



MODULIO APRAŠAS

Modulio pavadinimas	Kodas
Kompiuterių architektūra	

Dėstytojas(-ai/-os)	Padalinys
Koordinuojantis: prof. dr. Saulius Gražulis Kitas (-i/-os): -	Informatikos katedra Matematikos ir informatikos fakultetas Vilniaus universitetas

Studijų pakopa	Dalyko tipas
pirmoji	Privalomasis

Igyvendinimo forma	Vykdyto laikotarpis	Vykdyto kalbos
Auditorinė/nuotolinė	3 semestras	Lietuvių, anglų

Reikalavimai studijuojančiajam
Išankstiniai reikalavimai: -

Modulio apimtis kreditais	Visas studento darbo krūvis	Kontaktinio darbo valandos	Savarankiško darbo valandos
5	134	66	68

Modulio tikslas: studijų programos ugdomos kompetencijos		
<p>Modulio tikslas: suformuoti realaus programų vykdymo supratimą kaip iteracinį duomenų būsenos transformavimą kompiuterio komandų pagalba, įvaldyti mašininio lygmens sąvokų sistemą, išmokti skaityti ir rašyti mašininio lygmens programas.</p> <p>Bendrosios kompetencijos:</p> <ul style="list-style-type: none"> Analizuoti ir sisteminti informaciją (BK1). Žinias pritaikyti praktikoje (BK2). Organizuoti ir planuoti darbus, dirbti individualiai ir grupėje (BK3). <p>Dalykinės kompetencijos:</p> <ul style="list-style-type: none"> Programavimo (DK6). Sistemų architektūros (DK7). 		
Modulyje ugdomi studentų gebėjimai:	Mokymo ir mokymosi metodai	Vertinimo metodai
Gebės laisvai ir tikslingai operuoti kompiuterių architektūros sąvokomis	Mokymo metodai: <ul style="list-style-type: none"> Paskaitos; Laboratoriniai darbai. 	Egzaminavimas raštu. Laboratorinių darbų atsiskaitymas. Apklausa.
Supras kompiuterinių sistemų diagnostinius pranešimus mašininiais terminais	Mokymosi metodai: <ul style="list-style-type: none"> Faktinių žinių kaupimas; Žinių sintezė – faktinių žinių apibendrinimas, abstrahavimas ir agregavimas; Žinių analizė – naujų žinių derinimas su agreguotomis žiniomis, verifikavimas ir koregavimas; Agreguotų ir verifikuotų žinių taikymas. 	Ataskaita ir jos pristatymas. Vertinimo kriterijai: <ul style="list-style-type: none"> gebėjimas spręsti praktines užduotis; gebėjimas sudaryti, derinti, stebėti vykdymą, modifikuoti ir paaiškinti programas assemblerio kalba; gebėjimas paaiškinti kompiuterio ir procesoriaus darbo principus loginių schemų lygmenyje;
supras kompiuterio architektūros įtaką programų greitaveikai ir teisingumui		
Turės vaizdinius, reikiamus programavimo kalbų įsisavinimui		

