



SD(M) pavadinimas	<i>Course title</i>
Lygiagretusis programavimas	<i>Parallel Programming</i>

SD(M) priklausomybė studijų pakopai
Course subjection to study level

Studijos: <i>Studies:</i>	B – Pirmosios pakopos First cycle
-------------------------------------	--------------------------------------

SD(M) priklausomybė studijų programai
Course subjection to programme

SD(M) priklausomybė studijų krypčių ir krypčių grupei

The list of study fields and groups of fields

SD(M) priklausomybė dalykų grupei * <i>Course subjection to group</i>	1 – studijų dalyko Course	<table border="1"> <tr> <td>Studijų krypčių grupės kodas <i>Code of the group of study fields</i></td> <td>Studijų krypties kodas <i>Code of the study field</i></td> </tr> <tr> <td>B</td> <td>B01</td> </tr> </table>	Studijų krypčių grupės kodas <i>Code of the group of study fields</i>	Studijų krypties kodas <i>Code of the study field</i>	B	B01
Studijų krypčių grupės kodas <i>Code of the group of study fields</i>	Studijų krypties kodas <i>Code of the study field</i>					
B	B01					
SD(M) priklausomybė programos daliai ** <i>Course subjection to part of the programme</i>	B – Studijų krypties dalykų dalis Part of Study area Subjects					
Struktūrinė SD priklausomybė *** <i>Course structural subjection</i>	K – katedros Department					

*) **Grupė:** *) 1 - studijų dalyko; 2 - praktikos; 3 - baigiamojo darbo ar projekto; 4 - baigiamojo egzamino; 5 - tiriamojo darbo; 6 - profesinio testavimo; 7 - kitas.

**) A - Bendrųjų universitetinių studijų; B - Studijų krypties; C - Specializacijos.

***) U - universiteto; F - fakulteto; K - katedros.

*) **Group:** *) 1 - Course; 2 - Practice; 3 - Final Work or Project; 4 - Final Examination; 5 - Research Work; 6 - Professional Testing; 7 - Other.

**) A - General; B - Field; C - Specialization.

***) U - University; F - Faculty; K - Department.

SD(M) kodas
Course number

SD(M) kreditai
Course volume in credits

SD(M) Atsiskaitymo forma
Course assessment

Fakultetas <i>Faculty</i>	Katedra <i>Department</i>	Pakopa * <i>Study cycle</i>	Modulio Nr. <i>Number</i>
F M	M M	B	26601

Iš viso: <i>Total:</i>	Iš jų: KD, KS, KP, PR <i>There out:</i>
3	0

I, E1, E2, E, BE, BD, TD, A	KD, KS, KP, PR
E	-

*) B - pirmoji pakopa; A - vientisosios studijos; M - antroji pakopa.

*) B - first cycle studies; A - integrated studies; M - second cycle studies.

SD(M) valandų paskirstymas pagal studijų formas ir būdus

Distribution of course hours by study forms and ways

Studijų forma <i>Study form</i>	Valandos <i>Hours</i>								Kontaktinių <i>Contact</i>
	Kodas <i>Code</i>	Studijų būdas * <i>Study way</i>	Paskaitoms <i>Lectures</i>	Lab. darbams <i>Laboratory works</i>	Pratyboms <i>Practical works</i>	Konsultacijoms <i>Consultation</i>	Sav. darbui <i>Independent work</i>	Iš viso <i>Total</i>	
Nuolatinės studijos <i>Full-time studies</i>	NL	S	30	15	0	2	33	80	47

*) Studijų būdas: S - semestrais; M - moduliais; C - ciklais; T - nuotolinis; NI - neakivaizdinis intensyvusis.

*) Study process forms: S - semesters; M - modules; C - periods; T - distance; NI - part-time.

