



Проектування складних систем

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення комп'ютерних систем
Статус дисципліни	Нормативна
Форма навчання	Очна(денна)
Рік підготовки, семестр	4 курс, весняний семестр
Обсяг дисципліни	4,5 кредитів(ECTS),135 годин
Семестровий контроль/ контрольні заходи	Залік
Розклад занять	Лекцій -36 годин,лабораторних робіт - 18 годин,самостійна робота студентів-81 годин
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: доцент, к.т.н, с.н.с. Долголенко Олександр Миколайович, aleks.dolgolenko@gmail.com . Лабораторні роботи: доцент, к.т.н, с.н.с. Долголенко Олександр Миколайович, aleks.dolgolenko@gmail.com .
Розміщення курсу	https://campus.kpi.ua/tutor/index.php?mode=mob

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Дисципліна спрямована на вивчення можливостей інноваційних технологій, що реалізовані в сучасних мікропроцесорах фірми Intel. Вивчення даної дисципліни бакалаврами дозволить їм використовувати ці технології у своїй професійній діяльності.

Метою вивчення дисципліни “Проектування складних систем” є формування у студентів здатностей та навичок використання можливостей інноваційних технологій, що реалізовані в мікропроцесорах, з якими вони, скоріше за все, будуть мати справу під час своєї професійної діяльності.

Предметом дисципліни є:

- види паралелізму в роботі ядра сучасного мікропроцесора;
- архітектура x86-64;
- реалізація обмеженої схеми потоку даних в ядрі мікропроцесора;
- декодування CISC на набір RISC;
- принципи розробки пристрою передбачення галуження;
- механізм перейменування регістрів;
- принципи побудови кеш-пам'яті;
- асоціативність кеш-пам'яті та TLB буферів;

- механізми когерентності кешів;
- інноваційні технології, що реалізовані в сучасних мікропроцесорах.

Згідно з вимогами ОНП здобувачі після засвоєння дисципліни “Проектування складних систем” мають продемонструвати такі програмні результати навчання:

знання:

- особливостей виконання окремого потоку команд в ядрі сучасного мікропроцесора (ПРН26);
- особливостей виконання паралельної програми в сучасному мікропроцесорі (ПРН26);
- особливостей роботи кеш-пам'яті та механізму когерентності кешів в сучасних мікропроцесорах (ПРН26);
- можливостей інноваційних технологій, що реалізовані в сучасних мікропроцесорах (ПРН26)

уміння:

- вибирати оптимальну стратегію розміщення змінних паралельної програми: в стеку, відповідно до принципів локалізації та незмінності, чи в кучі;
- аналізувати ефективність паралельного алгоритму (ФК17);
- використовувати можливості інноваційних технологій, що реалізовані в мікропроцесорах, при розробці програмного забезпечення для високопродуктивних комп'ютерних систем (ФК18)

досвід:

- побудови та аналізу паралельного алгоритму;
- створення та налагодження паралельної програми;
- використання можливостей інноваційних технологій.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Для успішного оволодіння дисципліною необхідні знання з дисциплін: “Структури даних та алгоритми”, “Архітектура комп'ютерів”, “Комп'ютерні системи”, “Програмування”, “Об'єктно-орієнтоване програмування”, “Паралельні обчислення”, “Багатопотокове програмування на Java”.

Дисципліни, які базуються на результатах навчання з даної дисципліни: “Організація високопродуктивних обчислень”, “Планування обчислень в паралельних та розподілених комп'ютерних системах”.

Зміст навчальної дисципліни

Назви розділів і тем	Кількість годин				
	Всього	у тому числі			
		Лекції	Практичні (семінарські)	Лабораторні (комп'ютерний практикум)	СРС
1	2	3	4	5	6
Розділ 1. Архітектурні особливості побудови ядра сучасного мікропроцесора					
Тема 1.1. Види паралелізму в роботі ядра сучасного	5	2	-	-	3

