Operációs Rendszerek MSc

Valós idejű OS-ek (RTOS)

2019/2020/I.

Dr. Vincze Dávid
Miskolci Egyetem, IIT
vincze.david@iit.uni-miskolc.hu

Operációs Rendszerek MSc

- Real-Time OS (RTOS)
  - Real-Time: valósidejű, azonnal válaszoló, stb.
  - Cél:
    - Kiszolgálni alkalmazásokat, amiknek valós idejű igényei vannak
    - Megadott időn belül **garantál**ja a szolgáltatásait
    - Az OS funkcióinak végezniük kell egy adott határidőig (és természetesen nem lehet hibás a működésük)
    - Amennyire lehet, legyen **determinisztikus**
    - A nem várt eseményekre is **kiszámíthatóan** reagáljon
  - Nem is a maximális teljesítmény a lényeg, hanem a garantált erőforrás biztosítás

Operációs Rendszerek MSc

- Real-Time OS (RTOS)
  - Többnyire beágyazott rendszerekhez, de általános célú gépeken („sima PC”) is használhatóak
    - Beágyazott: kevés erőforrás (mikrokontroller, low-power)
    - MMU nélküli HW, de ha van is virtuális memória, swap általában nincs
  - Moduláris felépítés
    - Ami nem kell nem tesszük bele
    - Licenszelés...
  - Az összes rendszerhívásnak determinisztikusnak kell lennie
    - Időkorláttal kell rendelkezniük
  - Megszakításkezelőknek szintén

Operációs Rendszerek MSc

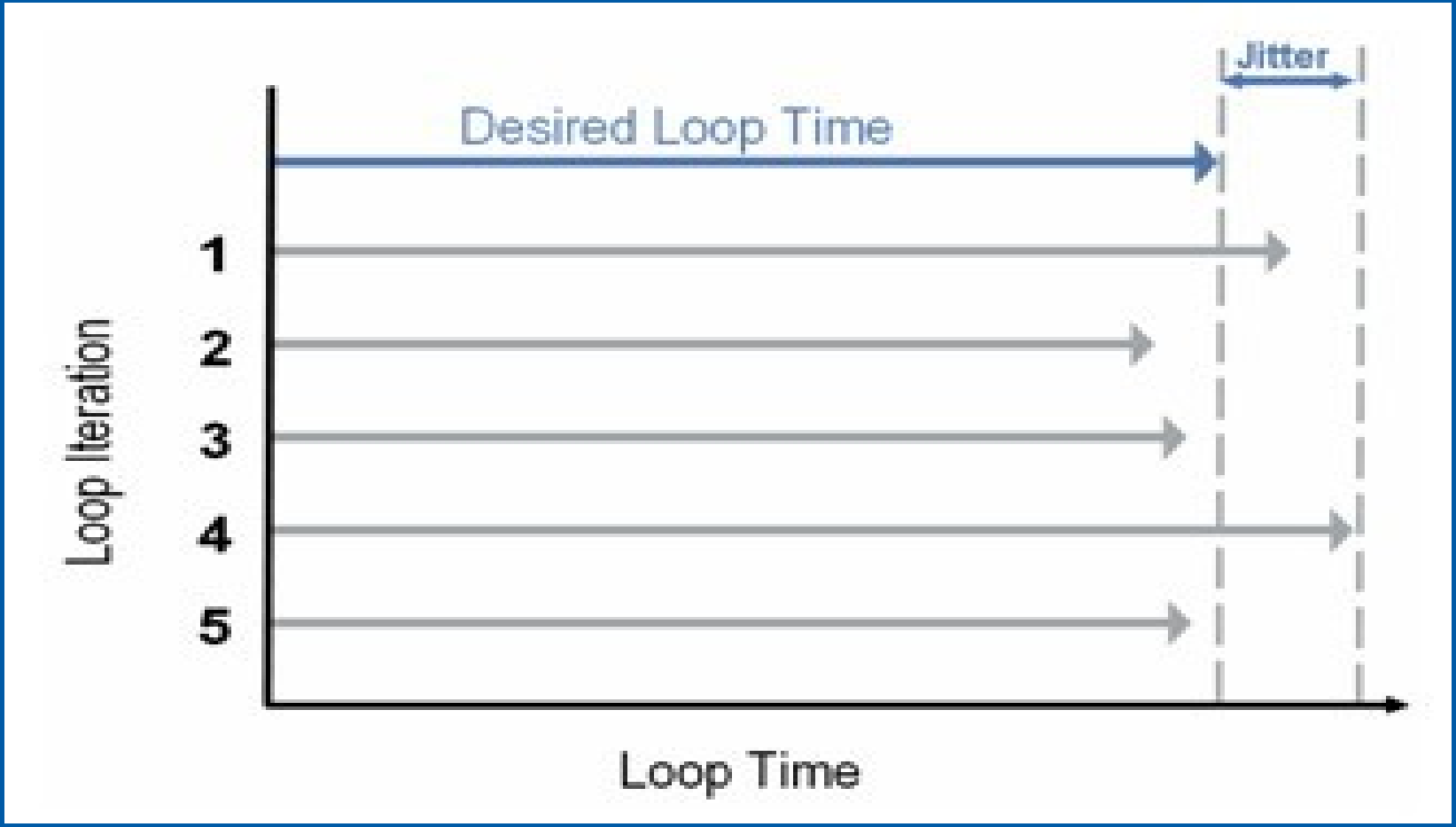
- Real-Time OS (RTOS)
  - Hard RTOS / Firm RTOS / Soft RTOS
  - **Hard** („merev”) real-time: *A határidő be nem tartása megengedhetetlen*
    (vagy csak nagyon minimálisan van tolerálva).
    Egy be nem tartott határidő katasztrofális következményekkel járhat a rendszerre nézve.
  - Biztonságilag kritikus rendszereknél.
    - pl. repülő, vonat, járművek, rakéta, atomerőmű, stb.