SD(M) ANOTACIJA

Šis kursas supažindina su lygiagrečiojo programavimo pagrindais, apimančiais tiek lygiagrečiųjų skaičiavimų technologinius pamatus, tiek pagrindinius naudojamus lygiagrečiojo programavimo modelius ir įrankius. Iš pradžių studentai nagrinėja lygiagrečiųjų kompiuterių architektūras ir jų klasifikacijas - įskaitant Flynn klasifikaciją, hierarchinę atmintį, instrukcijų lygio lygiagretumą šiuolaikiniuose procesoriuose, vektorinius procesorius, bendros atminties (UMA, NUMA) ir paskirstytos atminties lygiagrečiuosius kompiuterius bei naudojamus sujungimo tinklus. Toliau kurse nagrinėjamas bendros atminties programavimas naudojant gijų bibliotekas ir OpenMP standartą, paskirstytos atminties programavimas naudojant MPI biblioteką bei bendrosios paskirties skaičiavimai grafiniuose procesoriuose (GPGPU) CUDA platformoje su jos

heterogeninių skaičiavimų modelių. Kursas baigiamas lygiagrečiųjų algoritmų analize, apimant spartinimo ir efektyvumo vertinimą, Amdahlo dėsnį, išplečiamumo analizę, įskaitant pagrindinius lygiagrečiųjų algoritmų sudarymo būdus, duomenų ir funkcinę lygiagretumą. Baigę kursą studentai geba kurti lygiagrečiąsias programas naudodami gijų bibliotekas, OpenMP, MPI, CUDA ir analizuoti jų spartinimą, našumą bei išplečiamumą.

Studentai numatytu tvarkaraštyje metu privalo atlikti ne mažiau kaip 80 proc. laboratorinių darbų.

ANNOTATION OF COURSE

This course introduces the fundamentals of parallel programming, covering both hardware foundations and the main parallel programming models and tools. Students study the architecture of parallel and distributed computing systems - including Flynn's taxonomy, the von Neumann bottleneck, hierarchical memory, pipelining, vector processors, shared-memory (UMA, NUMA) and distributed-memory parallel computers, and their interconnection networks. The course then addresses shared-memory programming with thread libraries and the OpenMP standard, distributed-memory programming with MPI library, and general-purpose GPU computing on the CUDA platform with its heterogeneous model, kernels, threads, and SIMT principle. It concludes with the analysis and design of parallel algorithms, covering speedup, efficiency, Amdahl's law, scalability, decomposition techniques, and data and functional parallelism. Upon completion, students are able to develop parallel programs using thread libraries, OpenMP, MPI, and CUDA, and to analyze their performance and scalability.

Students must attend at least 80% of the time scheduled laboratory works.

SD(M) TIKSLAS

Dalyko tikslas - supažindinti studentus su lygiagrečiais algoritmais, jų sudarymo ir jų realizavimo principais bei lygiagretaus programavimo technologijomis: daugelio gijų, daugelio procesorių, paskirstytų sistemų.

AIM OF COURSE

The course aims to provide knowledge of modern parallel computing technologies, main methods for the design of parallel algorithms and parallel programming tools.

Studento pasiekimų vertinimo formulė

Assessments methods of students formula

Pagrindinė literatūra (ne daugiau kaip 5 šaltiniai):

Main references (not more than 5 references)

Eil. Nr. <i>No.</i>	Leidinio autoriai ir pavadinimas (elektroninių leidinių ir žiniatinklio adreso) <i>Authors and title (site address in case of e-publication)</i>
1.	R. Čiegis. Lygiagretieji algoritmai ir tinklinės technologijos. Vilnius, Technika. 2005.
2.	Bertil Schmidt, Jorge Gonzalez-Dominguez, Christian Hundt, Moritz Schlarb. Parallel Programming: Concepts and Practice. Morgan Kaufmann, 2017.
3.	J. J. Dongarra, I. Foster, G. C. Fox, W. Gropp, K. Kennedy, L. Torczon, A. White. Source Book of Parallel Computing. Morgan Kaufmann, 2003.
4.	D. Storti, M. Yurtoglu. CUDA for Engineers: An Introduction to High-Performance Parallel Computing. Addison-Wesley Professional, 1st Edition, 2015.
5.	B. Barney. Introduction to Parallel Computing. Lawrence Livermore National Laboratory. https://computing.llnl.gov/tutorials/parallel_comp/

