

Задача 1. Процесът P и няколко копия на процесът Q изпълняват поредица от 3 инструкции:

process P	process Q
p_1	q_1
p_2	q_2
p_3	q_3

Осигурете чрез семафори синхронизация на P и копията на Q така, че инструкцията p_1 да се изпълни преди инструкцията q_2 на всяко копие на Q . Освен това искаме след изпълнението на p_1 , всички копия на Q да завършат изпълнението си.

Решение. За синхронизацията ще използваме семефор t , който ще инициализираме с блокиращо начално състояние:

```
semaphore t
t.init(0)
```

Добавяме в кода на процеса P и всички копия на процесът Q синхронизиращи инструкции:

process P	process Q
p_1	q_1
$t.\text{signal}()$	$t.\text{wait}()$
	$t.\text{signal}()$
p_2	q_2
p_3	q_3

Когато даден процес Q започнат да изпълнява инструкциите си, те ще се блокират след като изпълнят инструкцията си q_1 и ще изчакат първо процесът P да изпълни инструкцията си p_1 , след което ще се подаде сигнал, че някой от процесите Q може да изпълни инструкцията си q_2 и каскадно след това всеки процес Q ще позволява на още един процес Q да си продължи изпълнението след инструкцията q_1 . По този начин ще гарантираме, че първо ще се изпълни инструкцията p_1 на P , след което всеки един процес Q ще завърши изпълнението си и рано или късно.

Задача 2. Всеки от процесите P , Q и R изпълнява поредица от 3 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 се изпълнят следните изисквания:

- (а) Инструкция p_1 да се изпълни преди q_2 и r_2 ;
- (б) Ако q_2 се изпълни преди r_2 , то и q_3 да се изпълни преди r_2 ;
- (в) Ако r_2 се изпълни преди q_2 , то и r_3 да се изпълни преди q_2 .

Решение. За синхронизацията ще използваме семафор t , който ще инициализираме с блокиращо начално състояние:

```
semaphore t
t . 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
$t . \text{signal}()$	$t . \text{wait}()$	$t . \text{wait}()$
p_2	q_2	r_2
p_3	q_3	r_3
	$t . \text{signal}()$	$t . \text{signal}()$

Всяка от инструкциите q_2 и r_2 може да се изпълни след като броячът на семафора t стане положителен.

Това се случва за първи път след изпълнението на ред $t . \text{signal}()$ в процеса P , който следва инструкция p_1 . Така гарантираме изпълнението на изискване (а).

След като броячът на семафора стане 1, един от процесите Q и R ще достигне пръв до ред $t . \text{wait}()$ и ще го нулира отново.

Ако процесът Q пръв достигне инструкцията $t . \text{wait}()$, той ще изпълни инструкции q_2 и q_3 , а процесът R ще чака ново увеличение на брояча на семафора, което се случва след изпълнението на последния ред $t . \text{signal}()$ в процеса Q . Така гарантираме изпълнението на изискване (б).

Ако процесът R пръв достигне инструкцията $t . \text{wait}()$, той ще изпълни инструкции r_2 и r_3 , а процесът Q ще чака ново увеличение на брояча на семафора, което се случва след изпълнението на последния ред $t . \text{signal}()$ в процеса R . Така гарантираме изпълнението на изискване (в).

Задача 3. Всеки от процесите P и Q изпълнява поредица от 3 инструкции:

process P	process Q
p_1	q_1
p_2	q_2
p_3	q_3

Осигурете чрез семафори синхронизация на P и Q така, че p_1 да се изпълни преди q_2 , а q_2 да се изпълни преди p_3 .

Решение. За синхронизацията ще използваме семафори \mathbf{t} и \mathbf{r} , които ще инициализираме с блокиращи начални състояния:

```
semaphore  $\mathbf{t}$ , semaphore  $\mathbf{r}$ 
 $\mathbf{t}.\text{init}(0)$ ,  $\mathbf{r}.\text{init}(0)$ 

process  $P$       process  $Q$ 
   $p_1$            $q_1$ 
 $\mathbf{t}.\text{signal}()$    $\mathbf{t}.\text{wait}()$ 
   $p_2$            $q_2$ 
 $\mathbf{r}.\text{wait}()$      $\mathbf{r}.\text{signal}()$ 
   $p_3$            $q_3$ 
```

Когато процесът Q се опита да изпълни инструкцията си q_2 , семафорът \mathbf{t} ще се блокира и ще се освободи едва когато инструкцията p_1 се изпълни. По този начин гарантираме изпълнението на p_1 преди изпълнението на q_2 . Дуално, когато P се опита да изпълни инструкцията си p_3 , семафорът \mathbf{r} ще се блокира и ще се освободи едва когато инструкцията q_2 се изпълни. По този начин гарантираме изпълнението на q_2 преди изпълнението на p_3 .

Задача 4. Всеки от процесите P , Q и R изпълнява поредица от 3 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 се изпълнят следните изисквания:

- (а) Инstrukция p_1 да се изпълни преди instrukция q_2 и instrukция q_2 да се изпълни преди instrukция r_3 ;
- (б) Инstrukция q_1 да се изпълни преди instrukция r_2 ;
- (в) Инstrukция r_1 да се изпълни преди instrukция p_2 и instrukция p_2 да се изпълни преди instrukция r_3 .

Решение. Искаме да са удовлетворени следните неравенства във времето на изпълнение на инструкциите:

- (а) $p_1 < q_2 < r_3$;
- (б) $q_1 < r_2$;
- (в) $r_1 < p_2 < r_3$.

За синхронизацията ще използваме семафори s , t , u , v и w , които ще инициализираме с блокиращи начални състояния:

semaphore s , semaphore t , semaphore u , semaphore v , semaphore w
 $s.init(0)$, $t.init(0)$, $u.init(0)$, $v.init(0)$, $w.init(0)$

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

Когато процесът Q се опита да изпълни instrukция q_2 преди instrukция p_1 на процеса P , Q ще се блокира от $s.wait()$ и ще изчака изпълнението на instrukцията p_1 да му подаде сигнал за събуждане. Аналогично, когато процеса R се опита да изпълни instrukция r_3 преди instrukция q_2 на процеса Q , R ще се блокира от $w.wait()$ и ще изчака изпълнението на instrukцията q_2 да му подаде сигнал за събуждане. По този начин чрез семафорите s и w осигоряваме изпълнението на условие (а). Чрез аналогични разсъждения проследяваме осигоряването на условие (б) чрез семафор t и осигуряването на условие (в) чрез останалите два семафора u и v .

Задача 5. Няколко копия на процесът P и няколко копия на процесът Q изпълняват поредица от 2 инструкции:

process P	process Q
p_1	q_1
p_2	q_2

Осигурете чрез семафори синхронизация на процесите P и Q така, че да се изпълни поне 3 пъти инструкция p_1 преди да се изпълни първата инструкция q_2 .

Решение. За синхронизацията ще използваме семафор t , който ще инициализираме с блокиращо начално състояние и мутекс m , който да пази брояча от прекомерно използване (предпазва от race condition):

semaphore t	
$t.init(0)$	
$cnt = 0$	
process P	process Q
p_1	q_1
$m.wait()$	$t.wait()$
if ($cnt < 4$) then	$t.signal()$
$cnt = cnt + 1$	
if ($cnt == 3$) then	
$t.signal()$	
$cnt = cnt + 1$	
$m.signal()$	
p_2	q_2

Преди първата инструкция q_2 да се изпълни, семафорът t ще приспи процеса Q , който е опитал да изпълни q_2 . Копията на процеса P ще изпълняват инструкциите си и след 3-тото изпълнение на инструкция p_1 , броячът cnt ще е равен на 3 и ще се подаде сигнал на t . Така първата инструкция q_1 ще се изпълни и каскадно ще позволи на други инструкции q_1 да се изпълняват. Освен това, броячът cnt ще се инкрементира още веднъж и ще стане равен на 4, което ще направи така, че да не се влиза в нито една от условните конструкции. Брояча е защитен чрез мутекс от прекомерна употреба.