Operációs Rendszerek MSc

- Real-Time OS (RTOS)
  - **Firm** (erős, határozott) real-time: *egy határidő elmulasztása elfogadhatatlan minőségromlást okozhat.*
    - (!minőség, okozHAT!)
    - pl. ipari irányítás, robotika
  - **Soft** (puha) real-time: *A határidők elmulaszthatóak (de azért törekszik a betartására), és az elmulasztásból származó gondokból fel lehet „épülni”. Elfogadható a rendszer minőségének romlása.*
    - pl. set-top box, telefon, PDA, VR, képalkotás, stb.

Operációs Rendszerek MSc

- Real-Time OS (RTOS)
  - **Jitter**: az időbeli eltérés az iterációk futási ideje között



Operációs Rendszerek MSc

- Real-Time OS (RTOS)
  - Architektúra
    - Kernel / kernal (rendszermag)
      - eszközmeghajtók
    - API
    - Egyéb szervizmodulok
      - pl. debugger, hálózati stack
  - Kernelben
    - Interruptok kezelése
    - Taszk ütemezés (scheduler, dispatcher)
    - Memória menedzsment
    - Erőforrás megosztás
    - Inter Task Communication (lásd IPC)

Operációs Rendszerek MSc

- Egy jó RTOS:
  - Multi-task képes
  - Alacsony interrupt késleltetés:
    - HW késleltetés, amíg az IRQ eljut a CPU-hoz
    - Az aktuális instrukció végrehajtása
    - Előkészület a vezérlés interrupt handler-be juttatásához
  - Scheduler kiszámítható legyen (max. időn belül)
  - Gyors kontextus váltás
    - Felismerni, hogy van („lett”) egy futtatható taszk
    - Meghívni (CPU-ra „tenni”)
  - Memória management vezérlése (deter):
    - Biztosítani kell a taszkoknak, hogy fixen elhelyezzék a kódjukat a valós memóriában
  - Nagyfelbontású időzítők (minimum millisec, de microsec az elvárt)
  - Sokszínű IPC (SHM, szinkronizáció, msg queue, stb.)