*) Kortelės pildymo metu

*) *At the form filling moment*

Papildoma literatūra (ne daugiau kaip 10 šaltinių):

Additional references (not more than 10 references)

Eil. Nr. <i>No.</i>	Leidinio autoriai ir pavadinimas (elektroninių leidinių ir žiniatinklio adreso) <i>Authors and title (site address in case of e-publication)</i>
1.	M. Ben-Ari. Principles of Concurrent and Distributed Programming. Addison-Wesley Publishing Company, 2006.
2.	V.Kumar, A.Grama, A.Gupta, G.Karypis. Introduction to Parallel Computing: Design and Analysis of Algorithms. The Benjamin/Cummings Publishing Company, 2003.

*) Kortelės pildymo metu

*) *At the form filling moment*

Reikalingi IT resursai * (nurodyti 1-3 alternatyvas, pageidautina, kad bent 1 būtų nemokama)

Required IT Resources

Eil. Nr. No.	Programinės įrangos pavadinimas, gamintojas <i>Name of the software, manufacturer</i>	Licencijos tipas (pagal įsigijimo būdą) <i>License type</i>
1	C++	Mokama, akademinė Paid, academic
2	C	Mokama, akademinė Paid, academic

*) Pildoma, jei tokie resursai reikalingi. Stulpelyje Licencijos tipas pasirenkamas iš sąrašo:

Mokama, akademinė

Mokama, komercinė

Nemokama

*) Should be completed if such reassures are needed. License type - select from the list:

Paid, academic

Paid, commercial

Savarankiško darbo turinys

Content of individual work

Užduoties pavadinimas <i>Assignment title</i>	Sav. darbo apimtis vienai užduočiai <i>Amount of hours of independent work for a single task</i>					Užduočių skaičius <i>Number of tasks</i>					Iš viso valandų <i>Total hours</i>					Įvertinimo dalis % <i>Part of Evaluation %</i>				
	Rekomenduojamos val. <i>Recommended hours</i>	Skirta val. <i>Separated hours</i>				NL (T)	NL (S)	NL (Sav.)	I(S)	I(T)	NL(T)	NL(S)	NL (Sav.)	I(S)	I(T)	NL(T)	NL(S)	NL (Sav.)	I(S)	I(T)
		NL(T)	NL(S)	NL (Sav.)	I(S)															
Laboratorinis darbas <i>Laboratory work</i>	2-12		4				1					4					10			
Namų darbas <i>Home work</i>	4-27		4				2					8					20			
Pasirengimas atsiskaitymui <i>Preparation for evaluation</i>	10-60		12				1					12								
Kolokviumas <i>Intermediate examination</i>	8-27		9				1					9					20			
Iš viso: Total:											33									

*) Papildomas laukas pildomas tik tada, kada taikomas SD(M) kortelėje nenurodytas studijų būdas: M - moduliai; C - ciklais; T - nuotolinis

*) Must be used in case study way does not fall into standard category: M - modules; C - periods; T - distance

Savarankiško darbo grafikas

Individual work schedule

Užduoties tipas <i>Task type</i>	Užduoties pateikimo(*) ir atsiskaitymo(+) savaitė <i>Week of Assignment setting (*) and assessment(+)</i>																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Nuolatinės studijos (S) <i>Full-time studies</i>																				
Laboratorinis darbas <i>Laboratory work</i>	*	1																		
Kolokviumas <i>Intermediate examination</i>	*				1															
Namų darbas <i>Home work</i>	*				1				2											
	+								1							2				

