



ПАРАЛЕЛЬНІ ТА РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	123 Комп'ютерна інженерія
Освітня програма	Комп'ютерні системи та мережі
Статус дисципліни	Вибіркова
Форма навчання	очна(денна)заочна
Рік підготовки, семестр	3 курс, весінній семестр
Обсяг дисципліни	4 кредитів 120 год
Семестровий контроль/ контрольні заходи	Залік, МКР
Розклад занять	//rozklad.kpi.ua
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: к.т.н, доцент, Корочкін Олександр Володимирович avcora@gmail.com Лабораторні: к.т.н, доцент, Корочкін Олександр Володимирович avcora@gmail.com
Розміщення курсу	//comsys.kpi.ua GOOGLE

Програма навчальної дисципліни

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

Дисципліна "Паралельні та розподілені обчислення" спрямована на опанування студентами технологій розробки паралельних програм. Знання основ паралельного програмування необхідні для створення програмного забезпечення для паралельних та розподілених комп'ютерних систем, систем реального часу, Інтернет-додатків, мобільних пристроїв.

Дисципліна є продовженням нормативної дисципліни «Паралельне програмування».

Метою викладання дисципліни "Паралельні та розподілені обчислення" є отримання студентом знань та навичок у сфері практичної розробки програмного забезпечення па для паралельних комп'ютерних та розподілених систем. Навчальна дисципліна формує у студентів компетентності розробки програмного забезпечення для комп'ютерних систем з паралельною або розподіленою архітектурою на основі володіння засобами сучасних мов та бібліотек паралельного програмування на робочих місцях і підрозділах під час майбутньої професійної діяльності на першій посаді. За результатами вивчення дисципліни студент має бути здатними вирішувати професійні завдання та володіти такими оцінювати особливості структури ПКС, що застосовуються; використовувати методи створення та оцінювання паралельних алгоритмів; використовувати засоби сучасних мов та бібліотек паралельного програмування для створення програмного забезпечення сучасних ПКС, володіти основними методами формування програм на основі концепції процесів, вміти організовувати оптимальну взаємодію потоків.

ЗДАТНІСТЬ:

- розробки програмного забезпечення для комп'ютерних систем з паралельною або розподіленою архітектурою на основі володіння засобами сучасних мов та бібліотек паралельного програмування
- аналізувати структуру паралельної комп'ютерної системи
- аналізувати паралельні властивості задачі, що вирішується
- розробляти та оцінювати паралельні алгоритми
- здійснювати організацію обчислень в паралельних та розподілених комп'ютерних системах

ЗНАННЯ:

- методів та засобів організації обчислень у високопродуктивних комп'ютерних системах
- структур сучасних паралельних комп'ютерних систем, організації пам'яті та зв'язку процесорів,
- методів аналізу, оцінювання та представлення паралельних алгоритмів, концепцій розробки паралельних програм
- методів програмування потоків, концепції потоку, операцій з потоками,
- методів та засобів організації взаємодії потоків ,
- моделей взаємодії процесів,
- постанови та методів вирішення завдання взаємного виключення та синхронізації процесів,
- методів налагодження паралельних програм.

УМІННЯ:

- створювати програмне забезпечення для високопродуктивних комп'ютерних систем з різною архітектурою ,
- виконувати побудову паралельного алгоритму,
- аналізувати ефективність паралельного алгоритму, формувати алгоритми потоків,
- програмувати потоки та створювати паралельну програму,
- організовувати ефективну взаємодію потоків в залежності від структури ПКС,
- розміщувати потоки по процесорах ПКС (вузлах ПКС)
- виконувати налагодження та виконання паралельної (розподіленої) програми

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

Необхідні дисципліни: "Програмування", "Об'єктна - орієнтоване програмування", "Системне програмування", "Структури даних та алгоритми", "Інженерія програмного забезпечення", "Алгоритми та методи обчислень", "Паралельне програмування"

Дисципліни, які базуються на результатах навчання з даної дисципліни: Системне програмне забезпечення", "Комп'ютерні системи"