Temos	Kontaktinio darbo valandos						Savarankiškų studijų laikas ir užduotys		
	Paskaitos	Konsultacijos	Seminarai	Pratybos	Laboratoriniai darbai (LD)	Konsultavimas LD metu	Visas kontaktinis darbas	Savarankiškas darbas	Užduotys
1. Bendra kompiuterių architektūros apžvalga. Perjungiančios elektrinės schemos.	2				2		4	4	
2. Pagrindiniai kompiuterių schemų elementai. Loginės schemos. Skaičiavimo sistemos ir skaičių atvaizdavimas. Posto teorema.	2				2		4	4	
3. Aritmetinių operacijų įgyvendinimas. Pozicinės skaičiavimo sistemos ir skaičių atvaizdavimas.	2				2		4	4	
4. Skaitmeninės schemos su būseną. Atmintis	2				2		4	4	
5. Procesoriaus duomenų traktas ir jo valdymas. Baigtiniai automatai. Mikroprogramavimas.	2				2		4	4	
6. Informacijos atvaizdavimas kompiuteriuose. Įvairūs sveikų ir racionalių skaičių atvaizdavimo metodai. Simbolių eilučių atvaizdavimas. koduotės. Unikodas.	2				2		4	4	
7. Slankaus kabelio skaičiai.	2				2		4	4	
8. Kintamo dydžio duomenų atvaizdavimas. Įvairūs sudėtingesni skaičių atvaizdavimo būdai. Padidinto tikslumo aritmetika. CISC ir RISC komandų skaičiams ir simboliams apdoroti pavyzdžiai.	2				2		4	4	
9. Aparatinė procesoriaus realizacija. Konvejeriai. Procesoriaus valdymo automatas. Skirtingi architektūrų tipai (steko mašinos, akumuliatorinės mašinos, arch. atmintis-atmintis ir registras-registras), CISC ir RISC architektūrų principai. Nulio, vieno, dviejų ir trijų adresų komandos.	2				2		4	4	
10. RISC-V ISA	2				2		4	4	
11. Programavimas assemblerio kalba. Komandų mnemonikos, operandai, adresavimo režimai, sekcijos, žymės, makrokomandos. Kompiliavimas iš aukšto lygio programavimo kalbų (pvz. C).	2				2		4	4	
12. Konvejerinės architektūros. Tarpinė atmintis. RISC-V emuliacijos. Programų pavyzdžiai ir analizė.	2				2		4	4	
13. CISC procesoriai. x86 architektūros pavyzdys.	2				2		4	4	
14. Virtuali atmintis. Puslapiavimas. Segmentavimas. Atminties apsauga.	2				2		4	4	
15. Mikrovaldikliai (pvz. AVR). Pertraukimai. Periferiniai įrenginiai (taimeriai, analoginiai-skaitmeniniai keitikliai).	2				2		4	4	
16. Egzotiškos, futuristinės, nestandartinės architektūros: DNN, žymių architektūros (angl. <i>tagged architectures</i>). FPGA, FORTH mašinos, ląstelių matrica. Aparatūros aprašymo kalbos.	2				2		4	4	
Pasiruošimas egzaminui ir jo laikymas							2	4	
Iš viso	32				32		66	68	

Vertinimo strategija	Svoris %	Atsiskaitymo laikas	Vertinimo kriterijai
Apklausa prieš paskaitas	10	10 min. prieš paskaitą ar praktikos darbą.	4-klausimų apklausa iš keleto praeitų paskaitų temų (1 ir antro lygio klausimai pagal Blūmo (Bloom) klasifikaciją) naudojant elektronines mokymo priemones (Moodle, Open edX ar pan).
Tarpinis kontrolinis	15	semestro vidurys	Apytikriai 30 klausimų apklausa iš visų praeitų dalykų (1 iki 9 lygio klausimai pagal Bloom klasifikaciją) naudojant elektronines mokymo priemones (Moodle, Open edX ar pan).
Pratybų (laboratoriniai) darbai	50	pagal pratybų vadovų paskelbtą grafiką	Praktikos darbai atliekami pagal pratybų vadovų nurodytas užduotis ir atsiskaitomi bei vertinami pagal kiekvieno pratybų vadovo nurodytą grafiką bei kriterijus.
Savarankiškas konkrečios architektūros studijavimas	10	semestro pabaiga	Studentai pateikia maždaug 4 psl. (A4, 9pt) techninį aprašymą apie jiems paskirtą ir savarankiškai išnagrinėtą kompiuterių architektūrą, arba 5 min. žodinį pristatymą su skaidrėmis (PDF formatu). Žodinis pristatymas galimas studentams, pasiekusiems gerų rezultatų kurso metu ir gali būti užskaitomas kaip egzaminas (sąlygas žr. punkte „Egzaminas“).
Egzaminas	15	semestro pabaiga	<p>Apytikriai 30 klausimų apklausa iš viso kurso dalykų (1 iki 9 lygio klausimai pagal Bloom klasifikaciją) naudojant elektronines mokymo priemones (Moodle, Open edX ar pan).</p> <p>Kad studentai būtų prileisti prie egzamino, jie turi:</p> <ol style="list-style-type: none"> 1. Atlikti bent vieną praktikos darbą ir surinkti teigiamą (didesnį už nulį) pratybų balą; 2. Sukaupti suminį balą už darbą per semestrą (iš praktikos darbų, tarpinio kontrolinio, darbo paskaitose, galimai kitų dėstytojų paskirtų užduočių), kad, parašius egzamino kontrolinį, būtų įmanoma pasiekti patenkinamo pažymio balą (t. y. balą, užtikrinantį bent pažymį „5“); 3. Ypač gerai pasirodžiusiems semestro metu studentams gali būti leidžiama laikyti išankstinį egzaminą, padarant žodinį pranešimą dėstytojo paskirtu laiku užsiėmimų metu. Norint laikyti išankstinį egzaminą, būtina: <ol style="list-style-type: none"> 1. Surinkti bent bent 60% semestro metu galimų gauti teorijos balų (pvz. bent 150 iš dabar prieinamų 250 balų); 2. Atsiskaityti laiku visus praktikos darbus; 3. Turėti teigiamas praktikos vadovo rekomendacijas. <p>Jei norinčių laikyti egzaminą iš anksto studentų yra daugiau, negu leidžia turimas užsiėmimų laikas, pirmenybė suteikiama studentams, surinkusiems daugiau balų.</p> <p>Egzamino kontrolinis yra būtinas visiems nepriklausomai nuo surinkto balų skaičiaus, išskyrus tuos studentus, kurie gavo leidimą laikyti egzaminą iš anksto ir kuriems užskaitytas kaip egzaminas žodinis savarankiško darbo pristatymas. Studentams, neatvykusiems į egzaminą, žiniaraštyje bus žymima „neatvyko“. Egzamine būtina</p>