Laboratorinių darbų sąrašas

List of the Course laboratory work

Temos pavadinimas <i>Topic title</i>	Valandų skaičius <i>Number of hours</i>				
	NL(T)	NL(S)	NL(Sav.)	I(S)	I(T)
1. Spartinančios (angl. "cache") atminties panaudojimo efektyvumo tyrimas matricos ir vektorių sandaugos algoritmu pagalba. Kompiliatorių optimizacijos tyrimas. BLAS bibliotekų panaudojimas. Darbas lygiagreto klasterio Linux terpėje. <i>Investigation of cache memory performance using matrix-vector multiplication. Compiler optimization. Application of BLAS libraries. Job preparation and submission on Linux cluster.</i>		3			
2. Lygiagrečiųjų algoritmu realizavimas OpenMP pagalba. Tiesinės algebras uždavinių sprendimas. Skaičiavimo eksperimentų atlikimas. Spartinimo ir efektyvumo koeficientų vertinimas ir analizė. <i>Parallel solution of linear algebra problems. Implementation of parallel algorithms with OpenMP. Speed-up and efficiency analysis.</i>		4			
3. Lygiagrečiųjų algoritmu realizavimas MPI bibliotekos pagalba. Tiesinės algebras uždavinių sprendimas. Skaičiavimo eksperimentų atlikimas. Spartinimo ir efektyvumo koeficientų vertinimas ir analizė. <i>Parallel solution of linear algebra problems. Implementation of parallel algorithms with MPI. Speed-up and efficiency analysis.</i>		4			
4. Lygiagrečiųjų algoritmu realizavimas CUDA platformos pagalba. Efektyvus atminties panaudojimas. Optimizavimo technikų taikymas. CUDA bibliotekų panaudojimas taikomiesiems uždaviniams. <i>Implementation of parallel algorithms on CUDA platform. Efficient memory usage and optimization techniques. Employment of CUDA libraries for applied problems.</i>		4			
Iš viso: <i>Total:</i>		15			

Paskaitų temų sąrašas

List of the Course lecture topics

Temos pavadinimas <i>Topic title</i>	Valandų skaičius <i>Number of hours</i>				
	NL(T)	NL(S)	NL(Sav.)	I(S)	I(T)
1. Lygiagrečiųjų kompiuterių ir paskirstytų sistemų architektūra. Flynn klasifikacija (kompiuterių architektūros taksonomija). Von Neumanno "butelio kakliukas", hierarchinės atminties sistemos, spartinančioji atmintinė. Konvejeriai procesorių architektūroje. Vektoriniai procesoriai. <i>Architecture of parallel computers and distributed systems. Flynn's classification (computer architecture taxonomy). Von Neumann bottleneck, hierarchical memory systems, cache memory. Pipelining in processor architecture. Vector processors.</i>		2			
2. Bendros atminties lygiagretieji kompiuteriai. UMA ir NUMA architektūros tipai. Paskirstytos atminties lygiagretieji kompiuteriai. Pagrindiniai naudojamų tinklų tipai ir jų charakteristikos. <i>Shared-memory parallel computers. UMA and NUMA architecture types. Distributed-memory parallel computers. Main types of interconnection networks and their key characteristics.</i>		2			
3. Bendrosios atminties lygiagretusis programavimas. Gijų programavimo bibliotekos. Gijų sukūrimas, valdymas ir sinchronizavimas. Lokalieji ir bendrieji kintamieji. Lenktynių konfliktas, duomenų lenktynės. <i>Shared-memory parallel programming. Thread programming libraries. Thread creation, management, and synchronization. Local and shared variables. Race conditions and data races.</i>		4			
4. OpenMP programavimo standartas: pagrindiniai principai ir konstrukcijos. OpenMP duomenų tipai, užduočių paskirstymas, sinchronizavimo konstrukcijos. <i>OpenMP programming standard: fundamental principles and constructs. OpenMP data-sharing attributes, workload distribution, and synchronization constructs.</i>		4			
5. Paskirstytosios atminties lygiagretusis programavimas. MPI programavimo biblioteka: pagrindiniai principai ir koncepcijos. Pagrindinės MPI funkcijos. <i>Distributed-memory parallel programming. MPI programming library: fundamental principles and concepts. Basic MPI functions.</i>		2			
6. MPI duomenų siuntimo būdai: sinchroninis, buferinis, blokuotas ir neblokuotas. Kolektyvinės MPI duomenų persiuntimo operacijos. <i>MPI data transfer methods: synchronous, buffered, blocking, and non-blocking. Collective MPI data transfer operations.</i>		4			
7. Grafinių procesorių panaudojimas bendros paskirties skaičiavimuose. GPU architektūros. GPGPU programavimo įrankiai. <i>General-purpose computing on graphics processing units (GPGPU). GPU architectures. GPGPU programming tools.</i>		2			
8. CUDA platforma. CUDA programavimo modelis: heterogeninių skaičiavimų principai, kernelis, gija, gijų blokas. SIMT principas. CUDA bibliotekos. <i>CUDA platform. CUDA programming model: principles of heterogeneous computing, kernel, thread, thread block. SIMT principle. CUDA libraries</i>		4			

