



PARTY ROOM

-arhitektura racunara i operativni sistemi-projekat-

Mentor:
Maja Miljanić

Student:
Dimitrije Ostojić 225/20

Beograd, jul 2022.

Party room problem	3
Rešenje problema	3
Rešenje problema u kodu	3
Main metoda	4
Nit studenta prodekana	4
Nit studenata koji prave žurku	5
Testiranje programa	6
Literatura	9

Party room problem

Problem je uzet iz knjige "The Little book of Semaphores".

U problemu postoje 2 niti, jedna za studente koji prave žurku i jedna za studenta prodekana.

Ograničenja koja su data u problemu:

1. Bilo koji broj studenata može biti u prostoriji u isto vreme.
2. Student prodekan može ući u prostoriju jedino ako je ona prazna (da izvrši pregled sobe) ili ako u prostoriji ima više od 50 studenata (da prekine žurku).
3. Dok je student prodekan u prostoriji nijedan student ne sme ući, već studenti izlaze iz sobe.
4. Student prodekan ne može izaći iz sobe dok ne izađu svi studenti.
5. Postoji samo jedan student prodekan

Zadatak: napisati kod koji će da sinhronizuje ova ograničenja za studente i studenta prodekana.

Rešenje problema

Postoje 2 situacije koje se mogu desiti u ovom problemu:

1. Student prodekan dođe prvi, pretraži sobu i ode. Dolazimo u situaciju koja nije definisana u zadatku. U ovom slučaju studenti bi dolazili u besonačnost, pa se ovaj problem naziva *starvation*, a rešićemo ga tako što ćemo pretpostaviti da je došlo 50 studenata na žurku i da su slavili celu noć.
2. Studenti koji prave žurku dođu prvi. U ovom slučaju žurka počinje i studenti dolaže dok ih ne bude 50. Student prodekan može da se pojavi pre nego što dođe 50 studenata na žurku ili kada ih bude 50. U slučaju da je došao kada je bilo manje od 50 studenata na žurci, on čeka da dođe tačno 50 i onda ulazi i rastura žurku. U slučaju da je došao kada ih je bilo 50 on odmah ulazi i rastura žurku.

Rešenje problema u kodu

Definišemo stanja studenta prodekana, tj gde se on nalazi:

```
#define dean_waiting 0
#define dean_in_the_room 1
#define dean_left 2
```

Uvodimo globalne promenljive pomoću kojih pratimo broj studenata u sobi i gde se nalazi student prodekan trenutno

```
int number_of_students;
int dean_status;
```

Takođe, uvodimo i promenljive:

```
pthread_mutex_t mutex;
pthread_cond_t students_leaving, students_comming;
```

Mutex koristimo kako bismo obezbedili da svaka nit funkcioniše normalno, bez uplitanja druge. Uslovne promenljive koristimo kako bismo sinhronizovali rad dveju niti.

Main metoda

```
number_of_students = 0;  
dean_status = -1;
```

Na početku main metode dodeljujemo vrednost 0 broju studenata iz razloga što niti još uvek nisu kreirane i nema studenata koji se nalaze u sobi. Takođe, dodeljujemo vrednost -1 statusu studenta prodekana koja označava da se on ne nalazi ni u jednom od stanja koja smo definisali na početku.

Deklarišemo niti:

```
pthread_t dean, students;
```

Vršimo inicijalizaciju mutex-a i uslovnih promenljivih:

```
pthread_mutex_init(&mutex, NULL);  
pthread_cond_init(&students_comming, NULL);  
pthread_cond_init(&students_leaving, NULL);
```

Kreiramo niti:

```
pthread_create(&students, NULL, students_thread, NULL);  
pthread_create(&dean, NULL, dean_thread, NULL);
```

Spajamo niti sa glavnim programom:

```
pthread_join(dean, NULL);  
pthread_join(students, NULL);
```

Oslobađamo prostor:

```
pthread_mutex_destroy(&mutex);  
pthread_cond_destroy(&students_comming);  
pthread_cond_destroy(&students_leaving);
```

Nit studenta prodekana

```
pthread_mutex_lock(&mutex);  
.  
.  
.  
pthread_mutex_unlock(&mutex);
```

Na početku i kraju f-je za studenta prodekana zaključavamo i otključavamo mutex kako se ne bi ometao njen rad

```
if(number_of_students>0 && number_of_students<50){  
    dean_status = dean_waiting;  
    printf("Dean is waiting\n");  
    pthread_cond_wait(&students_comming, &mutex);  
}
```

Proveravamo da li su studenti došli pre studenta prodekana. Ukoliko jesu i njihov broj je manji od 50 menjamo status studenta prodekana (on čeka da ih bude 50 na žurci) i postavljamo nit da čeka na signal da su svi studenti dosli tj da ih ima 50 na žurci.

```
if(number_of_students>=50){
    dean_status = dean_in_the_room;
    printf("Dean enters the room\n");
    pthread_cond_wait(&students_leaving, &mutex);
}
```

Proveravamo da li se na žurci već nalazi 50 i više studenata. Ukoliko je uslov ispunjen menjamo stanje studenta prodekana (on ulazi u sobu), a nit čeka na signal da su svi studenti izašli iz sobe.

```
else{
    printf("Dean is searching the room\n");
    dean_status = dean_in_the_room;
}
```

U suprotnom student prodekan je došao prvi, menjamo njegovo stanje i on pretražuje sobu.

```
dean_status = dean_left;
printf("Dean left the room\n");
```

Na kraju, menjamo stanje studenta prodekana (on izlazi iz sobe).

