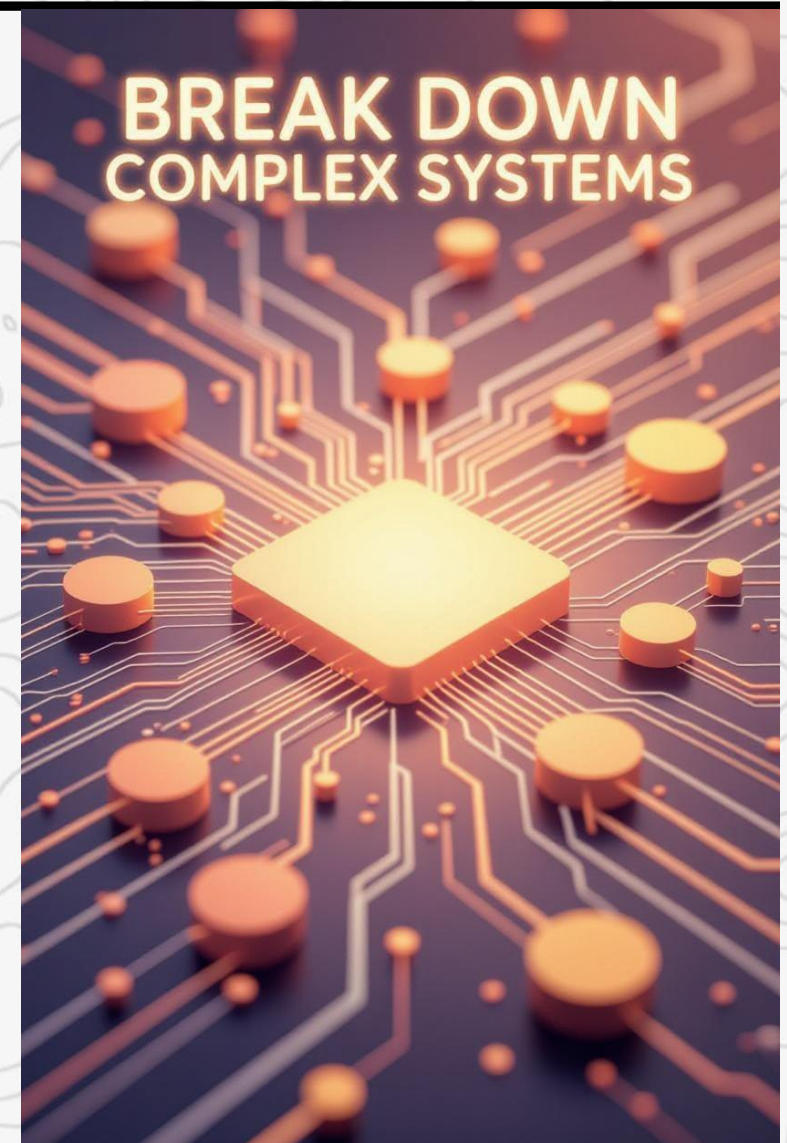
MI A KONCEPCIÓ?

A rendszer a folyamatok összessége.



Az folyamatokat időzíteni kell egymáshoz képest.