1	2	3	4	5	6
мікропроцесора, побітовий паралелізм та паралелізм на рівні даних (скалярні та векторні інструкції).					
Тема 1.2. Паралелізм на рівні команд.	5	2	-	-	3
Тема 1.3. Паралелізм на рівні потоків команд.	5	2	-	-	3
Тема 1.4. Архітектура x86-64.	5	2	-	-	3
Разом за розділом 1	20	8	-	-	12
Розділ 2. Суперскалярна архітектура(CISC-RISCархітектура)					
Тема 2.1. Декодування CISC на набірRISC.	5	2	-	-	3
Тема 2.2. Принципи побудови кеш-пам'яті.	5	2	-	-	3
Тема 2.3. Асоціативність кеш-пам'яті таTLB буферів.	5	2	-	-	3
Тема 2.4. Механізми когерентності кешів.	13	2	-	6	5
Тема 2.5. Принципи розробки пристрою передбачення галуження.	5	2	-	-	3
Тема 2.6. Перейменування регістрів.	5	2	-	-	3
Тема 2.7. Покращений алгоритмTomasulo.	5	2	-	-	3
Разом за розділом 2	43	14	-	6	23
Розділ 3. Інноваційні технології, що реалізовані в сучасних мікропроцесорах					
Тема 3.1. Апаратний пропуск замків.	12	2	-	4	6
Тема 3.2. Апаратна реалізація транзакційної пам'яті.	16	2	-	8	6
Тема 3.3.Intel® OoOE та Intel® QuickPath технології	3	2	-	-	1

1	2	3	4	5	6
Тема 3.4 Логічна сегментація оперативної пам'яті. Багатоканальність оперативної пам'яті.	11	4	-	-	7
Тема 3.5. Intel® vPro™ технології.	5	2	-	-	3
Тема 3.6. Віддалене адміністрування.	5	2	-	-	3
Разом за розділом 3	52	14	-	12	26
Модульна контрольна робота	8				8
Підготовка до заліку	4				4
Залік	8				8
Всього в семестрі:	135	36	-	18	81

3. Навчальні матеріали та ресурси

1. John L. Hennessy, David A. Patterson. Computer Architecture. A Quantitative Approach, USA, Morgan Kaufmann, 6th, 2017 – 665p.+ add-ins.
2. J. Shen, M. Lipasti. Modern Processor Design: Fundamentals of Superscalar Processors. Waveland Press, 2013 - 642 p.
3. Intel® 64 and IA-32 Architectures Software Developer Manuals (<https://software.intel.com/en-us/articles/intel-sdm>).
4. x86 and amd64 instruction reference (<https://www.felixcloutier.com/x86/>).

Навчальний контент

4. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

№ лекції	Назва теми лекції та перелік основних питань (перелік дидактичних засобів, посилання на літературу та завдання на СРС)
1	Архітектурні особливості побудови ядра сучасного мікропроцесора [1, с. 148-366]. Види паралелізму в роботі ядра сучасного мікропроцесора, побітовий паралелізм та паралелізм на рівні даних (скалярні та векторні інструкції).
2	Паралелізм на рівні команд: рівень IPC (instruction per clock cycle); алгоритми Scoreboarding та Tomasulo, як засіб досягнення IPC>1.
3	Паралелізм на рівні потоків команд: Hyperthreading, Speculation Multithreading, Tasks-Level Parallelism, Fiber паралелізм, Thread pool.
4	Архітектура x86-64: основні регістри, індексні регістри, додаткові регістри, показник команд, сегментні регістри, XMM (YMM, ZMM, SSE) регістри, регістр станів, скалярні та векторні регістри, введення в механізм перейменування регістрів.
5	Суперскалярна архітектура (CISC-RISC архітектура) , Data Flow, Restricted Data Flow, Front-end, декодування CISC на набір RISC, RISC кеш (L0), моделі розвитку:

