

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

Теоретични задачи за специалност СИ, 19.06.2021 г.:

**Задача 1,** (20 точки)

Всеки от процесите P, Q и R изпълнява поредица от инструкции:

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

Процесите P и Q са единични, процесът R се изпълнява в много копия.

Осигурете чрез семафори синхронизация на P, Q и R така, че да се изпълнят едновременно следните изисквания:

(а) Всички инструкции на P и Q да се изпълнят преди инструкция r\_1 на всяко копие на R.

(б) Процесите P и Q да се изпълнят ефикасно, т.е. да е възможно паралелното им изпълнение, без да се изчакват.

**Задача 2,** (20 точки)

Една от класическите задачи за синхронизация се нарича „Задача за читателите и писателите“ (readers-writers problem).

(а – 7 точки) Опишете условието на задачата.

(б – 13 точки) Опишете решение, използващо семафори.

## Примерни решения

**Задача 1. решение 1** За синхронизация използваме семафори `m` и `s`, и брояч `cnt`, инициализираме ги така:

```
semaphore m, s
m.init(1)
s.init(0)
int cnt=0
```

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

process P	process Q	process R
p_1	q_1	s.wait()
p_2	q_2	s.signal()
m.wait()	m.wait()	r_1
cnt++	cnt++	r_2
if (cnt==2)	if (cnt==2)	
s.signal()	s.signal()	
m.signal()	m.signal()	

**Задача 1. решение 2** За синхронизация използваме семафори `sp` и `sq`, инициализираме ги така:

```
semaphore sp, sq
sp.init(0)
sq.init(0)
```

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

process P	process Q	process R
p_1	q_1	sp.wait()
p_2	q_2	sp.signal()
sp.signal()	sq.signal()	sq.wait()
		sq.signal()
		r_1
		r_2

**Задача 1. решение 3** За синхронизация използваме семафори `sp` и `sq`, инициализираме ги така:

```
semaphore sp, sq
sp.init(0)
sq.init(0)
```

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

process P	process Q	process R
p_1	q_1	sq.wait()
p_2	q_2	sq.signal()
sp.signal()	sp.wait()	r_1
	sq.signal()	r_2