Paskaitų temų sąrašas
List of the Course lecture topics

Temos pavadinimas <i>Topic title</i>	Valandų skaičius <i>Number of hours</i>				
	NL(T)	NL(S)	NL(Sav.)	I(S)	I(T)
9. Lygiagrečiųjų algoritmų analizė. Algoritmo spartinimo ir efektyvumo vertinimas. Amdahlo dėsnis, jo išvados ir apibendrinimai. Lygiagrečiojo algoritmo išplečiamumas. <i>Analysis of parallel algorithms. Evaluation of algorithm speedup and efficiency. Amdahl's law, its implications and generalizations. Scalability of parallel algorithms.</i>		4			
10. Pagrindiniai lygiagrečių algoritmų sudarymo būdai. Duomenų lygiagretumas. Funkcinis lygiagretumas. Ganto schemas. <i>Main principals for designing parallel algorithms. Decomposition and mapping techniques. Data parallelism. Functional parallelism. Gantt charts.</i>		2			
Iš viso: <i>Total:</i>		30			

*) Papildomas laukas pildomas tik tada, kada taikomas SD(M) kortelėje nenurodytas studijų būdas: M - moduliai; C - ciklais; T - nuotoliniu

*) *Must be used in case study way does not fall into standard category: M - modules; C - periods; T - distance*

Fundamentinių mokslų fakulteto Programų inžinerijos (612I30003) 2016-07-01 programos studijų rezultatų sąsajos su SDM rezultatais bei studijų ir studentų pasiekimų vertinimo metodais

Links of the Software Engineering (612I30003) of the Faculty of Fundamental Sciences with the course unit and evaluation methods of students achievements

Programos studijų rezultatai <i>Study programme outcomes</i>	SD(M) rezultatai <i>Course results</i>	Studijų metodai <i>Methods of studies</i>	Studento pasiekimų vertinimo metodai <i>Evaluation methods of student achievements</i>	Studentų pasiekimų vertinimo kriterijai pagal lygmenis <i>Assessments criteria of students achievements by Assessment levels</i>
Z2. Žinios ir galės paaiškinti algoritmų ir duomenų struktūras, programavimo paradigmas, kalbų, karkasų ir technologijų bei jų vystymosi tendencijas, programų kūrimo įrankius ir kodavimo standartus bei mokės tai paaiškinti ir taikyti. Z2. Consistent knowledge of algorithms and data structures, programming paradigms, languages and frameworks, programming technologies and their evolution tendencies, coding standards and software development tools, competency in its explanation and application.	Studentai įgyja žinias apie lygiagrečiųjų algoritmų sudarymą, lygiagretaus programavimo paradigmas ir įrankius: OpenMP, MPI, CUDA. Students acquire knowledge on the design of parallel algorithms, paradigms and tools of parallel programming: OpenMP, MPI, CUDA.	Teorinės paskaitos, laboratoriniai užsiėmimai, literatūros studijos, savarankiškas darbas. Theoretical lectures, laboratory works, study of literature, independent work.	Laboratorinis darbas, namų darbai, tarpinis egzaminas, egzaminas. Laboratory work, home works, intermediate exam, final exam.	Slenkstinis (5-6): studentas žino pagrindines dalyko sąvokas ir moka jomis naudotis sprendžiamas paprasčiausias užduotis. Tipinis (7-8): studentas žino pagrindines sąvokas, gali jas paaiškinti ir teorinius faktus geba tinkamai taikyti atlikdamas praktinių ir teorinių užduočių analizę, skaičiavimams taiko kompiuterines programas. Puikusias (9-10): studentas yra ne tik pasiekęs tipinį lygmenį, bet įgytas žinias geba taikyti naujose situacijose. Threshold (5-6): The student knows the basic concepts of the subject and is able to apply them solving simple problems. Typical (7-8): The student knows the basic concepts, is able to explain them and to apply theoretical facts correctly analyzing practical and theoretical problems, applies computer programs for calculations. Excellent (9-10): The student not only reached the typical level, but is able to apply the