Beágyazott operációs rendszerek



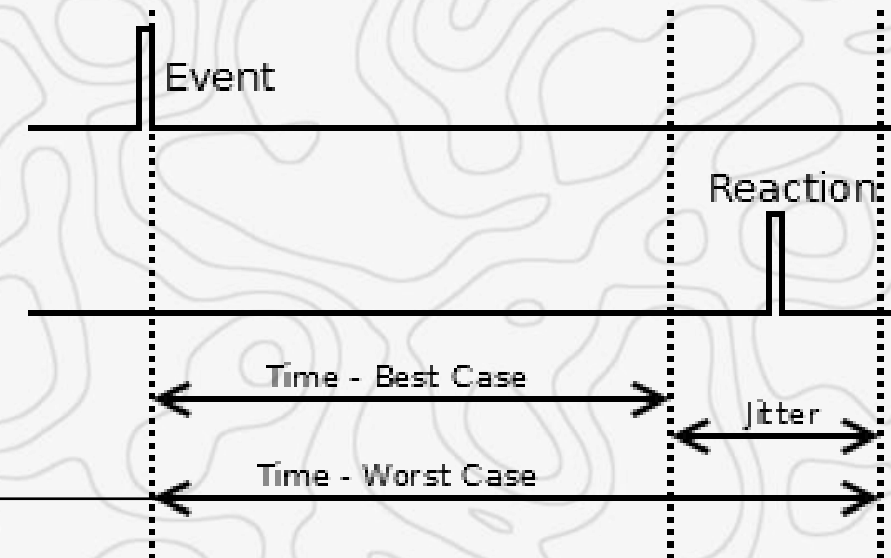
REAL TIME (VALÓS IDEJŰ) RENDSZEREK



- **Non Real Time (NRT):** A nem valós idejű rendszer olyan rendszer, **amelyben nincsenek időkorlátok**, az eseményre adott reakció valamikor a jövőben megtörténik. Pl: fájl másolás
- **Soft Real Time (SRT):** A Soft Real Time olyan rendszer ahol az **időkorlát elmulasztása** nem kívánatos ugyan, de **nem jár tragikus következménnyel**. A rendszer a lehető leghamarabb reagál egy ismert időkorláton belül. Pl: Streaming
- **Hard Real Time (HRT):** A Hard Real Time olyan rendszer ahol az **időkorlát elmulasztása katasztrofális következménnyel járhat**. A taskok végrehajtási ideje determinisztikus, **garantáltan** megtörténik a megadott időn belül. Pl: autó légzsák vezérlés

Kombinálhatók, de a leggyengébb elem határozza meg a rendszer viselkedését. Pl: ha egy HRT folyamatot blokkol egy NRT.

Jitter: A folyamatok sohasem állandó idő alatt reagálnak. A legjobb és legrosszabb eset között eltelt idő jelenti a Jitter-t.



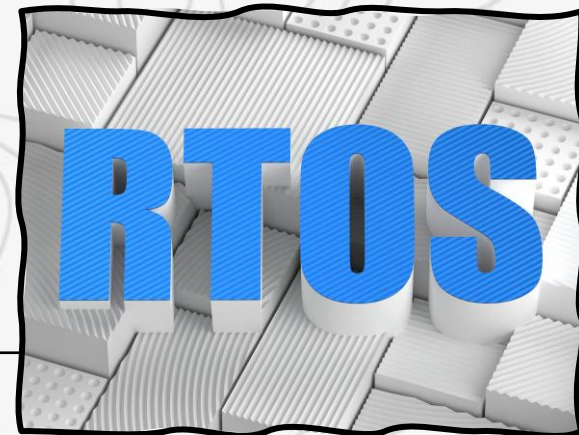
VALÓS IDEJŰ OPERÁCIÓS RENDSZEREK (RTOS)

Beágyazott operációs rendszerek

VALÓS IDEJŰ RENDSZEREK TULAJDONSÁGAI

- **Időkritikus alkalmazásokhoz**
- OS működése, ütemezője, rendszerhívásai megfelelnek a valós idejű követelményeknek.
- **Kiszámíthatóság:** ütemezési algoritmus, rendszerhívásainak végrehajtási ideje előre megjósolható. Ismerjük legrosszabb végrehajtási idejét.
- Nincsenek benne váratlan, hosszú ideig tartó műveletek.

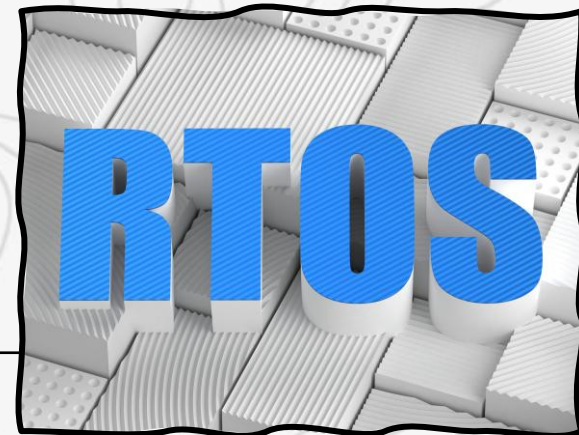
**Nem fejtétlen a sebesség a lényeg,
hanem a kiszámíthatóság!**



