

## Задачи със семафори ОС Теория 2020г

### Примерни задачи\*

**78зад.** 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.

#### Решение:

За да реализираме ком. тръба ще използваме следните инструменти:

- опашка (или масив)  $Q$  с  $n$  елемента – тъй като е възможно да имаме бърз и бавен процес (единият чете бързо, другият пише бавно например) е нужно опашката да бъде по-дълга, за да не се приспи по-бързия процес (приспиването е бавна операция, тъй като включва в себе си смяна на контекста). В началото тази опашка е празна.
- free\_bytes семафор, който ще индикира за свободни байтове в опашката  $Q$
- ready\_bytes семафор, който ще индикира колко байта са готови за четене (до колко е запълнена опашката  $Q$ )
- mutex\_read мутекс, който ще защитава критичната секция за четене
- mutex\_write мутекс, който ще защитава критичната секция за писане

#### Инициализация:

free\_bytes.init( $n$ )

ready\_bytes(0)

mutex\_read(1)

mutex\_write(1)

\* задачите от контролните от „материали на Скелета“ съвпадат с примерните задачи, качени в мудъл, затова ще използвам тяхната номерация

```

READ
byte b
p_1
p_2
:
ready_bytes.wait()
mutex_read.wait()
    b = Q.get()
mutex_read.signal()
free_bytes.signal()
:

```

```

WRITE
byte b
q_1
q_2
:
free_bytes.wait()
mutex_write.wait()
    Q.put(b)
mutex_write.signal()
ready_bytes.signal()
:

```

Процесът, който чете, първоначално ще проверява дали има готови за четене байтове. Ако няма, той ще се приспи и ще чака да постъпят такива. Ако има, той ще провери дали някое друго копие на Р не чете в момента. Ако да, той отново се приспива. Ако не, той ще прочете байт и ще го запише в променливата b. След това ще сигнализира, че в опашката има още един свободен байт чрез подаване на сигнал на семафора free\_bytes.

Процесът, който пише, първоначално ще провери дали има свободни байтове в опашката. Ако няма, ще се приспи и ще чака да освободят байтове. Ако има, ще трябва да провери дали някое друго копие на пишещ процес не пише в момента в опашката. Ако да, то отново процесът ще се приспи. Ако не, ще сложи байта си b в опашката и ще сигнализира, че още един байт е готов за четене, което се осъществява чрез подаване на сигнал на семафора ready\_bytes.

Кодовете по-горе, показващи как може да се реализира ком. тръба чрез семафори са валидни само за синхронни вх-изх операции, тъй като процесите се приспиват, когато има недостиг на байтове и по този начин се изчаква, за да може да се окомплектоват.