

Programozási nyelvek I. 9. gyakorlat

Balogh Ádám
bas@elte.hu

Eötvös Loránd Tudományegyetem
Informatikai Kar

Segítség a 2. feladathoz

- Kritikus szakasz: az a kódrészlet, amelyben a taszk a közös, kizárólagos erőforráshoz nyúl
- Egyszerre egy taszk lehet a kritikus szakaszban
- *Dijkstra*-féle szemaforok bináris változata:
 P művelet (*proberen*) jelzi a belépési kísérletet a kritikus szakasz, a V művelet (*verhogen*) pedig a kilépést onnan

2004. november 25.

Programozási nyelvek I. - 9. gyakorlat

4

Segítség az 1. feladathoz

- Versenyhelyzet: két taszk ugyanahhoz a kizárólagos erőforráshoz próbál meg hozzáférni úgy, hogy legalább az egyik változtatni is próbál rajta
- Kizárólagos erőforrások: változók, állományok (beleértve a szabványos kimenetet és bemenetet is)

2004. november 25.

Programozási nyelvek I. - 9. gyakorlat

2

2. feladat

Javítsd ki az előző programot egy *Szemafor* nevű taszk segítségével, ami egy *Dijkstra*-féle szemafor bináris változatát valósítja meg! A javított program neve legyen *Jancsi_Julcsa5*! Figyeld meg ismét a futás eredményét!

2004. november 25.

Programozási nyelvek I. - 9. gyakorlat

5

1. feladat

Írj egy saját karakterlánc kiíró eljárást amely karakterenként ír ki egy karakterláncot, és ezt használd a legutóbbi *Jancsi_Julcsa* programban a *Put_Line* helyett! Az új program neve legyen *Jancsi_Julcsa4*! Figyeld meg a futás eredményét! (Ha túl gyors, irányítsd át adatállományba!)

2004. november 25.

Programozási nyelvek I. - 9. gyakorlat

3

2. feladat megoldása (1)

```
jancsi_julcsa5.adb:
...
  task Szemafor is
    entry P; entry V;
  end Szemafor;
...
  task body Szemafor is
  begin
    accept P; accept V;
  end Szemafor;

(folyt. köv.)
```

2004. november 25.

Programozási nyelvek I. - 9. gyakorlat

6

2. feladat megoldása (2)

```
jancsi_julcsa5.adb:
(folyt.)
task Kiir is
...
  for I in Positive'Range loop
    Szemafor.P;
    Kiir ( Nev.all & ...
          Szemafor.V;
    ...
  end loop;
end Kiir;
...
```

2004. november 25.

Programozási nyelvek I. - 9. gyakorlat

7

3. feladat

Javítsd ki az Jancsi_Julcsa4 programot egy Vedett nevű védett objektum segítségével, ami a Kiir eljárást tartalmazza! A javított program neve legyen Jancsi_Julcsa6! Figyeld meg most is a futás eredményét!

2004. november 25.

Programozási nyelvek I. - 9. gyakorlat

10

Segítség a 3. feladathoz (1)

- Védett egység: egyszerre csak egy eljárás lehet benne aktív
- Taszkhoz hasonlóan nem önálló fordítási egység
- Specifikáció:
protected vagy protected type Név is
 Deklarációk
end Név;

2004. november 25.

Programozási nyelvek I. - 9. gyakorlat

8

3. feladat megoldása (1)

```
jancsi_julcsa6.adb:
...
protected Vedett is
  procedure Kiir ( S : in String );
end Vedett;
...
task body Vedett is
  procedure Kiir ( S : in String ) is
    ...
  end Kiir;
end Vedett;
(folyt. köv.)
```

2004. november 25.

Programozási nyelvek I. - 9. gyakorlat

11

Segítség a 3. feladathoz (2)

- Törzs:
protected body Név is
 Definíciók
end Név;
- Taszkokhoz hasonlóan itt is lehet a törzs separate

2004. november 25.

Programozási nyelvek I. - 9. gyakorlat

9

3. feladat megoldása (2)

```
jancsi_julcsa6.adb:
(folyt.)
task Kiir is
...
  for I in Positive'Range loop
    Vedett.Kiir ( Nev.all & ...
    ...
  end loop;
end Kiir;
...
```

2004. november 25.

Programozási nyelvek I. - 9. gyakorlat

12