Поправка за дисциплина “Операционни системи” (КН), СУ, ФМИ, 28.08.2019 г.

Задача 1, КН, 28.08

Множество паралелно работещи копия на процесите `P` и `Q` изпълняват поредица от две инструкции:

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

Осигурете чрез семафори синхронизация на работещите копия, така че:

Инструкцията `q_2` на всяко от работещите копия на `Q` да се изпълни след като инструкция `p_1` е завършила изпълнението си в поне 3 работещи копия на `P`.

Упътване: Освен семафори, ползвайте и брояч.

Задача 2, КН, 28.08

Опишете реализацията на комуникационна тръба (pipe) чрез семафори.

Предполагаме, че тръбата може да съхранява до n байта, подредени в обикновена опашка.

Тръбата се ползва от няколко паралелно работещи изпращачи/получатели на байтове.

Процесите изпращачи слагат байтове в края на опашката, получателите четат байтове от началото на опашката.

Упътване: В теорията на конкурентното програмиране задачата е известна като "producer-consumer problem".

Примерни решения

Задача 1

За исканите в условието синхронизации използваме брояч `cnt` и два семафора – `m1` и `m2`, инициализираме ги така:

```
semaphore m1, m2
m1.init(1)
m2.init(0)
int cnt=0
```

Добавяме в кода на процеса `P` синхронизиращи инструкции:

```
process P
  p_1
  m1.wait()
   cnt=cnt+1
   if cnt=3 m2.signal()
  m1.signal()
  p_2
```

Добавяме в кода на процеса `Q` синхронизиращи инструкции:

```
process Q
  q_1
  m2.wait()
  m2.signal()
  q_2
```

Семафорът `m1` ползваме като мутекс, който защитава брояча.

Стойността на `cnt` е равна на броя копия на процеса `P`, които са изпълнили своята първа инструкция.

Семафорът `m2` блокира изпълнението на инструкция `q_2`.

Когато третото копие на процеса `P` изпълни `p_1`, към семафора `m2` се подава сигнал, който го деблокира и позволява на всички копия на `Q` да изпълнят втората си инструкция.