			surinkti bent 50% egzamino balo.
Viso	100		Galutinis pažymys gaunamas susumavus už visas veiklas surinktus balus, padalinant sumą iš 100 ir su apvalinant iki <i>didesnio</i> sveiko skaičiaus (t.y. 0.001 apvalinama link 1; 9.1 apvalinama iki 10).

Autorius	Leidimo metai	Pavadinimas	Periodinio leidinio Nr. arba tomas	Leidimo vieta ir leidykla ar internetinė nuoroda
Privalomoji literatūra				
Andrew S. Tanenbaum	2005	Structured computer organization		Prentice Hall PTR, Fifth Edition
D. A. Patterson and J. L. Hennessy	2017	Computer Organization and Design: The Hardware/Software Interface. RISC-V edition.		Elsevier
A. Waterman, Y. Lee, D. Patterson, and K. Asanović	2011	The RISC-V instruction set manual. Volume I: base user-level ISA. Version 1.0.	Vol. 1, ver. 1.0	https://inst.eecs.berkeley.edu/~cs250/fa11/handouts/riscv-spec.pdf
Papildoma literatūra				
Antanas Mitašiūnas	2016	Computer architecture. Teaching book (in Lithuanian Kompiuterių architektūra)		Vilnius, 126 p. http://www.mif.vu.lt/katedros/cs/Asmen/Kompiuteriu%20architektura.pdf
D. E. Knuth	2005	MMIX – A RISC Computer for the New Millennium	Vol. 1, Fasc. 1	Addison–Wesley, http://www.mmix.cs.hm.edu/doc/fasc1.pdf , https://www-cs-faculty.stanford.edu/~knuth/fasc1.ps.gz
C. W. Kann	2016	Implementing a One Address CPU in Logisim		Gettysburg College; https://open.umn.edu/opentextbooks/textbooks/implementing-a-one-address-cpu-in-logisim
C. W. Kann	2019	Digital Circuit Projects: An Overview of Digital Circuits Through Implementing Integrated Circuits	Second Edition	Gettysburg College; http://cupola.gettysburg.edu/oe/1
C. W. Kann	2019	Introduction To MIPS Assembly Language Programming		Gettysburg College; https://cupola.gettysburg.edu/oe/r/2
M. J. Murdocca and V. P. Heuring	1999	Principles of Computer Architecture		Prentice Hall
D. A. Patterson and J. L. Hennessy	2013	Computer Organization and Design: The Hardware/Software Interface. MIPS edition.		Elsevier
E. Upton	2016	Learning Computer Architecture with Raspberry Pi		John Wiley & Sons
A. P. Malvino and J. A. Brown	1999	Digital Computer Electronics		McGraw-Hill
R. E. Bryant and D. R. O'Hallaron	2001	Computer Systems: A Programmer's Perspective	3rd Edition	https://github.com/smellslikekeenspirit/askreddit-list-of-compsci-books/blob/master/Randal%20E.%20Bryant%2C%20David%20R.%20O%20E2%80%99Hallaron%20-%20Computer%20Systems.%20A%20Programmer%20E2%80%99s%20Perspective

				%20%5B3rd%20ed.%5D%20(2016%2C%20Pearson).pdf
D. Goldberg	1991	What every computer scientist should know about floating-point arithmetic		https://doi.org/10.1145/103162.103163
J. L. Gustafson	2015	The End of Error: Unum Computing		CRC Press