RTOS SZUPERKÉPESSÉGEI

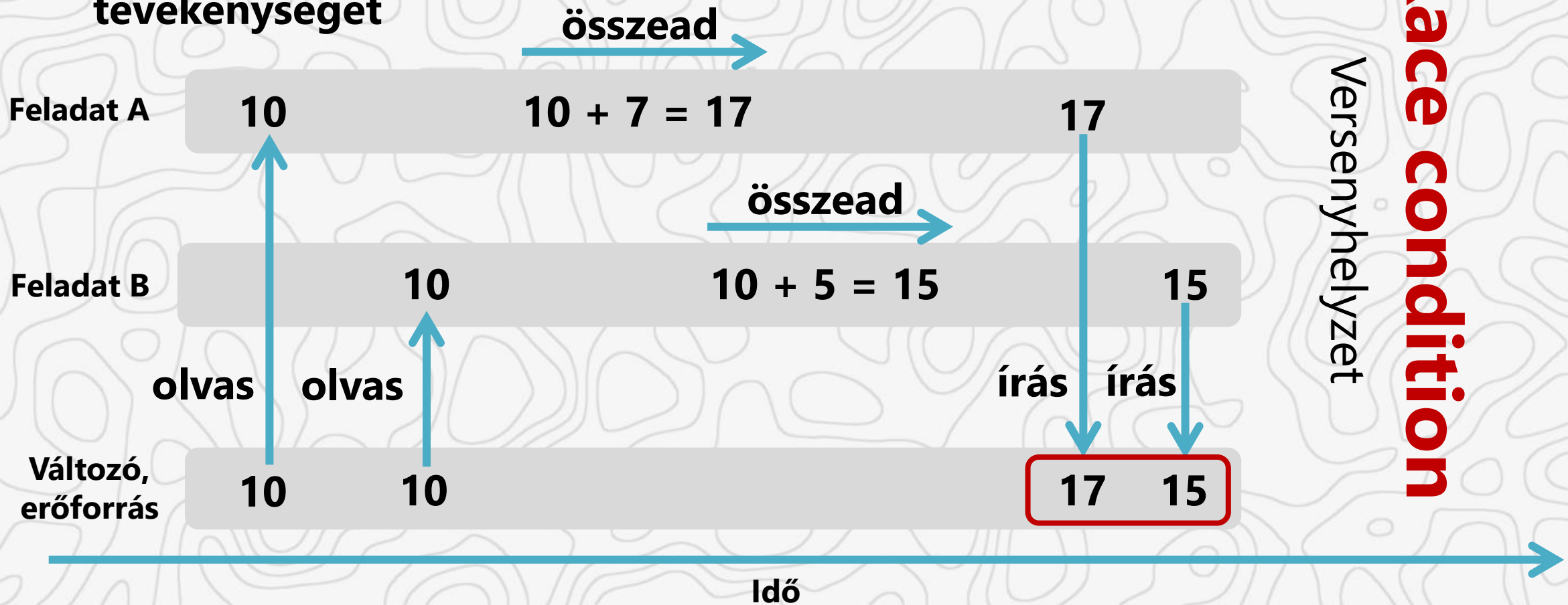
- **Többszálú, egyalkalmazásos modell**
- **Fix prioritású ütemezés,** legsürgősebb feladatot hajtja végre először
- **ISR kezelése,** azonnal tud reagálni a váratlan eseményekre

Bármely pillanatban a legmagasabb prioritású, futásra kész feladat kerül végrehajtásra!



