Задачи теория за първо контролно за курса 'Операционни системи', 23-31.03.2019 г.

Задача 4. СИ Всеки от процесите `P` и `Q` изпълнява поредица от три инструкции:

```
process P                process Q
  p_1                      q_1
  p_2                      q_2
  p_3                      q_3
```

Осигурете чрез два семафора синхронизация на `P` и `Q` така, че да са изпълнени едновременно следните времеви зависимости:
(1) инструкция `p_1` да се изпълни преди `q_2`
(2) инструкция `q_2` да се изпълни преди `p_3`
(3) инструкция `q_1` да се изпълни преди `p_2`
(4) инструкция `p_2` да се изпълни преди `q_3`

Забележка: За решение с повече семафори ще получите 20 точки.

Задача 4. КН1 Всеки от процесите `P`, `Q` и `R` изпълнява поредица от три инструкции:

```
process P                process Q                process R
  p_1                      q_1                      r_1
  p_2                      q_2                      r_2
  p_3                      q_3                      r_3
```

Осигурете чрез семафори синхронизация на `P`, `Q` и `R` така, че да са изпълнение едновременно условията:
(1) инструкция `p_1` да се изпълни преди `q_2` и `r_2`.
(2) инструкция `p_3` да се изпълни след `q_2` и `r_2`.

Задача 4. КН2 Всеки от процесите `P`, `Q` и `R` изпълнява поредица от две инструкции:

```
process P         process Q         process R
  p_1               q_1               r_1
  p_2               q_2               r_2
```

Осигурете чрез три семафора синхронизация на `P`, `Q` и `R` така, че отделните инструкции да се изпълнят в следния времеви ред:
`p_1, q_1, r_1, p_2, q_2, r_2`

Примерни решения

Задача 4. СИ Условията (1) и (3) определят времева среща (randevouz) на процесите след първата им инструкция.

Аналогично, (2) и (4) определят randevouz на процесите след втората им инструкция.

За двате срещи използваме два семафора – `t1` и `t2`, инициализираме ги с блокиращо начално състояние:

```
semaphore t1,t2
t1.init(0)
t2.init(0)
```

Добавяме в кода на процесите `P` и `Q` синхронизиращи инструкции:

```
process P                process Q
  p_1                      q_1
  t1.signal()              t2.signal()
  t2.wait()                t1.wait()
  p_2                      q_2
  t1.signal()              t2.signal()
  t2.wait()                t1.wait()
  p_3                      q_3
```

Инструкцията `q_2` ще се изпълни след като броячът на семафора `t1` стане положителен. Това се случва след изпълнението на ред `t1.signal()`, който следва инструкция `p_1`.

Аналогично, инструкцията `p_2` ще се изпълни след като броячът на семафора `t2` стане положителен. Това се случва след изпълнението на ред `t2.signal()`, който следва инструкция `q_1`.

По подобен начин ще се развият събитията и след вторите инструкции.

Лесно се вижда, че след първото randevouz стойностите на броячите в семафорите ще са 0 и процесите коректно ще реализират втората среща със същите семафори.

Задача 4. КН1 За синхронизация използваме семафори `s`, `t` и `u`, инициализираме ги така:

```
semaphore s, t, u
s.init(0)
t.init(0)
u.init(0)
```

Добавяме в кода на процесите `P`, `Q` и `R` синхронизиращи инструкции:

```
process P                process Q                process R
  p_1                      q_1                      r_1
  s.signal()               s.wait()                 s.wait()
  p_2                      s.signal()               s.signal()
  t.wait()                 q_2                      r_2
  u.wait()                 t.signal()               u.signal()
  p_3                      q_3                      r_3
```

Всяка от инструкциите `q_2` и `r_2` може да се изпълни след като съответният процес премине бариерата `s.wait()`.

Това се случва за пръв път след изпълнението на ред `s.signal()` в процеса `P`, който следва инструкция `p_1`. Така изпълнението на `p_1` преди `q_2` и `r_2` е гарантирано.

Да допуснем, че процесът `Q` преминава през инструкцията си `s.wait()` преди процеса `R`. Веднага след това той изпълнява `s.signal()`, което ще позвали и на `R` да премине през своята инструкция `s.wait()`. Така ще се осигури изпълнението и на двете инструкции `q_2` и `r_2`.

Аналогична е ситуацията, когато `R` преминава през `s.wait()` преди процеса `Q`.

Работата със семафорите `t` и `u` осигурява изпълнението на условие (2).

Задача 4. КН2 Използваме семафорите `t1`, `t2` и `t3`, инициализираме ги така:

```
semaphore t1, t2, t3
t1.init(1)
t2.init(0)
t3.init(0)
```

Добавяме в кода на процесите синхронизиращи инструкции:

```
process P                process Q                process R
  t1.wait()                t2.wait()                t3.wait()
  p_1                      q_1                      r_1
  t2.signal()              t3.signal()              t1.signal()
  t1.wait()                t2.wait()                t3.wait()
  p_2                      q_2                      r_2
  t2.signal()              t3.signal()              t1.signal()
```