

# **Lygiagretusis programavimas (LP)**

## **P170B328**

### **Concurrent Programming**

D. Barisas,  
M. Jančiukas,  
R. Marcinkevičius  
Programų inžinerijos katedra

# Tikslai ir uždaviniai

- išnagrinėti LP principus ir ypatybes;
- supažindinti su lygiagrečiaisiais procesais, jų komunikavimu, konkuravimu ir sinchronizavimu;
- supažindinti su LP kalbomis;
- supažindinti su lygiagrečiųjų taikomųjų programų kūrimo ypatybėmis.

# Paskaitų tematika

- procesų (gijų) kūrimas, vykdymas, nutraukimas, naikinimas;
- bendra ir paskirstyta atmintis: procesų komunikavimas ir sinchronizavimas;
- lygiagretieji algoritmai;
- lygiagrečiojo programavimo kalbos ir priemonės.

# Mokymo planas

- **paskaitos:** 32 val. (2 val. per sav.);
- **lab. darbai:** 32 val. (2 val. per sav.);
- **sav. darbas:** 96 val. (lab. darbai, individualus darbas<sup>1</sup>, egzaminas);
- **kreditai:** 6.

---

<sup>1</sup> programa arba referatas

# Literatūra (knygos)

- **Peter Pacheco.** An Introduction to Parallel Programming. Elsevier, 2011.
- **Jeff Magee, Jeff Kramer.** Concurrency: State Models & Java Programs. Wiley Publishing Company, 2006.
- **Anthony Williams.** C++ Concurrency in Action: Practical Multithreading. Manning Publications, 2012.
- **Stephen Cleary.** Concurrency in C# Cookbook, 2014.
- **David B. Kirk, Wen-mei W. Hwu.** Programming Massively Parallel Processors (2nd Edition). Elsevier Inc., 2013.
- **R. Čiegis.** Lygiagretieji algoritmai ir tinklinės technologijos. Vilnius: Technika, 2005.

# Literatūra (interneto šaltiniai)

- **OpenMP.** <http://www.openmp.org/>.
- **Google Go.** The Go Programming Language.  
<http://golang.org/>.
- **JCSP.** Communicating Sequential Processes for Java.  
<http://www.cs.kent.ac.uk/projects/ofa/jcsp/>.
- **C++CSP2.** Easy Concurrency for C++.  
<http://www.cs.kent.ac.uk/projects/ofa/c++csp/>.
- **MPI.** Message Passing Interface Forum.  
<http://www.mpi-forum.org/>.
- **Paskaitų medžiaga.** <https://moodle.ktu.edu//>

# JAVA

- plačiai naudojama programavimo kalba, į kurios standartą įtrauktos lygiagrečiojo programavimo priemonės;
- lygiagrečiai vykdomos gijos;
- pradedant Java 5.0 (1.5.0) kalbos versija — naujos gijų komunikavimo ir sinchronizavimo priemonės.
- gali būti įtrauktos procesų kūrimo, komunikavimo ir sinchronizavimo priemonės.

# C, C++

- plačiai naudojamos programavimo kalbos, tačiau į jų standartą ilgą laiką nebuvo įtrauktos lygiagrečiojo programavimo priemonės;
- C++11 standarte numatytos priemonės lygiagretumui realizuoti.
- lygiagrečioms programoms kurti gali būti naudojamos funkcijų ir klasių bibliotekos;



# Kitos kalbos

- kalbos, į kurių standartą įtrauktos LP priemonės;
- NP kalbų versijos, į kurias įtrauktos LP priemonės;
- kompiuterių klasterių ir superkompiuterių programavimo paketai ir bibliotekos.

# LD ir ID programavimo kalbos

## 1 Numatytosios:

- laboratorinių darbų programoms: Java (C++, C#), C++ & OpenMP, C++ & CUDA, Google Go (Java & JavaCSP, C++ & C++CSP2);
- individualaus darbo programai: C (C++) & CUDA, C (C++) & MPI, Google Go;

## 2 Individualiai pasirenkamos<sup>2</sup>.

---

<sup>2</sup>pasirinkimą reikia derinti su dėstytoju

# Atsiskaitymų datos

## 1 Numatytas grafikas:

- LD programų demonstravimas:  
**L1** – 3, 4, 5; **L2** – 7, 8; **L3** – 11, 12; **L4** – 14, 15 sav.;
- LD gynimas:  
**L2** – 10; **L3** – 13; **L4** – 16 sav.;
- individualaus darbo programa: 12-15 sav. pagal grafiką;
- referatas: 12-15 sav. pagal grafiką;

## 2 Individualus grafikas<sup>3</sup>.

---

<sup>3</sup>grafiką reikia suderinti su dėstytoju iki spalio 1 d.

# LD ir ID programų pateikimo tvarka

- LD ir ID programų, duomenų bei rezultatų failus įkelti į LP puslapį [moodle.ktu.edu](http://moodle.ktu.edu);
- Failus įkelti:
  - LD: iki programos atsiskaitymo savaitės pabaigos;
  - ID: iki darbo pateikimo savaitės pabaigos.

# 1 lab. darbas: Procesai-gijos

- Kalbos gijos – 4 taškai (Java, C++ arba C#)
- OpenMP – 3 taškai (GNU gcc, MS VS C++);
- CUDA – 3 taškai (C, C++ & CUDA);

## 2 lab. darbas: Bendra atmintis

- Java – 3 taškai (monitoriai arba semaforai);
- OpenMP – 3 taškai (kritinės sritys arba užraktai);
- LD gynimas – 4 taškai.