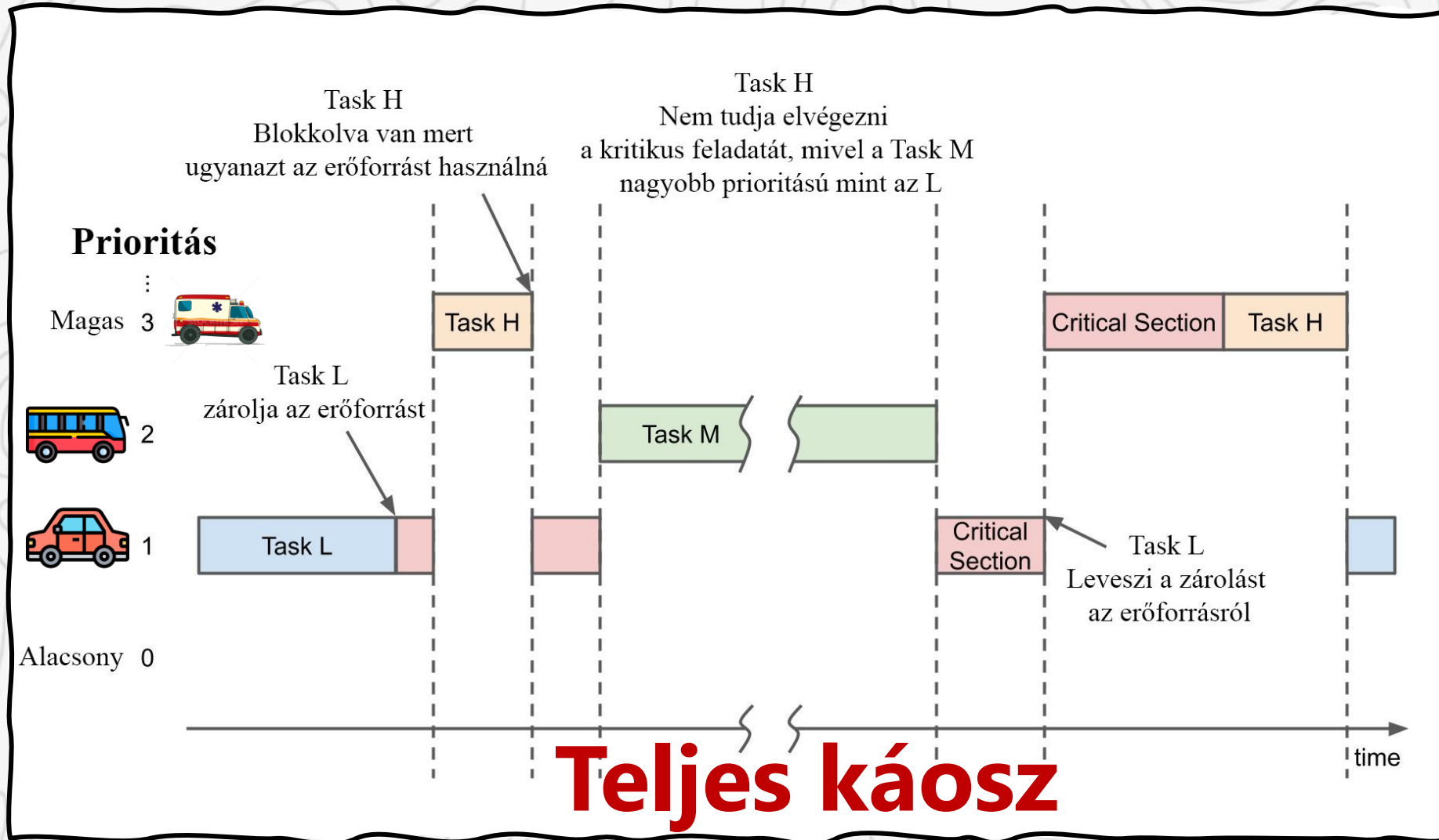
ÜTEMEZŐ SZINKRONIZÁCIÓ KIHÍVÁSA

- Az ütemező kisebb feladatokra bontja a teljes tevékenységet



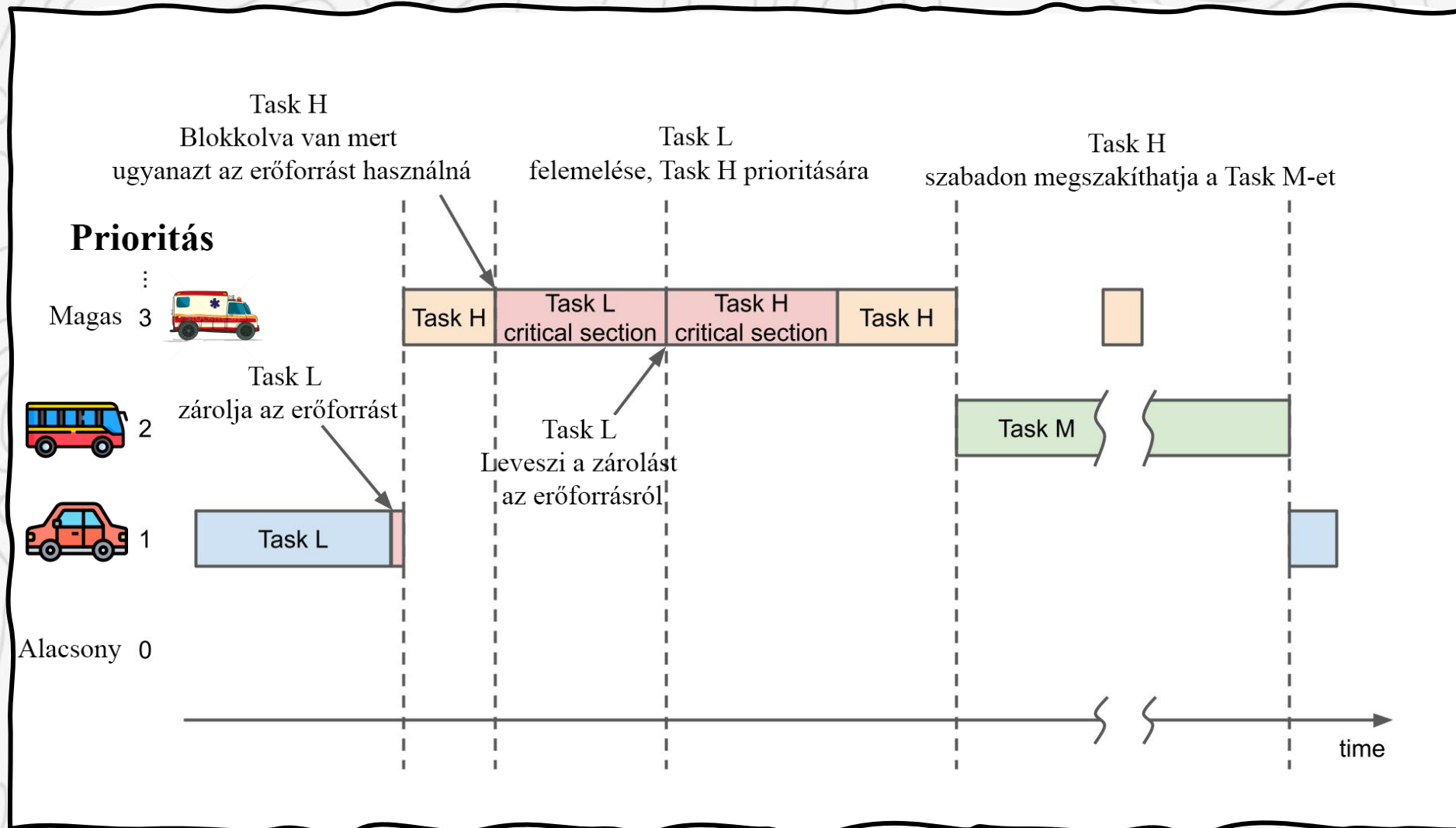
MEGOLDÁS?

Prioritás inverzió



MEGOLDÁS

Prioritás öröklés





ChibiOS

CHIBIOS/RT (ちびOS/RT) EJTÉSE „CSIBIOESZ”

- Fejlesztője **Giovanni Di Sirio**
- Ingyenes (GPL3 license).
- Valós idejű alkalmazásokhoz.
- Könnyen portolható.
- Preemptive ütemező.
- 128 prioritási szint. Azonos prioritái szinten megengedett több szál használata.
- Round robin ütemező azonos prioritással rendelkező szálakra.
- Alapelemek: threads, virtual timers, semaphores, mutexes, condvars, event flags, messages, mailboxes, I/O queues.
- Szinkron (blokkoló) és aszinkron (nem blokkoló) I/O kezelés.
- Kis rendszer követelmény: 6KiB ROM. Lecsökkenthető 2KiB-ra, ha a méretre optimalizálás be van kapcsolva és a nem használt elemek ki vannak kapcsolva.
- Szinte teljesen C-ben van megírva (assembly a portoláshoz kell).
- Lehetőség van használni a Hardware Abstraction Layer (HAL) modult, ami az eszközmeghajtókat tartalmazza.