	tick-tock та process-architecture-optimization в розробці мікропроцесорів Intel з архітектурою CISC-RISC.
6	Принципи побудови кеш-пам'яті: TLB буфери та віртуальна пам'ять; дані таблиці сторінок; багаторівневі таблиці сторінок; flat memory model; paged memory model; X86 segmented memory model; віртуальна, лінійна та фізична адреси; три режими обробки сторінок в архітектурі X86 - 64 [1, с. 247-284].
7	Асоціативність кеш-пам'яті та TLB буферів, політики заміщення блоків даних в кеші, вплив асоціативності на роботу процесорного ядра.
8	Механізми когерентності кешів, snooping, MESI, MOESI та MESIF протоколи.
9	Принципи розробки пристрою передбачення галуження, передбачення галуження з використанням BTB, історія галужень, кінцевий автомат станів для передбачення напряду галуження на підставі його попередньої історії.
10	Back-end, покращений алгоритм Tomasulo, загрози відмов за даними, станції резервування [1, с. 71-105; 3, с. 177-209, 246-261].
11	Перейменування регістрів, Allocator, буфер переупорядкування результатів [3, с. 217-245].
12	Інноваційні технології, що реалізовані в сучасних мікропроцесорах. Апаратний пропуск замків, IntelTransactionalSynchronizationExtensions (TSX)-NI® технологія, speculativemultithreading, HardwareLockElision (HLE), виникнення та розвиток HLE.
13	Апаратна реалізація транзакційної пам'яті, Restricted Transactional Memory, виникнення та розвиток Hardware Transactional Memory, Haswell's Transactional Memory.
14	Intel® OoOE технологія, Intel® QuickPath технологія.
15	Ієрархія пам'яті комп'ютера [1, с. 1158-1169], стіна пам'яті, логічна сегментація оперативної пам'яті.
16	Багатоканальність оперативної пам'яті, вбудована DRAM.
17	Технології Intel® vPro™, Hyperthreading, VT-x, VT-d, Turbo Boost 3.0, Trusted Execution технологія, Cisco Self Defending Network, Microsoft Network Access Protection, Execute disable біт.
18	Віддалене адміністрування, Intel Active Management технологія, універсальний унікальний ідентифікатор, vPro-enabled gateway, Intel Management Engine, MINIX 3.

Лабораторні роботи (комп'ютерний практикум)

Основні завдання циклу лабораторних робіт (комп'ютерного практикуму) - придбання студентами необхідних практичних навичок по використанню можливостей інноваційних технологій, реалізованих в сучасних мікропроцесорах від Intel.

№ з/п	Назва лабораторної роботи (комп'ютерного практикуму)	Кількість ауд. годин
1	Дослідження протоколу когерентності кешівMESIF. Розробіть два екземпляра свого варіантабагатопотокової програми наJava. Перший екземплярбагатопотокової програми розробіть за дотриманням принципу локалізації, при цьому як скалярні так і векторні змінні не повинні розділюватися між потоками – механізм когерентності кешівMESIF при цьому	6

	працювати не буде. Другий екземпляр багатопотокової програми розробіть за використанням скалярних та векторних глобальних змінних та механізмів синхронізації потоків при зверненні до цих змінних – пошук актуального значення змінної при цьому буде відбуватися, відповідно до механізму когерентності кешівMESIF, у кешах L1D всіх ядер мікропроцесора. Побудуйте порівняльні графіки й зробіть висновки по відносній швидкодії цих екземплярів програми.	
2	Дослідження технології апаратного пропуску замків (Intel's HardwareLockElision). Для виконання цієї роботи потрібен доступ до мікропроцесора Intel, що підтримує технологію TSX-NI. Створіть потоки з використанням інтерфейсу <i>Callable</i> з пакету <i>java.util.concurrent</i> (синхронізація потоків за допомогою інтерфейсу <i>Lock</i> із пакета <i>java.util.concurrent.lock</i>). Дослідіть швидкість виконання Вашої паралельної програми на цьому мікропроцесорі. Після цього відключіть Intel® TSX-NI за допомогою наступного налагоджування реєстру (для Windows 10): <code>regadd "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SessionManager\Kernel" /vDisableTsx /tREG_DWORD /d 1 /f</code> . Перепузьте комп'ютер, щоб ці зміни мали ефект та знову дослідіть швидкість виконання варіантів Вашої паралельної програми. Побудуйте порівняльні графіки й зробіть висновки по роботі. Для наступного включення технології Intel® TSX-NI, аналогічно виконайте: <code>regadd "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SessionManager\Kernel" /vDisableTsx /tREG_DWORD /d 0 /f</code> .	4
3	Дослідження технології апаратної реалізації транзакційної пам'яті. Для виконання цієї роботи потрібен доступ до мікропроцесора Intel, що підтримує технологіюTSX-NI. Розгорніть на ньому якусь СУБД. Організуйте доступ до СУБД за допомогою спеціального API, наприклад, <i>javaх.sql</i> . Розробіть тестову програму на Java для роботи з СУБД і дослідіть продуктивність своєї СУБД при використанні <i>hardwaretransactionalmemory</i> . Після цього відключіть на своєму мікропроцесорі підтримку технологіїTSX-NI(див. виконання другої лабораторної роботи”) і на цій же апаратурі розгорніть ту ж саму СУБД і Java (деякі бібліотеки при цьому будуть іншими). Дослідіть продуктивність цього варіанту СУБД на тій же тестовій програмі при використанні <i>softwaretransactionalmemory</i> . Приведіть порівняльні графіки й зробіть висновки по роботі.	8
	Разом:	18