				acquired knowledge in new situations.
<p>SG3. Gebės taikyti skirtingas programavimo paradigmas, algoritmus, programinius karkasus, naudoti programų kūrimo įrankius, aplinkas ir priemones, testuoti programinę įrangą, kurti testavimo scenarijus pagal funkcinius ir nefunkcinius reikalavimus, vertinti testavimo rezultatus ir pateikti išvadas.</p> <p>SG3. Ability to apply different programming paradigms, algorithms, programming frameworks, to use relevant software development tools, environments and approaches, to apply software testing methods and develop test-cases according to functional and non-functional requirements, to evaluate testing results and make conclusions.</p>	<p>Studentai gebės taikyti skirtingus lygiagrečiųjų algoritmų sudarymo metodus, lygiagretaus programavimo paradigmas ir įrankius, testuoti ir vertinti lygiagrečiųjų algoritmų ir jų implementacijų efektyvumą.</p> <p>Students will be able to apply different methods for the design of parallel algorithms, use different paradigms and tools of parallel programming, test and evaluate the efficiency of parallel algorithms and their implementations.</p>	<p>Teorinės paskaitos, laboratoriniai užsiėmimai, literatūros studijos, savarankiškas darbas.</p> <p>Theoretical lectures, laboratory works, study of literature, independent work.</p>	<p>Laboratorinis darbas, namų darbai, tarpinis egzaminas, egzaminas.</p> <p>Laboratory work, home works, intermediate exam, final exam.</p>	<p>Slenkstinis (5-6): studentas žino pagrindines dalyko sąvokas ir moka jomis naudotis sprendamas paprasčiausias užduotis.</p> <p>Tipinis (7-8): studentas žino pagrindines sąvokas, gali jas paaiškinti ir teorinius faktus geba tinkamai taikyti atlikdamas praktinių ir teorinių užduočių analizę, skaičiavimams taiko kompiuterines programas.</p> <p>Puikūs (9-10): studentas yra ne tik pasiekęs tipinį lygmenį, bet įgytas žinias geba taikyti naujose situacijose.</p> <p>Threshold (5-6): The student knows the basic concepts of the subject and is able to apply them solving simple problems.</p> <p>Typical (7-8): The student knows the basic concepts, is able to explain them and to apply theoretical facts correctly analyzing practical and theoretical problems, applies computer programs for calculations.</p> <p>Excellent (9-10): The student not only reached the typical level, but is able to apply the acquired knowledge in new situations.</p>

SD(M) sudarytojas (-ai) (parašas, vardas ir pavardė)

Course compiled by (full name, signature)

Vadimas Starikovičius

Katedros vedėjas (parašas, vardas ir pavardė)

Head of Department (full name, signature)

Raimondas Čiegis

SD(M) atestuojamas <i>The Course is certified</i>		
SD(M), skirtas studijų programai: <i>The Course for the programme of studies:</i>	Programų inžinerija Software Engineering	
SD(M) atestacija galioja: <i>Course certification is valid:</i>	nuo <i>from</i>	iki <i>till</i>

SD(M) atestavo <i>the Course certified by</i>	Fundamentinių mokslų fakulteto studijų komitetas Faculty of Fundamental Sciences Study Committee		
Fakulteto studijų komiteto pirmininkas (vardas ir pavardė, parašas) <i>Chairman of the Studies committee (full name, signature)</i>		Data <i>Date</i>	