

## Zadatak za lab3, student: \_\_\_\_\_

1. Proučiti kôd s desne strane i odgovoriti na pitanja.  
Pretpostaviti da su monitor i red uvjeta inicijalizirani prije stvaranja dretvi te da se dretve nasumično stvaraju (istovremeno može biti nula dretvi, malo ili puno dretvi "ista ili različita tipa").

- a. (1) Što treba dodati (npr. "iza retka xx dodaj: ..."), što maknuti ("bez retka xx") a da program radi ispravno, da se ne dogodi zastoj (za sada ignorirati "nelogičnosti u redoslijedu ispisa")?

- b. (2) Što još treba promijeniti (maknuti/dodati, premjestiti) a da bi ponašanje bilo sljedeće (sve navedeno):

- naizmjeničan ispis "Krenuo ping" i "Krenuo pong"
- rad dretvi ("sleep") neka bude paralelan
- nakon 25 ispisa neka dretve završe s radom.

```
1 pthread_mutex_t m;
2 pthread_cond_t red;
3
4 char *ispis[] = {"ping", "pong"};
5 int na_redu = 0;
6 int kraj = 0;
7
8 void *dretva (void *p)
9 {
10     int tip = *((int *) p); //0 ili 1
11
12     pthread_mutex_lock (&m);
13     while (!kraj) {
14         printf("Krece dretva %s\n", ispis[tip]);
15         while (na_redu != tip) {
16             pthread_cond_wait (&red, &m);
17             pthread_mutex_lock (&m);
18         }
19         pthread_mutex_unlock (&m);
20
21         printf("Krenuo %s\n", ispis[tip]);
22         sleep(5);
23
24         pthread_mutex_lock (&m);
25         na_redu = 1 - tip;
26     }
27     pthread_mutex_unlock (&m);
28
29     return NULL;
30 }
```

2. (0,5) Dretva stvorena funkcijom `pthread_create(&id, NULL, poc_fun, NULL)` će završiti svoj rad (ili biti prekinuta i maknuta iz sustava):

- ako ona pozove `exit`
- ako ona pozove `pthread_exit`
- ako ona pozove `pthread_join`
- ako ona završi početnu funkciju (`poc_fun`) – izađe iz nje
- ako početna dretva, koja kreće s funkcijom `main`, pozove `pthread_join`
- ako početna dretva, koja kreće s funkcijom `main`, izađe iz te funkcije (`main`) ili na drugi način završi s radom (npr. pozove `exit`)

3. (0,5) Nekim pozivima dretva ulazi u monitor, s nekim izlazi, s nekim privremeno izlazi. Nakon kojih poziva će dretva sigurno nastaviti raditi unutar monitora (kada je kod ispravno napisan)?

- `pthread_mutex_lock`
- `pthread_mutex_unlock`
- `pthread_cond_wait`
- `pthread_cond_signal`
- `pthread_mutex_init`
- `pthread_cond_init`

## Zadatak za lab3, student: \_\_\_\_\_

4. Proučiti kôd s desne strane i odgovoriti na pitanja.  
Pretpostaviti da su monitor i red uvjeta inicijalizirani prije stvaranja dretvi te da se dretve nasumično stvaraju (istovremeno može biti nula dretvi, malo ili puno dretvi "ista ili različita tipa").

- a. (1) Što treba dodati (npr. "iza retka xx dodaj: ..."), što maknuti ("bez retka xx") a da program radi ispravno, da se ne dogodi zastoje (za sada ignorirati "nelogičnosti u redoslijedu ispisa")?

**crno u kodu desno**

**bez retka 17**

**dodati pthread\_cond\_signal(&red)**

- b. (2) Što još treba promijeniti (maknuti/dodati, premjestiti) a da bi ponašanje bilo sljedeće (sve navedeno):

- naizmjeničan ispis "Krenuo ping" i "Krenuo pong"
- rad dretvi ("sleep") neka bude paralelan
- nakon 25 ispisa neka dretve završe s radom.

**Sve se može ostvariti na razne načine – ovdje je prikazan jedan – crveno u kodu desno**

```
1 pthread_mutex_t m;  
2 pthread_cond_t red;  
3  
4 char *ispis[] = {"ping", "pong"};  
5 int na_redu = 0;  
6 int kraj = 0;  
7  
8 void *dretva (void *p)  
9 {  
10     int tip = *((int *) p); //0 ili 1  
11  
12     pthread_mutex_lock (&m);  
13     while (kraj < 25) {  
14         printf("Krece dretva %s\n", ispis[tip]);  
15         while (na_redu != tip && kraj < 25) {  
16             pthread_cond_wait (&red[tip], &m);  
17             pthread_mutex_lock (&m);  
18         }  
19         pthread_mutex_unlock (&m);  
20         kraj++;  
21         if (kraj > 25) break;  
22         printf("Krenuo %s\n", ispis[tip]);  
23         sleep(5);  
24         pthread_mutex_lock (&m);  
25         na_redu = 1 - tip;  
26         pthread_cond_signal(&red[na_redu]);  
27         pthread_mutex_unlock (&m);  
28         sleep(5);  
29         pthread_mutex_lock (&m);  
30     }  
31     pthread_mutex_unlock (&m);  
32     pthread_cond_broadcast(&red[0]);  
33     pthread_cond_broadcast(&red[1]);  
34  
35     return NULL;  
36 }
```

5. (0,5) Dretva stvorena funkcijom `pthread_create(&id, NULL, poc_fun, NULL)` će završiti svoj rad (ili biti prekinuta i maknuta iz sustava):
- ako ona pozove `exit`
  - ako ona pozove `pthread_exit`
  - ako ona pozove `pthread_join`
  - ako ona završi početnu funkciju (`poc_fun`) – izađe iz nje
  - ako početna dretva, koja kreće s funkcijom `main`, pozove `pthread_join`
  - ako početna dretva, koja kreće s funkcijom `main`, izađe iz te funkcije (`main`) ili na drugi način završi s radom (npr. pozove `exit`)
6. (0,5) Nekim pozivima dretva ulazi u monitor, s nekim izlazi, s nekim privremeno izlazi. Nakon kojih poziva će dretva sigurno nastaviti raditi unutar monitora (kada je kod ispravno napisan)?
- `pthread_mutex_lock`
  - `pthread_mutex_unlock`
  - `pthread_cond_wait`
  - `pthread_cond_signal`
  - `pthread_mutex_init`
  - `pthread_cond_init`