Operációs Rendszerek MSc

- Egy jó RTOS:
  - Minden OS hívásra meg kell határozni a végrehajtási időt a legrosszabb esetre vonatkoztatva (worst-case)
  - Determinisztikus ütemező algoritmusra és erőforrás megosztásra van szükség
  - *Preemptív*: alacsonyabb prioritású taszktól lehetőség legyen elvenni az erőforrásokat, ha egy nagyobb prioritásúnak van szüksége rá

Operációs Rendszerek MSc

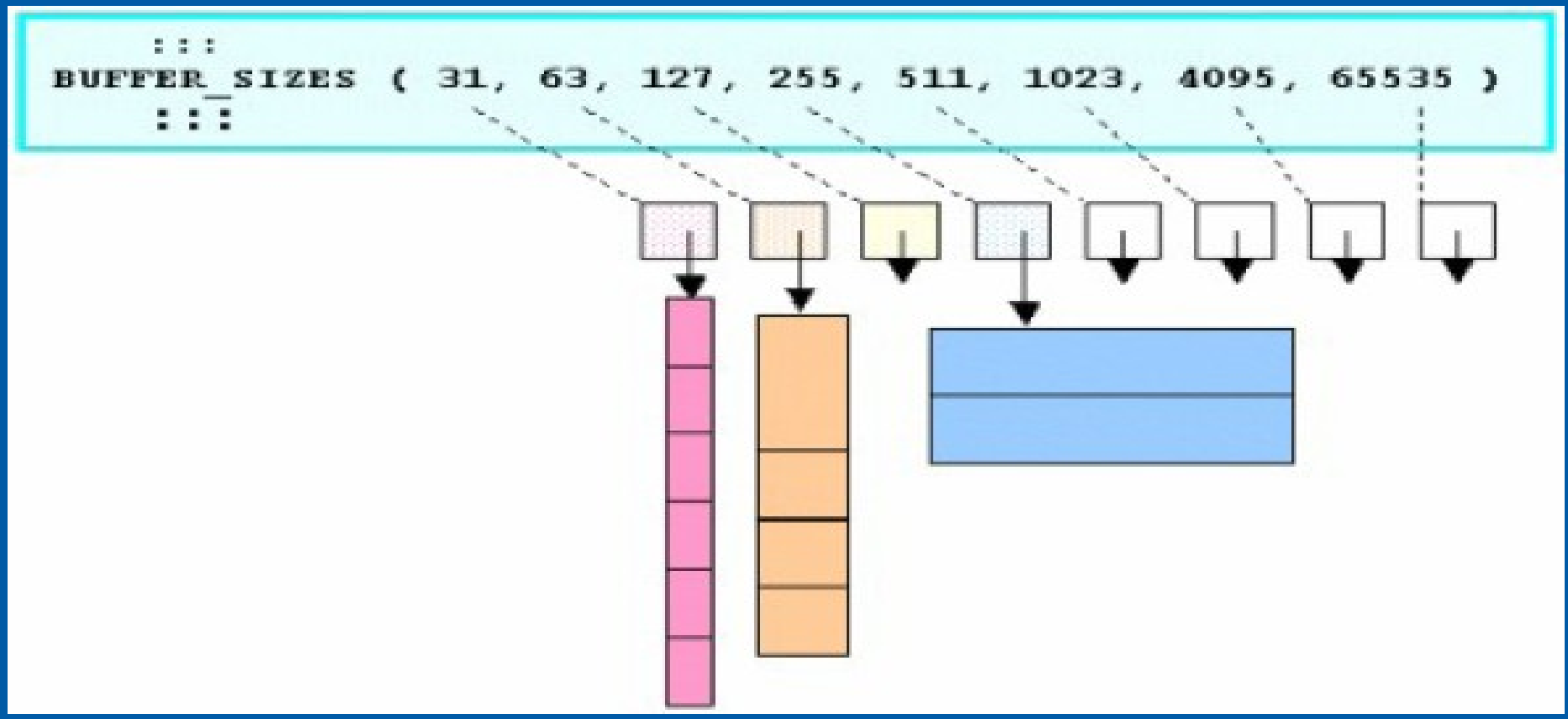
- **Kernel space vs. User space**
  - Sokszor az alkalmazás maga is kernel módban futhat:
    - nem kell syscall-oknál lépkedni ki-be a kernelbe, kernelből (hw oldalról is költséges)
    - jobb kontroll a környezett felett
    - biztonsági kockázat (jól megírt SW kell...)

