

Többszálú programozás

LEGO Kör robotika tanfolyam

- C-hez hasonló szintaxis
- Csak inline
- Van visszatérési értéke (fontos)
- Akármennyi lehet belőlük
- Szintaxis:
 - inline <visszítésési típus><név>(<param>)
- Ritkán használatos

- C-s függvényhez hasonló működés
- Nincs visszatérési értéke
- Maximum 255 db
- Nincs névtúlterhelés
- Szintaxis:
 sub <név> (<paraméterek>)

- Legfontosabb műveletvégzési egység
- Kell legyen task main()
- Maximum 255 db
- Nincs paraméter
- Párhuzamos futás
- Szintaxis:
task <név> ()

- Keletkezés:
 - Csak mikor a hívó task lefutott
 - Start: csak egy indítása
 - Precedes: egyszerre több indítása párhuzamosan
- Lezáródás:
 - Függvénytörzs végén
 - Stop: kívülről megállítva
 - ExitTo: aktuális task befejezése, és átlépés ÚJ taskra

Konkurens folyamatok vezérlése

- Mi a probléma?
 - Párhuzamos művelet végrehajtás
 - Közös erőforrás egyidejű használata
 - Nincs felügyelő Operációs rendszer

Mi a hiba?

LEGO

KÖR

```
task main()
{
  Precedes(move_square, check_sensors);
  SetSensorTouch(IN_1);
}
```

```
task move_square()
{
  while (true)
  {
    OnFwd(OUT_AC, 75);
    Wait(1000);
    OnRev(OUT_C, 75);
    Wait(500);
  }
}
```

```
task check_sensors()
{
  while (true)
  {
    if (SENSOR_1 == 1)
    {
      OnRev(OUT_AC, 75);
      Wait(500);
      OnFwd(OUT_A, 75);
      Wait(500);
    }
  }
}
```

Konkurens folyamatok vezérlése

- Megoldás: Szemafor
 - Mi is ez?
 - Típusai
 - Bináris (leggyakoribb)
 - Inkrementális
 - Többállapotú
 - Rajta végezhető műveletek
 - Lefoglalás
 - Felszabadítás
 - Elemi művelet

Konkurens folyamatok vezérlése

- Szemafor implementálása NXC-ben
 - Bool típusú változó:
 - Két állapota a true és false
 - Foglálás globális változónak való értékadás (NEM ELEMÍ MŰVELET!)
 - Várakozás until(s); vagy while (!s);
 - Mutex (MUTual EXtension):
 - Speciális NXT típus (mutes)
 - Foglálás Acquire(s); függvényel, amely a változó felszabadulásáig várakozik is. ELEMÍ MŰVELET!
 - Felszabadítás Release(); függvényel

Bool használatával

LEGO

KÖR

```
bool semaphor = false;
task move_square()
{
    while (true)
    {
        until(semaphor);
        semaphor = true;
        OnFwd(OUT_AC, 75);
        Wait(1000);
        OnRev(OUT_C, 75);
        Wait(500);
        semaphor = false;
    }
}
```

```
task check_sensors()
{
    while (true)
    {
        if (SENSOR_1 == 1)
        {
            until(semaphor);
            semaphor = true;
            OnRev(OUT_AC, 75);
            Wait(500);
            OnFwd(OUT_A, 75);
            Wait(500);
            semaphor = false;
        }
    }
}
```