3. Зміст навчальної дисципліни

Розділ 1. Розробка паралельних програм. Життєвий цикл

Тема 1.1 Розробка паралельного математичного алгоритму

Тема 1.2 Розробка алгоритмів потоків

Тема 1.3 Розробка схеми взаємодії потоків

Тема 1.4 Розробка програми

Розділ 2. Програмування для ПКС з загальною пам'яттю (ПКС СП). Використання низкорівневих засобів організації взаємодії потоків

Тема 2.1 Мова Ада. Семафори. Атоміс змінні

Тема 2.2 Мова Java. Семафори. Бар'єри

Тема 2.3 Бібліотека WinAPI. Семафори. Мютекси. Події

Тема 2.4 Мова C#. Семафори. Мютекси. Події

Тема 2.5 Мова Python. Семафори. Мютекси. Події

Розділ 3. Бібліотека OpenMP.

3.1 Створення потоків

3.2 Взаємодія потоків

Розділ 4. Програмування для ПКС з загальною пам'яттю. Використання критичних секцій для організації взаємодії потоків

4.1 Бібліотека WinAPI. Критичні секції

4.2 Мова Java. Синхронізовані блоки

4.3 Мова Мова C#. Мова C#.

Розділ 5. Програмування для ПКС з загальною пам'яттю. Використання моніторів для організації взаємодії потоків

Тема 5.1 Мова Ада. Захищений модуль

Тема 5.2 Мова Java. Монітор

Розділ 6. Програмування для ПКС з загальною пам'яттю. Використання моніторів для організації взаємодії потоків

Тема 6.1 Мова Ада. Захищений модуль

Тема 6..2 Мова Java. Монітор

Розділ 7. Програмування для ПКС з розподіленою пам'яттю.

7.1. Мова Ада. Механізм рандеву

7.2 Бібліотека MPI

Розділ 8 Програмування для розподілених ПКС

Тема 8.1 Використання сокетів

Тема 8.2 Використання віддалених методів

Методичні рекомендації

Методика вивчення дисципліни потребує використання сучасних апаратних і програмних засобів, що пов'язані з паралельною та розподіленою обробкою. Що стосовно апаратної складової, то в першу чергу слід розглядати паралельні комп'ютерні системи з багатоядерною архітектурою, а також розподілені (кластерні) системи. Слід приділити увагу структурній організації сучасних багатоядерних процесорів, організації багаторівневої кеш-пам'яті та зв'язку ядер і вузлів кластерів. Лабораторні роботи повинні виконуватися в класі, які обладнано таким багатоядерними комп'ютерами, які об'єднано потужною мережею .

Сучасні програмні засоби, що дозволяють програмувати для паралельних (розподілених) комп'ютерних систем базуються на використанні мов та бібліотек паралельного програмування. До таких відносяться мови Java, C#, Ada. OpenMP, MPI, PVM. Слід розглянути всі такі системи, бо вони відрізняються як по концепції створення потоків, так і по засобам організації взаємодії потоків.

В якості базової мови можна обрати мову Java, яка підтримує класичний підхід до створення потоків через спеціальні модулі task, а також реалізує обидві моделі організації взаємодії процесів як через спільні змінні, так і через посилання повідомлень. Крім того, ресурси мови Java дозволяють створювати програмне забезпечення і для розподілених систем.

4. Навчальні матеріали та ресурси

Базова:

1. Жуков І., Корочкін О. Паралельні та розподілені обчислення. Навч. посібн. 2-ге видання - Київ: «Корнійчук», 2014. - 284 с. Гриф надано МОН, лист від 04.07.05 № 14/18.2– 1540).
2. Методичні указівки для виконання лабораторних робіт з дисципліни “Паралельні та розподілені обчислення” //Корочкін О.В. – Київ : КПІ ім. Ігоря Сікорського, 2022 – 24 с. Електронний ресурс Погоджено Методичною радою ФІОТ (протокол № 10 від 9.06.2022 р.), <https://comsys.kpi.ua/metodichni-vkazannya-po-disciplinam>
3. Loutsky G., Zhukov I., Korochkin A. Parallel Computing. – Kyiv, Kornechuk, 2007. - 216 pp.
4. Паралельні та розподілені обчислення. Вибрані розділи: Навч. посібник для здобувачів ступеня бакалавр за спеціальністю 123 «Комп’ютерні системи та мережі» / Корочкін О.В., Русанова О.В. – Київ : КПІ ім. Ігоря Сікорського, 2020. – 123с Електронний ресурс. Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 6 від 31.01.2020 р.) за поданням Вченої ради ФІОТ (протокол № 4 від 25.11.2019 р.), <http://ela.kpi.ua/handle/123456789/48224>