## 3 lab. darbas: Paskirstyta atmintis

- Any2One – 3 taškai (JavaCSP, C++CSP, Google Go);
- One2One – 3 taškai (JavaCSP, C++CSP, Google Go);
- LD gynimas – 4 taškai.

## 4 lab. darbas: Lygiagretusis programavimas CUDA

- CUDA funkcijos – 3 taškai;
- Thrust – 3 taškai; ;
- LD gynimas – 4 taškai.



# Galutinis įvertinimas: galimi variantai

- darant individualaus darbo programą:
  - 4 lab. darbai – 40%;
  - individualus darbas – 20%;
  - egzaminas – 40%;
- rašant referatą:
  - 4 lab. darbai – 40%;
  - refer. (+pasisak.) – 20% (+10%);
  - egzaminas – 40% (–10%);
- įskaita:
  - LD ir ID įvertinimai (kiekvienas)  $\geq 5\%$ .

# Individualaus darbo programa: galimi variantai

- C++ & MPI – 20%;
- C, C++ & CUDA – 20%;
- Google Go – 20%.
- Kita kalba<sup>4</sup> – 15-20%.

---

<sup>4</sup>suderinta su dėstytoju

## Egzaminas (40%): galimi variantai

- kompiuteriu/raštu sesijos metu:
  - 1 klausimas – 10%;
  - 2 klausimas – 10%;
  - 3 klausimas – 10%;
  - 4 klausimas – 5%;
  - 5 klausimas – 5%;
- individuali užduotis sesijos metu<sup>5</sup>:
  - programos veikimas – 15%;
  - atsakymai į klausimus – 20%;
  - programos apiforminimas – 5%.

---

<sup>5</sup>yra papildomų sąlygų

# Pakartotiniai atsiskaitymai ir vėlavimo įtaka vertinimams

- Jei laiku neatsiskaitėte laboratorinio darbo ar individualaus darbo (neatvykote ar gavote įvertinimą 0-4), pakartotinai galite atsiskaityti:
  - LD1: iki lapkričio 16 d.,
  - LD2: iki lapkričio 23 d.,
  - LD3: iki gruodžio 7 d.,
  - LD4: iki gruodžio 21 d.,
  - ID: iki gruodžio 21 d.
- Pavėluotas LD pristatymas – (iki -1%) už kiekv. progr.;
- Pavėluotas ID pristatymas – (-2%) už kiekv. sav.

# LD failai

- viena programa – vienas failas;
- visi vienos programos duomenys – viename faile;
- visi vienos programos rezultatai – viename faile.

# LD programų failų forma

- programos pradžioje komentarai: LD variantas, autorius;
- komentuoti:
  - klasių paskirtį,
  - klasių kintamųjų paskirtį,
  - klasių metodų paskirtį,
  - lygiagretaus programavimo priemonių panaudojimą.

# Individualus darbas-programa (.pdf)

- titulinis lapas ir užduotis;
- užduoties analizė ir sprendimo metodas;
- programos aprašymas (trumpi klasių, duomenų struktūrų ir metodų aprašai, išsamūs procesų ir LP priemonių panaudojimo programoje aprašai);
- programos pagrindinių dalių tekstai su komentarais;
- testavimas ir programos instaliavimo bei vykdymo instrukcija;
- vykdymo laiko kitimo tyrimas;
- išvados ir literatūra.

# Individualus darbas-referatas 1: LP kalbos analizė (.pdf)

- titulinis lapas;
- kalbos vystymasis ir naudojimas:  $\approx 2$  psl.;
- NP priemonės (objektai, duomenų struktūros, priskyrimai, šakojimai, ciklai, įvestis/išvestis, paprogramės ir pan., pavyzdžiai):  $\approx 3$  psl.;
- LP priemonės (procesai, semaforai, monitoriai, pranešimų perdavimo priemonės, pavyzdžiai):  $\approx 5$  psl.;
- lygiagrečių programų pavyzdžiai:  $\approx 3$  psl.;
- išvados ir literatūra:  $\approx 2$  psl.



# Individualus darbas-referatas 2: LP aplinkos ir kitos pagalbinės priemonės (.pdf)

- titulinis lapas;
- priemonės paskirtis, vystymasis ir paplitimas:  $\approx 3$  psl.;
- priemonės aprašymas (naudojimas, vykdomos funkcijos):  $\approx 7$  psl.;
- panaudojimo pavyzdžiai:  $\approx 3$  psl.;
- išvados ir literatūra:  $\approx 2$  psl.

# Vietoj reziumė

*There are 3 concepts to grasp on your way to becoming a decent developer.*

*The 1st concept is **recursion**, which is a great filter but probably most CS students can make ends with it.*

*The 2nd concept is **pointers** (and the underlying memory model), which seems to become more and more extinct in modern programmers, with the high-level abstractions we have today.*

*Finally, the 3rd concept is **parallelism, concurrency and multi-threading** — and while there are countless programmers I know who can recite a recursive Fibonacci algorithm, there is perhaps only half a dozen who can wisely opine on concurrency-related issues.*

Sasha Goldshtein, Sela Group, Israel.