



# Multithreading

# Pojmovi

- **Multitasking** – računar može da izvrši više zadataka istovremeno
- **Multithreading** – program se sastoji od više zadataka koji se izvršavaju konkurentno

# Pojmovi

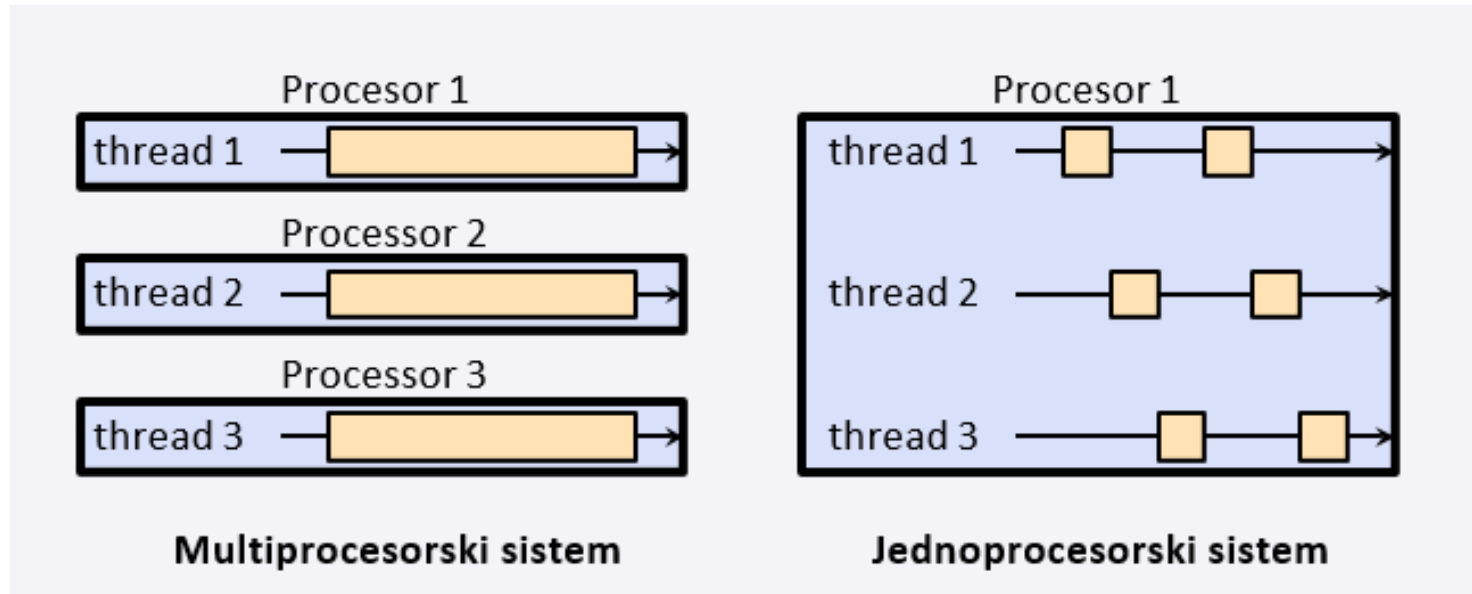
- **Task** – programska jedinica koja se izvršava nezavisno u odnosu na druge programske jedinice programa
- **Thread** – tok izvršavanja, od početka do kraja programa

# Java multithreading

- Java program može kreirati mnogo niti za pokretanje istodobnih zadataka
- Nudi bolje performanse
- Primer:
  - Program za obradu teksta vam omogućava štampanje datoteka u toku kucanja

# Java multithreading

- Primer:



# Java multithreading

- Svaki java program kreira barem jednu nit za pokretanje glavne main() metode
- Glavna nit može kreirati druge niti za pokretanje paralelnih zadataka
- Ostale niti mogu se nastaviti prokazivati i nakon što je glavna nit završena

# Java multithreading

- Zadatak za paralelno izvršavanje mora biti objekat koji implementira **Runnable** interfejs
- **Runnable** interfejs sadrži samo jednu metodu čija implementacija određuje funkciju zadatka

```
public void run();
```

# Java multithreading

- Primer:

```
public class TaskClass implements Runnable {  
  
    . . .  
  
    // Implementacija metoda run()  
    public void run() {  
        // . . .  
        // Zadaci  
        // . . .  
    }  
}
```



# Java multithreading

- Zadatak mora biti pokrenut u niti koja je objekt klase **Thread**:

```
// Kreiranje instance zadatka
TaskClass task = new TaskClass(...);
. . .
// Kreiranje niti koja pokreće
zadatak
Thread thread = new Thread(task);
. . .
// Pokretanje niti koja pokreće
zadatak
thread.start();
```

# Java multithreading

- Primer istoimenog izvršavanja klase niti i zadatka:

```
public class ThreadTask extends Thread {  
  
    . . .  
  
    // Implimentacija metoda run()  
    public void run() {  
        // . . .  
        // Telo zadatka  
        // . . .  
    }  
}
```

# Java multithreading

- Metode u klasi Thread koje kontrolišu tok izvršavanja:
  - void start()
  - boolean isAlive()
  - static void sleep(long milisec)
  - static void yield()
  - void join()
  - void setPriority(int p)

# Java multithreading

- Potpuno nezavisne niti je relativno lako stvoriti
- Zavisne niti međusobno komuniciraju i saraduju koristeći zajedničke resurse (varijable, objekte, uređaje..)
- Problem se javlja kada zajedničkom resursu pristupa više niti paralelno
- Potrebno je izvršiti sinhronizaciju niti

**Hvala na pažnji!**  
**Pitanja?**