Додаткова:

5. Korochkin O. Багатоядерне програмування в мові Ада. (Multicore programming in Ada): Навч. посібник для здобувачів ступеня бакалавр за спеціальністю 123 «Комп’ютерні системи та мережі» Київ, НТУУ-КПІ, 2018.- 114 с. Електронний ресурс. Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 6 від 31.01.2020 р.) за поданням Вченої ради ФІОТ (протокол № 4 від 25.11.2019 р.), <https://comsys.kpi.ua/metodichni-vkazannya-po-disciplinam>

SemaphiresНавчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Очна форма

Назви розділів і тем	Кількість годин			
	Всього	у тому числі		
		Лекції	Практичні (семінарські)	Лабораторні роботи

1	2	3	4	5	6
Розділ 1. Розробка паралельних програм. Життєвий цикл					
Тема 1.1 Розробка паралельного математичного алгоритму	1	0,5	-	-	0,5
Тема 1.2 Розробка алгоритмів потоків	1	0,5	-	-	0,5
Тема 1.3 Розробка схеми взаємодії потоків	1	0,5			0,5
Тема 1.4 Розробка програми	1	0,5			0,5
Разом за розділом 1	4	2	-	-	2
Розділ 2. Програмування для ПКС СП. Використання низкорівневих засобів організації взаємодії потоків					
Тема 2.1 Мова Ада. Семафори. Атомік змінні	3	2	-	-	1
Тема 2.2 Мова	3	2	-	-	1

1	2	3	4	5	6
<i>Java.Семафори.Бар'єри</i>					
<i>Тема 2.3 Бібліотека WinAPI. Семафори.Мютекси.Події</i>	2	2			
<i>Тема 2.4 Мова С#. Семафори.Мютекси.Події</i>	3	2			1
<i>Тема 2.5 Мова Python. Семафори.Мютекси.Події</i>	3	2			1
<i>Разом за розділом 2</i>	14	10	-	-	4
Розділ 3. Бібліотека OpenMP.					
<i>3.1 Створення потоків</i>	3	1	-	-	2
<i>3.2 Взаємодія потоків</i>	3	1	-	-	2
<i>Разом за розділом 3</i>	6	2			4
Розділ 4. Програмування для ПКС СП. Використання критичних секцій для організації взаємодії потоків					
<i>4.1 Бібліотека WinAPI. Критичні секції</i>	10	2		3	5
<i>4.2 Мова Java. Синхронізовані блоки</i>	15	2	-	3	10
<i>4.3 Мова С#. Замки</i>	15	2		2	11
<i>Разом за розділом 4</i>	40	6	-	8	26
Розділ 5. Програмування для ПКС з загальною пам'яттю. Використання моніторів для організації взаємодії потоків					
<i>Тема 5.1 Мова Ада. Захищений модуль</i>	4	2	-	2	2
<i>Тема 5.2 Мова Java. Монітор</i>	2	2	-	-	
<i>Разом за розділом 5</i>	6	4	-	2	2
Розділ 6. Програмування для ПКС СП. Використання Atomic змінних					
<i>Тема 6.1 Мова Ада. Atomic змінні</i>	8	2		2	4
<i>Тема 6.2 Мова Java. Atomic, Volatile змінні</i>	8	2	-	-	6
<i>Разом за розділом 6</i>	16	4	-	2	12
Розділ 7. Програмування для ПКС ЛП					
<i>7.1.Мова Ада. Механізм рандеву</i>	5	2	-	-	6
<i>7.2 Бібліотека MPI</i>	5	2	-	-	6
<i>Разом за розділом 7</i>	10	4	-	3	3
Розділ 8. Програмування для розподілених КС (ПКС)					
<i>Тема 8.1 Використання сокетів</i>	7	2	-	-	3
<i>Тема 8.2 Використання віддалених методів</i>	7	2	-	-	4
<i>Разом за розділом 8</i>	14	4	-	3	7
МКР	2	2			2
Залік	8				8
Всього в семестрі:	120	36	-	18	66