Operációs Rendszerek MSc

- Real-Time OS (RTOS)
  - Memória menedzsment
    - Garantált idő alatt kell memóriát találni
    - Pl. láncolt listás allokálás nem elfogadható
    - Egyéb keresések (best-fit) sem
    - Foglalás ideje függ:
      - blokkmérettől
      - lapok állapotától (clean, dirty - fragmentáció)
      - swap...
    - Megoldás:
      - Nincs memória management
      - Minimális funkciójú MM
      - Lásd pl. fix partíciós memória kezelés (BSc OS anyagban)

Operációs Rendszerek MSc

- Real-Time OS (RTOS) – Memória MGMT
  - Csak meghatározott méretű tartományok foglalhatóak le
  - A blokkok egy „szabad blokk sorban” vannak nyilvántartva



Operációs Rendszerek MSc

- Real-Time OS (RTOS) – Memória MGMT
  - Előnye
    - Nincs fragmentáció
    - Egy blokk mindig le van foglalva egy egységként
    - Nem kell garbage collection, átrendezés
    - Implementáció egyszerű
    - Determinisztikus
  - Hátránya
    - Csak adott méret választható → pazarlás
    - Nem túl optimális az erőforrás (mem) kihasználtsága

Operációs Rendszerek MSc

- Real-Time OS (RTOS)
  - I/O Menedzsment
    - Nem lehet megjósolni az eszközök viselkedését
    - Kontrollálni sem mindig
    - Na jó. Azért a CPU és RAM jól kontrollálható.
    - Befolyásolhat pl.:
      - Gyorsítótár (cache)
      - Ethernet CSMA/CD (ütközés + random idő)
      - HDD hozzáférés (fej pozíció, pörgés sebessége, stb.)
      - Fájlrendszer fragmentáció
      - HW hiba
      - stb.
    - A drivereket általában újraírják, csak a szükséges funkciókkal (adhoc módon)
      - Sokszor gépi kódban, így ezért sem hordozható

Operációs Rendszerek MSc

- Real-Time OS (RTOS)
  - I/O Menedzsment
    - Az I/O műveletek általában nem megszakíthatóak/ folytathatóak
      - Non-reentrant
      - Ezért külön zárolás szükséges
        - pl.: printf()

```
void safe_printf(char *txt) {
   get_semaphore(s);
   printf(txt);
   release_semaphore(s);
}
```

Operációs Rendszerek MSc

- Real-Time OS (RTOS)
  - Aszinkron I/O
    - Nem kell megvárni a befejeztét
    - Elindítja a kérést
    - Fut tovább a taszk
    - Ha befejeződött, akkor a taszk kap egy signal-t

Operációs Rendszerek MSc

- Real-Time OS (RTOS)
  - Timer, idő kezelése
    - Alapvető fontosságú
    - Szigorú időzítési követelmények
    - Az ütemező is ezen alapszik
    - A max. időket/határidőket is ez alapján kell betartani
    - Várakoztatás (delay)
    - Várakozás (timeout)
    - Időmérés, időzítők felbontása
      - Elsősorban HW függő...
      - Ezen belül jó dolog, ha állíthatja a felhasználó
      - De! Nagyobb felbontás → nagyobb overhead
      - És fordítva.

