

# **Zadaća 3**

## **Sistemska programiranje**

## Zadatak 1: Implementacija višezadačnog planera (scheduler-a) sa višenitnim izvršavanjem i RAII upravljanjem

### Opis zadatka:

Zadatak podrazumijeva implementaciju planera (engl. *scheduler*) za izvršavanje zadataka u okviru jednog korisničkog procesa. Planer mora omogućiti upravljanje izvršavanjem više zadataka na osnovu unaprijed definiranog vremena pokretanja (scheduling na osnovu sistemskog sata), kao i omogućiti njihovu paralelnu obradu korišćenjem višenitnog (engl. *multithreaded*) pristupa. Svi resursi u sistemu, uključujući niti i mehanizme za sinhronizaciju, moraju biti upravljani korišćenjem RAII (engl. *Resource Acquisition Is Initialization*) paradigme, čime se obezbeđuje bezbjedno i automatsko oslobađanje zauzetih resursa.

Svaki zadatak sadrži:

- `id` – jedinstveni identifikator zadatka
- trenutak kada zadatak treba da počne
- trajanje zadatka u milisekundama
- prioritet zadatka (manja vrijednost označava viši prioritet)

Ovakav planer omogućava višestruko izvršavanje zadataka sa ograničenim brojem paralelnih niti, koji se inicijalizira sa fiksним brojem niti (`N`).

### Zahtjevi za testiranje:

U glavnom programu (`main()` funkciji), neophodno je demonstrirati funkcionalnost planera kroz kreiranje više zadataka sa različitim vremenima početka i trajanjem izvršavanja. Primjer testnog scenarija:

- Kreirati više zadataka sa različitim vremenima pokretanja i prioritetima
- Inicijalizirati thread pool sa nekoliko radnih niti
- Obezbediti korektno i vremenski ispravno izvršavanje zadataka po redu:
  - Prema `start_time`, ali uz poštovanje `priority` među zadacima koji startuju istovremeno

### Primjer ponašanja:

Zadaci:

ID	Start nakon (ms)	Trajanje (ms)	Prioritet
1	1000	3000	2
2	1000	1000	1
3	1500	2000	3
4	1500	1000	1

## FAKULTET ELEKTROTEHNIKE SISTEMSKO PROGRAMIRANJE

Redoslijed izvršavanja:

1. Zadatak 2 (startuje prvi zbog prioriteta)
2. Zadatak 1 (startuje drugi jer ima niži prioritet od 2)
3. Zadatak 4 (veći prioritet od zadatka 3, isti start time)
4. Zadatak 3 (posljednji)

### **Dodatne Napomene:**

- Opcionalna Poboljšanja:
  - Praćenje statistike o trajanju izvršavanja i broju aktivnih niti
  - Omogućavanje prekida (cancel) zadataka koji još nisu započeti