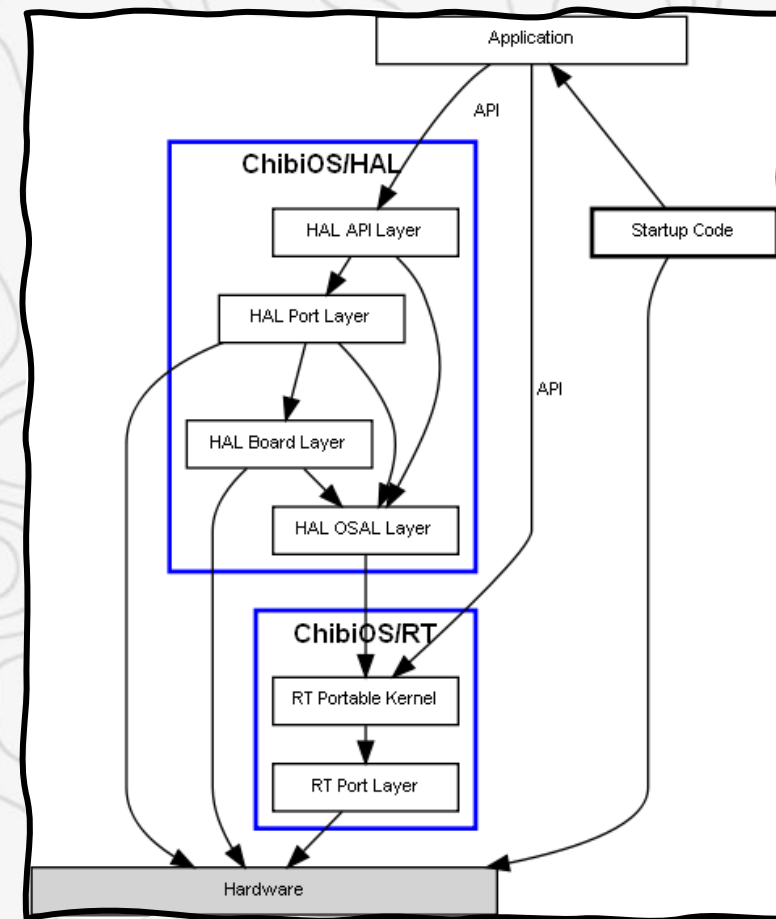
CHIBIOS/RT ARCHITEKTÚRÁJA

- **RT**, az RTOS ütemezője. Az RT egy nagyon nagy teljesítményű RTOS, teljes körű szolgáltatásokkal és kis alapterülettel.
- **NIL**, egy alternatív RTOS. A NIL kompatibilis az RT-vel, de a belső architektúrája teljesen más, minimális kódméretre tervezték.
- **OSLIB**, egy kiterjesztés az RTOS könyvtárhoz. Az RT vagy a NIL tetején található, magasabb szintű szolgáltatásokat kínál.
- **A HAL**, a Hardver Absztrakciós Réteg illesztőprogramokat tartalmaz a leggyakoribb perifériákhoz.
- **Az SB**, az RT vagy NIL kiterjesztése, amely elszigetelt „homokozót” kínál, ahol a nem biztonságos kód futtatható. A „homokozóban” lévő kód nem képes összeomlani az egész rendszert.

A ChibiOS/RT API-k:

ch<group><action><suffix>()

Groups: *Sys, Sch, Time, VT, Thd, Sem, Mtx, Cond, Evt, Msg, Reg, SequentialStream, IO, IQ, OQ, Dbg, Core, Heap, Pool.*



„Ha beérjük annyival, hogy elátkozzuk vagy dicsőítjük a technikát, akkor sohasem jutunk el lényegének a megragadásához.”

Martin Heidegger

KÖSZÖNÖM A FIGYELMET!

Zsupányi Krisztián

Beágyazott operációs rendszerek



ENDING ENDING ENDING ENDING E