Operációs Rendszerek MSc

- Real-Time OS (RTOS)
  - Mi a taszk?
    - Egy kis darabka futtatni való kód
      - Procedúra, függvény
      - Komplett program
      - Kernel modul
    - Egy program több taszkból is állhat
    - A taszkok konkurensen futnak (versengenek az erőforrásokért)
      - UniProcessing (UP) rendszeren ál-konkurens végrehajtás
      - Symmetrical MultiProcessing (SMP) rendszeren valós konkurens
    - A konkurens hozzáférés biztosítása az ütemező dolga
      - Eldönti, hogy melyik taszk kaphatja meg a CPU-t használatra

Operációs Rendszerek MSc

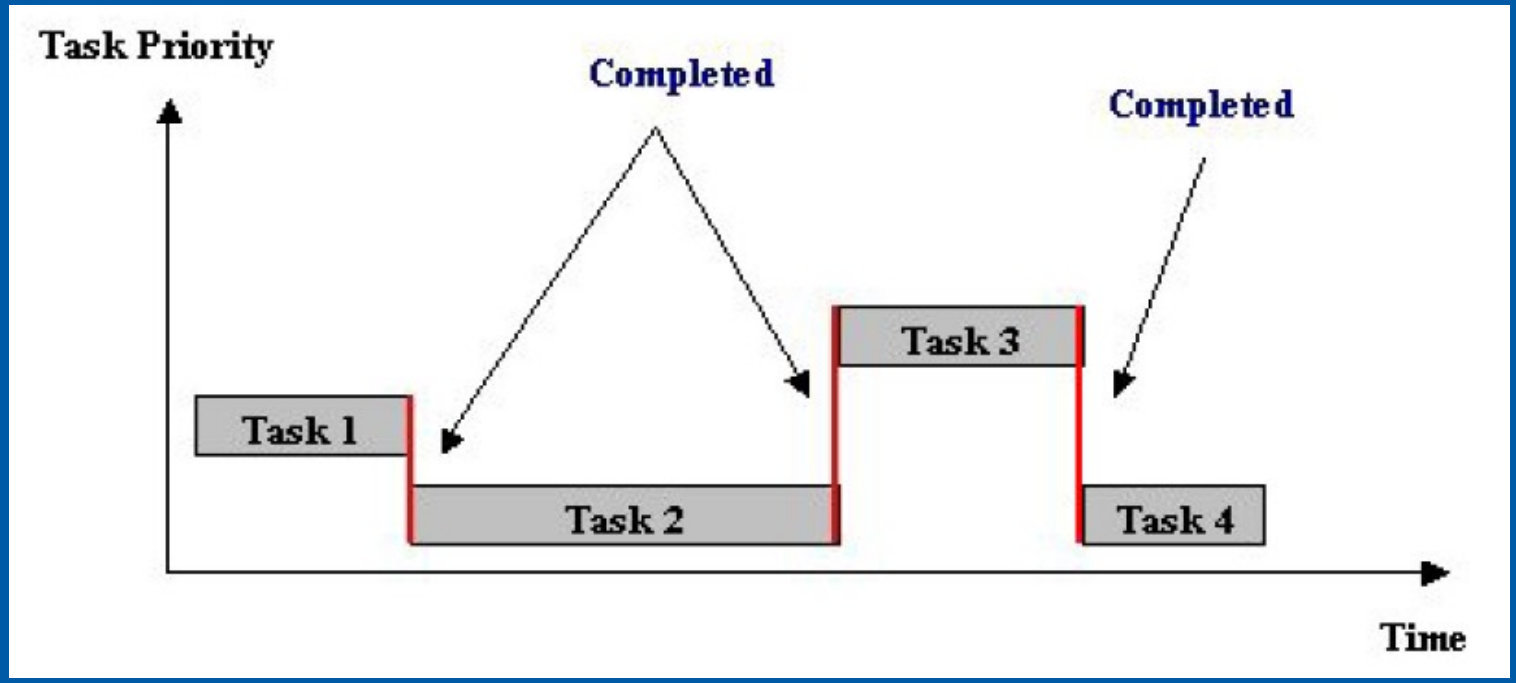
- Real-Time OS (RTOS)
  - Taszkok nyilvántartása
    - Task Control Block (TCB)
    - Azonosító (Task ID)
    - Prioritás
    - A legutóbbi futáskori kontextus másolata
    - Állapot *(Executing, Ready, Blocked, Waiting)*
    - Egyéb stb.
      Ismerős?
  - Új taszk létrehozásakor, egy TCB-t kell létrehozni
  - Taszk törléskör, pedig az adott TCB-t el kell távolítani (egyéb hozzá tartozó erőforrásokat felszabadítani)