Вказівки до виконання лабораторних робіт:

- для вибраної Вами розмірності векторів та матриць (не менше 100) розробіть багатопотокову програму, що обчислює Ваш варіант завдання та час її рішення;
- всі початкові дані – додатні числа з плаваючою крапкою;
- згенеруйте початкові дані, при цьому забезпечте, щоб порядки елементів матриць та векторів були різними (чим більше будуть відрізнятися порядки, тим більш помітним буде, що стандартні математичні закони для арифметики з плаваючою крапкою не виконуються);
- початкові дані збережіть в файлі (в двох перших лабораторних роботах початкові дані вводьте з цього файлу);

- організуйте вивід результатів обчислень функцій в файл та їх синхронізований вивід на консоль із потоків, де вони обчислюються;
- для підвищення точності обчислень з плаваючою крапкою кожне додавання послідовності чисел виконуйте з використанням алгоритму типу *Kahan*, або перед виконанням додавання послідовності чисел сперш уїї відсортуйте по зростання та потім підсумовуйте, починаючи з найменших чисел;
- з використанням засобів синхронізації розробіть паралельну програму на Java для обчислення двох математичних функцій;
- Ваш варіант математичних функцій для обчислення двох перших лабораторних робіт визначається Вашим номером в списку групи в Camrus(де: MA_i т. п. – матриця; A_i т. п.–вектор; a_i т. п.– скаляр):

ВАРІАНТ 1ВАРІАНТ 2

$$A=B*(MM+MZ)+E*MM; D=B*(ME+MZ)-E*(MM+ME);$$

$$MA=\max(B-E)*MM*MT-MZ*(MT+MM). MA = \min(MM)*(ME+MZ)-ME*MM.$$

ВАРІАНТ 3

ВАРІАНТ 4

$$MG=MB*MM+MC*MO-MM; A=B*MC+D*MZ+E*MM;$$

$$A=D*MC-\min(D)*B. MA=\max(D+B)*MZ*MM-ML*MC*a.$$

ВАРІАНТ 5

ВАРІАНТ 6

$$A=B*MC+D*MT;$$

$$MA=\min(D)*MC*ME+MZ*MT.$$

$$MA=\max(B+C)*MD*MT+MZ*MB;$$

$$E=B*MD+C*MT*a.$$

ВАРІАНТ 7

ВАРІАНТ 8

$$A=C*MC+D*MM*a-B*MT; C=B*MC-D*MM;$$

$$MA=\max(B-D)*MD*MT-MC*(MT+MM). MF=\min(B+D)*MC*MZ+MM*(MC+MM)*a.$$

ВАРІАНТ 9

ВАРІАНТ 10

$$E=B*MC+D*\min(MC); MG=MB*MO+MM*(MO-MB)d;$$

$$MA=b*MD*(MC-MX)+MX*MC*b. A=\min(B)*D-C*MO*d.$$

ВАРІАНТ 11

ВАРІАНТ 12

$$Y=D*MT+\max(B)*D; A=B*MC+D*MZ+E*MM;$$

$$MA=MT*(MT+MZ)-MZ*MT. MG=\min(D+E)*MM*MT-MZ*ME.$$

ВАРІАНТ 13

ВАРІАНТ 14

$$B=D*MO-\min(D)*C; MA=\max(MD)*(MT+MZ)-MT*MD;$$

$$S=\text{SORT}(MO*D+D+C). E=B*MD+D*MT.$$

ВАРІАНТ 15

ВАРІАНТ 16

$$MA=MD*MT+MZ-ME*MM; MG=MD*(MT+MZ)-MT*MZ;$$

$$A=D*MT-\max(D)*B. A=\min(B)*D*MT+B.$$

ВАРІАНТ 17

$MA = \min(D) * MD * MX + MZ * ME; MA = \min(D + B) * MD * MT + MX * ME;$
 $E = B * MD - D * MZ. C = B * MT + D * MX * a.$