Nit studenata koji prave žurku

```
while(1){
    pthread_mutex_lock(&mutex);
    .
    .
    .
    pthread_mutex_unlock(&mutex);
```

Kod ove niti se odvija kroz beskonačnu petlju pri čemu zaključavamo i otključavamo mutex u svakoj iteraciji.

```
if(number_of_students<50 && dean_status!=dean_in_the_room){
    printf("Students are having a party\n");
    number_of_students++;
}
```

Proveravamo da li je broj studenata koji se nalaze u sobi manji od 50 i da li se student prodekan nalazi van sobe (samo u tom slučaju studenti mogu da ulaze u sobu). Ukoliko je uslov ispunjen povećava se broj studenata u sobi i oni nastavljaju sa žurkom.

```
if(dean_status==dean_waiting && number_of_students==50){
    pthread_cond_signal(&students_comming);
}
```

Ovo je slučaj u kome je student prodekan došao do sobe kada su studenti već počeli da prave žurku, međutim bilo ih je manje od 50. Objasnili smo ranije da će nit studenta prodekana čekati na signal da je došlo 50 studenata na žurku, a u ovom delu šaljemo taj signal. Nastaviće sa radom nit studenta prodekana i on će ući u sobu i rasturiti žurku.

```
if(dean_status==dean_in_the_room){  
    if(number_of_students>0){  
        printf("Students are leaving\n");  
        number_of_students--;  
    }  
}
```

Proveravamo da li je student prodekan u sobi. Ukoliko jeste i ukoliko je broj studenata veći od 0 studenti će jedan po jedan napuštati sobu.

```
else{  
    pthread_cond_signal(&students_leaving);  
    pthread_mutex_unlock(&mutex);  
    printf("End of the party\n");  
    break;  
}  
}
```

U slučaju da je student prodekan u sobi ali da je broj studenata jednak 0 (svi su izašli) šaljemo signal niti studenta prodekana da su svi studenti izašli iz sobe i otključavamo mutex kako bi nit studenta prodekana nastavila sa radom i student prodekan izašao iz sobe.

```
if(dean_status==dean_left && number_of_students==50){  
    printf("Students are having a party all night long\n");  
    break;  
}
```

Ovo je slučaj u kome je student prodekan stigao prvi, pretražio sobu i otišao. Kao što sam već napomenula, u ovom slučaju pretpostavljamo da će doći 50 studenata na žurku i da će slaviti celu noć. Prekidamo program jer je nit studenta prodekana već završila sa radom-

Testiranje programa

Prilikom testiranja programa najpre moramo da kompajliramo program što sam uradilo u terminalu pomoću komande:

```
gcc -o projekat projekat.c -pthread  
Pokrećemo program iz komandne linije:  
./projekat
```

Rešenja koja možemo dobiti su sledeća:

Students are having a party
Students are having a party
Dean is waiting
Students are having a party
Students are having a party

.
.
.

Dean enters the room
Students are leaving
Students are leaving
Students are leaving

.
.
.

Students are leaving
End of the party
Dean left the room

U ovom slučaju studenti dolaze prvi, a student prodekan dolazi kada je bilo manje od 50 studenata na žurci, on čeka dok ne dođe i 50. student i tada ulazi i rastura žurku.

Students are having a party
Students are having a party
Students are having a party
Students are having a party

.
.
.

Dean enters the room
Students are leaving
Students are leaving

.
.
.

Students are leaving
End of the party
Dean left the room

U ovom slučaju studenti dolaze prvi, međutim student prodekan dolazi tačno u trenutku kada je 50 studenata na žurci pa ne mora da čeka već odmah ulazi i rastura žurku.

Dean is searching the room

Dean left the room

Students are having a party

Students are having a party

.

.

.

Students are having a party

Students are having a party all night long

U ovom slučaju student prodekan je došao prvi, pretražio sobu i otisao, a studenti su slavili celu noć.

Literatura

1. "The Little Book of Semaphores" - Allen B. Downey -
<http://greenteapress.com/semaphores/LittleBookOfSemaphores.pdf>
2. <http://www.mmklab.fon.bg.ac.rs/wp-content/uploads/2016/01/CSkripta.pdf>
3. <https://stackoverflow.com/questions/1162587/what-is-starvation>