Назви розділів і тем	Кількість годин				
	Всього	у тому числі			
		Лекції	Практичні (семінарські)	Лабораторні роботи	СРС
1	2	3	4	5	6
Розділ 1. Розробка паралельних програм. Життєвий цикл					
Тема 1.1 Розробка паралельного математичного алгоритму	1	0,5	-	-	0,5
Тема 1.2 Розробка алгоритмів потоків	1	0,5	-	-	0,5
Тема 1.3 Розробка схеми взаємодії потоків	1	0,5			0,5
Тема 1.4 Розробка програми	1	0,5			0,5
Разом за розділом 1	4	2	-	-	2
Розділ 2. Програмування для ПКС СП. Використання низкорівневих засобів організації взаємодії потоків					
Тема 2.1 Мова Ада. Семафори. Атомік змінні	3	2	-	-	1
Тема 2.2 Мова Java. Семафори. Бар'єри	3	2	-	-	1
Тема 2.3 Бібліотека WinAPI. Семафори. Мютекси. Події	2	2			
Тема 2.4 Мова С#. Семафори. Мютекси. Події	3	2			1
Тема 2.5 Мова Python. Семафори. Мютекси. Події	3	2			1
Разом за розділом 2	14	10	-	-	4
Розділ 3. Бібліотека OpenMP.					
3.1 Створення потоків	3	1	-	-	2
3.2 Взаємодія потоків	3	1	-	-	2
Разом за розділом 3	6	2			4
Розділ 4. Програмування для ПКС СП. Використання критичних секцій для організації взаємодії потоків					
4.1 Бібліотека WinAPI. Критичні секції	10	2		1	7
4.2 Мова Java. Синхронізовані блоки	15	2	-	1	12
4.3 Мова С#. Замки	15	2		1	12
Разом за розділом 4	40	6	-	3	31
Розділ 5. Програмування для ПКС з загальною пам'яттю. Використання моніторів для організації взаємодії потоків					
Тема 5.1 Мова Ада. Захищений	4	2	-	1	1

1	2	3	4	5	6
модуль					
Тема 5.2 Мова Java. Монітор	2	2	-	-	
Разом за розділом 5	6	4	-	1	1
Розділ 6. Програмування для ПКС СП. Використання Atomic змінних					
Тема 6.1 Мова Ада. Atomic змінні	8	2		1	5
Тема 6.2 Мова Java. Atomic, Volatile змінні	8	2	-	-	6
Разом за розділом 6	16	4	-	1	11
Розділ 7. Програмування для ПКС ЛП					
7.1. Мова Ада. Механізм рандеву	5	2	-	1	2
7.2 Бібліотека MPI	5	2	-	-	3
Разом за розділом 7	10	4	-	1	5
Розділ 8. Програмування для розподілених КС (РКС)					
Тема 8.1 Використання сокетів	7	2	-	1	4
Тема 8.2 Використання віддалених методів	7	2	-	-	5
Разом за розділом 8	14	4	-	1	9
МКР	2	2			2
Залік	8				8
Всього в семестрі:	120	8	-	8	104