Mutex használatával

LEGO

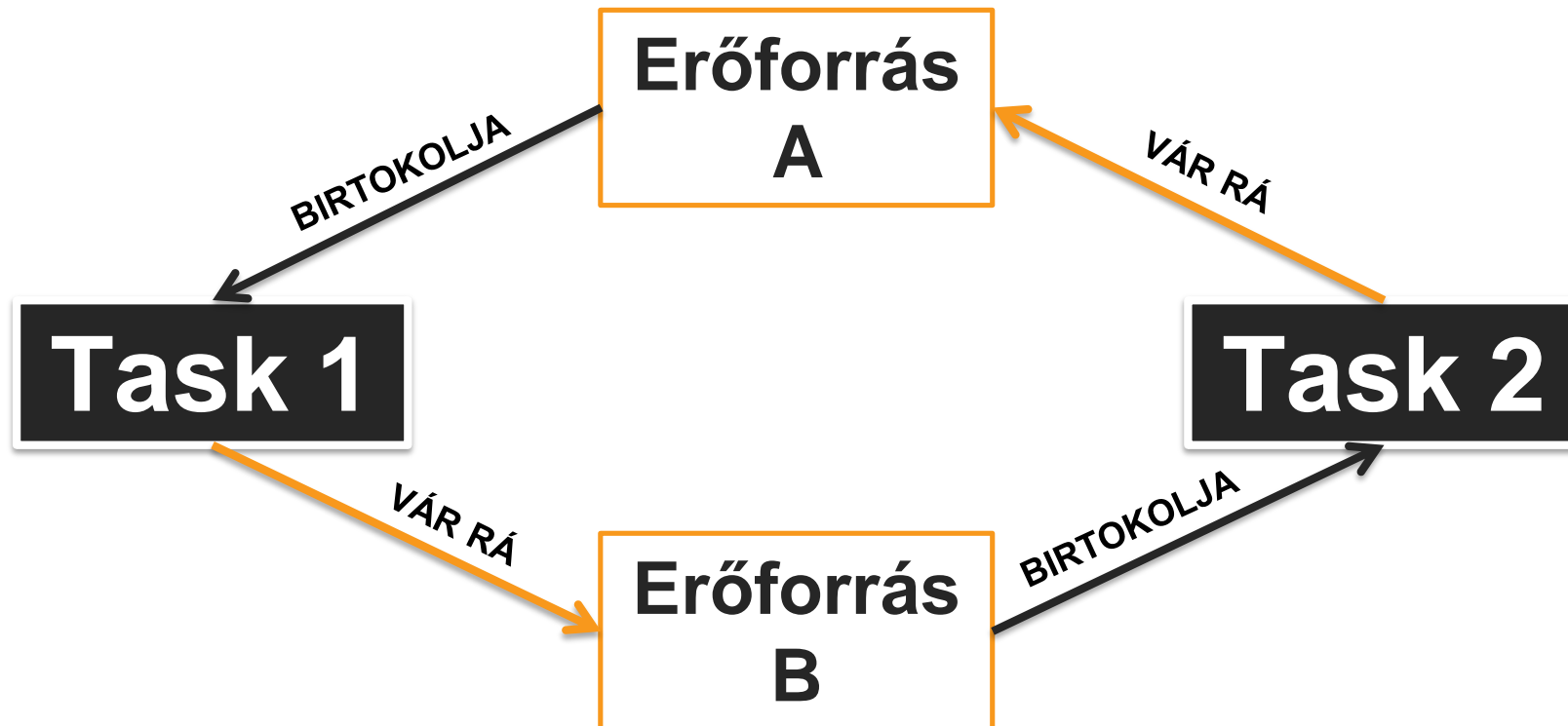
KÖR

```
mutex semaphor;
task move_square()
{
    while (true)
    {
        Acquire(semaphor);
        OnFwd(OUT_AC, 75);
        Wait(1000);
        OnRev(OUT_C, 75);
        Wait(500);
        Release(semaphor);
    }
}
```

```
task check_sensors()
{
    while (true)
    {
        if (SENSOR_1 == 1)
        {
            Acquire(semaphor);
            OnRev(OUT_AC, 75);
            Wait(500);
            OnFwd(OUT_A, 75);
            Wait(500);
            Release(semaphor);
        }
    }
}
```

Konkurens folyamatok vezérlése

- Mire kell figyelni?
 - Szemafor foglalása legyen
 - Szemafor felszabadítása legyen
 - Holtpont



1. Készíts olyan programot, amely az NXT-t háromszög úton vezérli, de ha az NXT túl közel kerülne valamihez 180° -ot fordul.
2. Egészítsük ki a feladatot úgy, hogy az NXT tegyen 90° -os fordulatot jobbra, amennyiben piros színt lát a padlón.
3. Egészítsük ki a fenti feladatot úgy, hogy az NXT touch szenzorát megnyomva az NXT leáll, majd kiírja az LCD kijelzőn a feladat végrehajtása közben megtett út hosszát cm-ben.