ВАРІАНТ 18**ВАРІАНТ 19**

$MG = MB * MO + MS * (MB - MO); O = \text{SORT}(P) * \text{SORT}(MR * MS);$
 $S = \text{SORT}(O * MO + B * MB).$

ВАРІАНТ 20

$MG = MR * MS + MC * (MR + MS).$

ВАРІАНТ 21

$A = B * MC + D * MZ + E * MT;$
 $MG = \min(D + B) * MC * MT - MZ * ME. B = D * MT - \min(D) * C.$

ВАРІАНТ 22

$MG = MB * MT + MC * (MB * MM + MT);$

ВАРІАНТ 23

$MF = MD * (ME + MM) - ME * MM; A = \min(C) * C * MT + C;$
 $E = B * ME + D * \max(MM). MX = \min(C - D) * MD * MT - MC * (MC + MT).$

ВАРІАНТ 24**ВАРІАНТ 25**

$MD = \min(MT) * (MT + MX) - MT * MX; MG = \max(A + C) * MB * MT - MZ * ME * a;$
 $D = B * (MT + MX) - E * (MT - MX);$

ВАРІАНТ 26

$X = A * MB - \min(C) * C.$

ВАРІАНТ 27

$MA = MM * (ME - MX) + ME * MX * q; MF = \min(C - D) * MC * MZ + MM * (MC + MM);$
 $Y = B * ME + D * \min(B). X = \text{SORT}(MC * M + D - C).$

ВАРІАНТ 28**5. Самостійна робота бакалавра:**

- підготовка до лекцій (рекомендована література приведена в табл. «Лекційні заняття»);
- підготовка лабораторних робіт;
- підготовка до заліку.

Політика та контроль**6. Політика навчальної дисципліни (освітнього компонента)**

Під час занять з навчальної дисципліни «Проектування складних систем» бакалаври повинні дотримуватись певних дисциплінарних правил:

- забороняється запізнюватись на заняття;
- не допускаються сторонні розмови або інший шум, що заважає проведенню занять;
- виходити з аудиторії під час заняття допускається лише з дозволу викладача;
- не допускається користування мобільними телефонами та іншими технічними засобами без дозволу викладача.

7. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Поточний контроль:

Протягом семестру студенти виконують 3 лабораторні роботи.

Максимальна кількість балів за кожну з лабораторних робіт: 20 балів.

Бали нараховуються за коректність оформлення протоколу лабораторної роботи, експрес-опитування за темою лабораторної роботи та відповідність змісту виконаної роботи завданню на неї. На перші дві робіт діє граничний термін (deadline) на її здачу, відповідно до табл. «Лабораторні роботи». Перший комп'ютерний практикум, за розкладом, відноситься до часу виконання першої лабораторної роботи. Максимальний бал за першу роботу можливо отримати на 2 та 3 комп'ютерних практикумах. На 4 комп'ютерному практикумі: за 1 роботу можливо отримати не більше 17 балів; за другу – до 20 балів. Наступний крок deadline починається з 6 комп'ютерного практикуму: за 1 роботу можливо отримати не більше 14 балів; за другу – не більше 17 балів; за третю – до 20 балів. Виконана робота не може бути оціненою нижче ніж 10 балами.

Разом, за лабораторним роботами, можливо набрати до 60 балів. За модульну контрольну роботу можна отримати до 10 балів.

Календарний контроль:

Для бакалаврів передбачено один календарний контроль, що відбувається відповідно до терміну, визначеного в Campus, та виставляється в пункті Campus\«Календарний контроль». Для його успішного проходження потрібно мати хоча б одну здану лабораторну роботу. Оцінки за здані лабораторні роботи виставляються в пункті Campus\«Поточний контроль».

Семестровий контроль:

Семестровий контроль включає залік. На залікові можливо набрати до 30 балів.

Умови допуску до семестрового контролю: семестровий рейтинг – 30 балів і вище.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка ECTS
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

8. Додаткова інформація з дисципліни (освітнього компонента)

Перелік запитань, які виносяться на семестровий контроль буде надано в останній лекції курсу.

Робочу програму навчальної дисципліни (силабус):

Складено к.т.н, с.н.с., доцентом, Долголенко Олександром Миколайовичем.

Ухвалено кафедрою обчислювальної техніки (№ 10 від 25. 05. 2022 р.).

Погоджено Методичною комісією ФІОТ (протокол № 10 від 09. 06. 2022 р.).