Operációs Rendszerek MSc

- Real-Time OS (RTOS)
  - Scheduler (ütemező)
    - Melyik legyen a futó taszk?
      - Különböző algoritmusok
      - Prioritások kezelése
      - Cél a maximális CPU kihasználás
      - Cél a várakozási idő minimalizálása
    - Mikor lehet taszk váltás?
      - Külső interrupt, pl. időzítő megszakításai
      - Eseményvezérelt (nem időzítő, hanem egyéb megszakítás, vagy vége valamelyik taszknak)
    - Kontextus váltás költséges
      - A scheduler-be lépés is kontextus váltás !
      - → rendszerkontextus

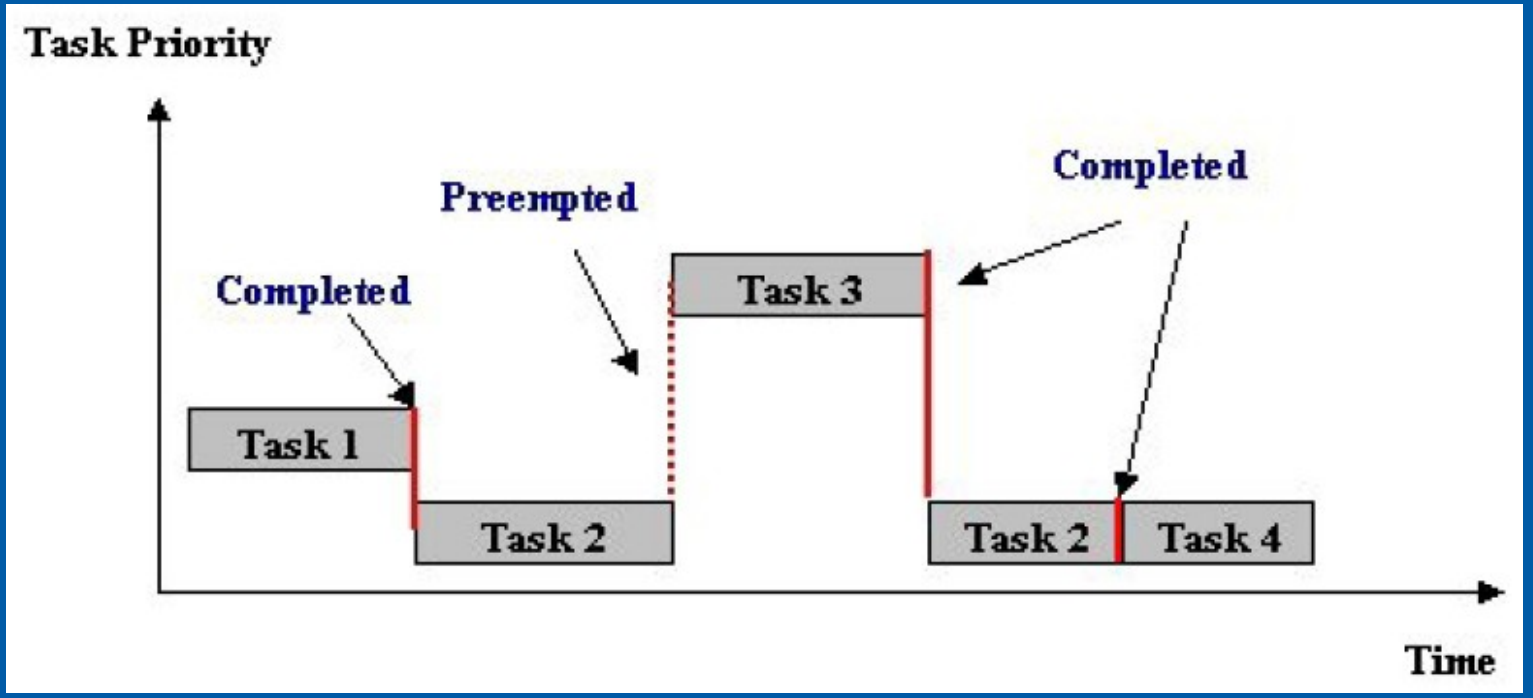
Operációs Rendszerek MSc

- Real-Time OS (RTOS)
  - Scheduler
    - Nem-preemptív
      - (kooperatív multitaszk)
      - Maguknak a taszkoknak kell lemondani a CPU-ról
      - Még akkor is ha van nagyobb prioritású taszk



Operációs Rendszerek MSc

- Real-Time OS (RTOS)
  - Scheduler
    - Preemptív
      - A taszkoktól el lehet venni a CPU-t
      - Nagyobb prioritású taszk megkaphatja azonnal a CPU-t



Operációs Rendszerek MSc

- Real-Time OS (RTOS) - Ütemező
  - Priorty Inheritance (prioritás öröklődés)
    - Példa

    - Egy kisebb prioritású taszk blokkol egy másik nagyobb prioritású taszkot valami erőforráson
    - Akkor arra az időre kaphat nagyobb prioritást
    - Azét a taszkét „örökli meg”, amelyik ugyanarra az erőforrásra vár, és nagyobb a prioritása

