

Inžinerinis projektas

Vertinimo tvarka

- Ataskaita — maks. 5 balai
- Programos veikimas ir darbų tarp gijų paskirstymas — maks. 2,5 balo
- Atsakymai į klausimus — maks. 2,5 balo

Ataskaitos struktūra

1. Titulis lapas ir užduotis.
2. Užduoties analizė ir sprendimo metodas.
3. Programos aprašymas (trumpi klasių, duomenų struktūrų ir metodų aprašai).
4. Programos pagrindinių dalių tekstai su komentarais.
5. Testavimas ir programos instaliavimo bei vykdymo instrukcija (kaip patikrinta, ar gautami teisingi rezultatai, ir kaip pasileisti programą).
6. Vykdymo laiko kitimo tyrimas (**Svarbiausias dalis**; diagramos, jų paaiškinimai ir įžvalgos).
7. Išvados ir literatūra.

Vykdymo laiko kitimo tyrimas

- Apie 10 skirtingos apimties duomenų rinkinių.
- Kiekvienam duomenų rinkiniui nustatyti vykdymo laiko priklausomybę nuo procesų skaičiaus.
- Nubraižyti diagramas, kurios parodytų vykdymo laiko priklausomybę nuo procesų skaičiaus kiekvienam pasirinktų duomenų rinkiniui.
- Vykdymo laiko tyrimą kartoti keletą kartų ir pateikti vykdymo laikų vidurkius.

Pateikimo ir gynimo tvarka

- Projekto pristatymo laikas pateiktas Moodle, gintis reikia paskirtą savaitę.
- Jei naudojamos ne siūlomos priemonės, jos turi būti iš anksto suderintos su lab. darbų dėstytoju.
- Pateikiamos ataskaitos formatais — PDF.
- Ataskaitą, programą ir duomenų failus (1 variantą) įkelti į Moodle prieš projekto gynimą.
- Neatsiskaičius projekto laiku pakartotinai gintis galima 16 savaitę.

Užduotis

- Tiems, kas turi skaitinių metodų modulį, **rekomenduojama** išlygiagretinti antrojo namų darbo optimizavimo uždavinį.
- Galima susigalvoti savo užduotį, bet reikia iš anksto suderinti su laboratorinių darbų dėstytoju.

Priemonės Tikslas — išbandyti duomenų lygiagretumo programavimo priemonės, kurios gijas valdo automatiškai, bet paleidžiamų gijų kiekį galima nurodyti kokį norima. Galima rinktis savo priemonės (turi būti suderinta su lab. darbų dėstytoju) arba iš siūlomų:

- C++ su OpenMP `parallel for`
- C++ su MPI `Scatter / Gather`
- C++ su CUDA
- Haskell `rpar` ir `parListChunk` ar kt. strategija
- Python `multiprocessing.Pool`
- Ruby su `parallel gem`
- Java / Kotlin `parallel stream`
- C# `parallel LINQ`
- Scala `parallel collections`
- F# `PSeq`
- Rust `rayon`

Referatas

Vertinimo tvarka

- Ataskaita — maks. 7,5 balai
- Atsakymai į klausimus — maks. 2,5 balo

Tematika

- Laboratorinių darbų metu naudojamų kalbų (C++, C#, Java ir Go) analizuoti negalima.
- Pasirinktą programavimo kalbą galima lyginti su laboratorinių darbų metu naudojamomis kalbomis.

Referato struktūra

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

Lygiagrečiojo programavimo priemonių apžvalga

Reikia apžvelgti visas kalbos ir jos standartinės bibliotekos teikiamas lygiagretumo priemones, palyginti su kitomis programavimo kalbomis, trumpai apžvelgti pagrindines esamas pasirinktos kalbos lygiagretumo bibliotekas.

Pateikimo ir gynimo tvarka

- Referato pristatymo laikas pateiktas Moodle, gintis reikia paskirtą savaitę.
- Pateikiamos ataskaitos formatas — PDF.
- Atskaitą, programą ir duomenų failus (1 variantą) įkelti į Moodle prieš referato gynimą.
- Neatsiskaičius referato laiku pakartotinai gintis galima 16 savaitę.