# Paralelno programiranje

– Seminarski rad –

Aleksandar Đukić, Marija Zrnić, Iva Minić, Luka Mitrović

Matematički fakultet
Univerzitet u Beogradu

Beograd, 2022.

# Literatura

- Zasnovano na:
  Aleksandar Đukić, Marija Zrnić, Iva Minić, Luka Mitrović:
  Paralelno programiranje, 2022.
  (https://github.com/mi22092/7_TNP2022/blob/main/7_%C4%90uki%C4%87Zrni%C4%87Mini%C4%87Mitrovi%C4%87/7_%C4%90uki%C4%87Zrni%C4%87Mini%C4%87Mitrovi%C4%87.pdf)

# Pregled

# Uvod i istorija

- Pojam paralelnog programiranja
- Razlika između paralelne obrade i paralelnog programiranja

- Paralelna obrada se prvi put pominje 1950-ih
- 1962. Burroughs proizvodi sistem D825 sa 4 procesora
- 1967. Amdalov zakon nastaje na *Spring Joint* konferenciji
- 1983. *Caltech* proizvodi *The Cosmic Cube* sa 64 procesora

- 1992. *Standards for Message-Passing in a Distributed Memory Environment* radionica; 1994. *MPI* interfejs
- 1997. *OpenMP Architecture Review Board* objavljuje interfejs *OpenMP* za *Fortran*, zatim za C/C++.

# Osnove i forme paralelne obrade

- Forme paralelne obrade:
  - **Nivo bita** (eng. *bit-level*)
  - **Nivo naredbe** (eng. *instruction level*)
  - **Paralelizam podataka** (eng. *data parallelism*)
  - **Paralelizam zadataka** (eng. *task parallelism*)

| Naziv | Formula |
|---|---|
| Amdalov zakon | $S_{max} = \frac{1}{(1-p)+\frac{p}{s}}$ |
| Gustafsonov zakon | $S_{max} = N + (1 - N) * s$ |

Tabela: Modeli izračunavanja efikasnosti paralelne obrade

# Koraci paralelizacije i modeli paralelnog programiranja

- Koraci paralelizacije programa:
  - **Dekompozicija** - Razdvaja obradu na više zadataka
  - **Dodela** - Zadaci se dele među procesima, ***particionisanje***
  - **Orkestracija** - Organizuje rad
  - **Mapiranje** - Deli procese među procesorima

- **Komunikacija** - Razmena podataka između procesa koji se trenutno izvršavaju

- Modeli paralelnog programiranja:
  - **Model deljene memorije** (eng. *shared memory model*)
  - **Model prenošenja poruke** (eng. *message passing model*)
  - **Particionisani globalni adresni prostor** (eng. *partitioned global address space*) – hibridni model

# Upotrebe, prednosti i mane

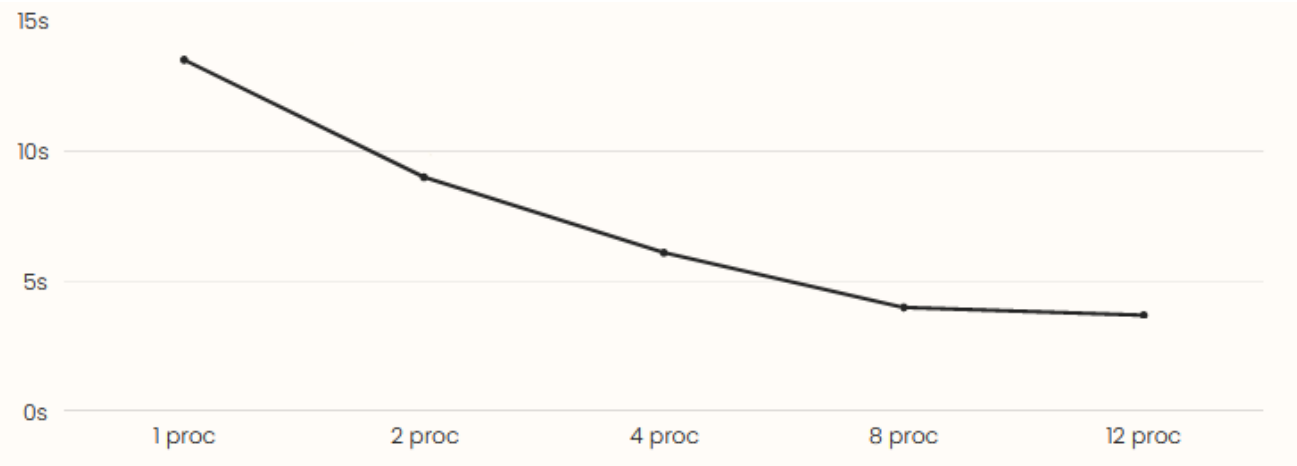
- Paralelno programiranje koristimo kada imamo velike količine podataka, kompleksne račune ili velike simulacije.
- Neke od oblasti u kojima se paralelno programiranje koristi su primenjena fizika, elektrotehnika, finansijsko i ekonomsko modeliranje, veštačka inteligencija, kvantna mehanika i druge.

- Prednosti paralelnog programiranja: brzina, poboljšan GUI (*grafički korisnički interfejs*), istovremeno pokretanje različitih logika programiranja, bolje korišćenje keš memorije i CPU resursa.
- Mane paralelnog programiranja: promena konteksta, nepredvidljivost, otežano programiranje, *data race* i *deadlock*.

# Primer - Prebrojavanje prostih brojeva u listi

- Modul *multiprocessing*, klasa *Pool*

```
pool = multiprocessing.Pool(processes=4)
nums_list = generate_list(15000, 9, 12)
nums_list = chunks(nums_list, 4)
print(sum(pool.map(count_primes, nums_list)))
```




Slika: Performanse

# Zaključak

- Pogodna tehnika za ubrzanje određenih vrsta programa
- Sjajna budućnost, velikim delom zbog fizičke ograničenosti modernih mikroprocesora