    - Ha egyszerűen csak odaadná a nagyobb prioritású taszknak a vezérlést, az blokkolódna ugyan úgy
    - Ezért „segít” a felszabadításában

Operációs Rendszerek MSc

- Real-Time OS (RTOS) - Ütemező
  - Megjegyzés IRQ
    - Az interruptok blokkolják a legmagasabb prioritású taszkokat is!
    - Minimalizálni kell a megjósolhatatlanságot

Operációs Rendszerek MSc

- Real-Time OS (RTOS)
  - Kommunikáció és szinkronizáció
    - Taszkok között
    - Kommunikáció
      - Nyilvánvaló, hogy szükség van rá
      - Producer/Consumer felállás pl.
      - Kölcsönös kizárás
    - Szinkronizálás
      - Találkozási pontok
      - Megvárni a másikat az eredménnyel
      - Megvárni egy másik taszk befejeztét
    - Emlékezzünk a korábbi tanulmányainkra:
      - Mailbox, message queue, osztott memória, mutex, szemafor, signal...

Operációs Rendszerek MSc

- Real-Time OS (RTOS)
  - RTLinux
    - Real-Time Linux kiegészítés
  - Szoftverből adódó kiszámíthatatlanságok
    - ütemező algoritmus
      - maximális teljesítményre van optimizálva
    - Eszközmeghajtók
    - Nem megszakítható rendszerhívások
    - IRQ kezelés felfüggesztése (letiltása/engedélyezése)
    - Virtuális memória kezelés
    - Priorty inheritance-t nem támogat
  - Megoldás:
    - párhuzamosan egy másik kernel is fut, ami csak épp annyi funkciót tartalmaz, amire szükség van, de minden hívása "megjósolható".

Operációs Rendszerek MSc

- Real-Time OS (RTOS) - RTLinux