

Име: \_\_\_\_\_ ФН: \_\_\_\_\_ Група: \_\_\_\_\_

Теоретични задачи за специалност СИ, 21.08.2021 г.:

**Задача 1, (20 точки)**

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

**Задача 2, (20 точки)**

Каква задача решава инструментът spinlock (активно изчакване)?

Опишете хардуерните инструменти, необходими за реализацията на spinlock.

В кои ситуации не бива да се ползва spinlock?

## Примерни решения

### Задача 1.

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()

**Задача 2.** Spinlock се ползва за достъп до споделени данни в режим на взаимно изключване. Така се предотвратява разрушаването им, когато паралелно работещи процеси временно нарушат структурата им и създават условия за race condition.

Реализацията на spinlock използва байт(бит) `lock`, разположен в споделената памет. Стойността на `lock` показва дали паметта се ползва (`lock=1`) или е свободна (`lock=0`).

Преди изпълнението на критичната секция макросът `spin_lock(lock)` проверява и сменя с 1 стойността на `lock` в цикъл, докато завари стойност 0. Това става със специална атомарна инструкция, примерно `test_and_set`.

След изпълнението на критичната секция макросът `spin_unlock(lock)` сменя с 0 стойността на `lock`, така освобождава споделената памет.

Преди заемането на ресурса се забраняват прекъсванията, а след освобождаването се разрешават, с цел по-бързо изпълнение на критичната секция и защита от второ влизане в `spin_lock(lock)`.

Операциите по манипулиране на прекъсванията, както и `test_and_set` са специфични хардуерни инструменти, без които не може да се направи удобна и бърза реализация на spinlock.

Spinlock не бива да се ползва при защита на критична секция, която може да продължи дълго време или да извика отново `spin_lock(lock)`.