Лекційні заняття

№ лекції	Назва теми лекції та перелік основних питань (перелік дидактичних засобів, посилання на літературу та завдання на СРС)
1	<p>Розробка паралельних програм. Життєвий цикл.</p> <p>Особливості завдання паралельного програмування: дослідження задач на паралелізм, розробка паралельних алгоритмів, програмування паралельних процесів, взаємодія процесів, рішення задачі взаємного виключення, синхронізація процесів, розміщення процесів по процесорах, планування виконання процесів, налагодження та виконання [1, с.37-38.]</p> <p>Особливості життєвого циклу розробки програм для паралельних систем. Етапи розробки: аналіз вимог, проектування, реалізація, тестування, виробництво, модифікація, супровід. [1, с.38- 48].</p> <p>СРС: Виконати аналіз життєвого циклу розробки програм для операції пошуку максимального елемента матриці. [1, с. 66-67].</p>
2	<p>Аналіз і побудова паралельних алгоритмів</p> <p>Особливості завдання паралельного програмування: дослідження задач на паралелізм, розробка паралельних алгоритмів, програмування паралельних процесів, взаємодія процесів, рішення задачі взаємного виключення, синхронізація процесів, розміщення процесів по процесорах, планування виконання процесів, налагодження та виконання [1, с.37-38.]</p> <p>Особливості життєвого циклу розробки програм для паралельних систем.</p>

	<p>Етапи розробки: аналіз вимог, проектування, реалізація, тестування, виробництво, модифікація, супровід.[1, с.38- 48].</p> <p>СРС: Виконати аналіз життєвого циклу розробки програм для операції пошуку максимального елемента матриці .[1, с. 66-67].</p>
3	<p>Програмування для КС з спільною пам'яттю. Низкорівневі засоби .Мова Ада Використання семафорів. Тип <code>Synchronized_Object</code> [1, с.38- 44].</p> <p>СРС: Пакети <code>Asynchronous_Task_Control</code>, <code>Synchronous_Task_Control</code>, [1, с. 76-77].</p>
4	<p>Програмування для КС з спільною пам'яттю. Низкорівневі засоби .Мова С# Використання семафорів. Клас <code>Semaphores</code>. Використання мютексів. Клас <code>Mutex</code>. Використання подій. Клас <code>Event</code>. [1, с. 42-45].</p> <p>СРС: Розробити паралельні алгоритми для заданих векторно-матричних операцій [1, с. 76-77].</p>
5	<p>Програмування для КС з спільною пам'яттю. Низкорівневі засоби .Мова Python Використання семафорів. Клас <code>Semaphores</code>. Використання мютексів. Клас <code>Mutex</code>. Використання подій. Клас <code>Event</code>. [1, с. 42-45].</p> <p>СРС: Виконати аналіз різних підходів до програмування процесів (потоків) в сучасних мовах та бібліотеках паралельного програмування [1, с. 26-54].</p>
6	<p>Програмування для КС з спільною пам'яттю. Низкорівневі засоби. Бібліотека WinAPI. Використання семафорів. Функції <code>Semaphores</code>. Використання мютексів. Клас <code>Mutex</code>. Використання подій. Клас <code>Event</code>. [1, с. 42-45].</p> <p>[1, с. 26-55]. СРС: Програмування обчислень в мові Python [1, с. 50]</p>
7	<p>Програмування для КС з спільною пам'яттю. Критичні секції . Бібліотека WinAPI Процеси у спеціальних бібліотеках паралельного програмування. Бібліотека <code>Win32</code>. Бібліотека <code>MPI</code>. Бібліотека <code>PVM</code>. Бібліотека <code>OpenMP</code> [1, с.26-55].</p> <p>СРС: Програмування обчислень в бібліотеці <code>PVM</code> [5, с. 30]</p>
8	<p>Програмування для КС з спільною пам'яттю. Критичні секції . Мова Java. Задачі взаємодії процесів: синхронізація та обмін даними. Особливості механізмів синхронізації.[1, с. 50-54]</p> <p>СРС: Особливості взаємодії процесів [1, с.28-29]</p>
9	<p>Програмування для КС з спільною пам'яттю. Критичні секції.Мова С#. Конструкція <code>lock</code>. [Додаткова 3, с. 124-130].</p> <p>СРС: Завдання Завод-Магазін [1, с.76-79]</p>
10	<p>Програмування для КС з спільною пам'яттю. Критичні секції . Мова Python.. [1, с.78-82].</p> <p>СРС: [Додаткова 3, с. 62-63].</p>
11	<p>Програмування для КС з спільною пам'яттю. Монітори. Мова Ада. Захищені модулі. Організація взаємодії потоків. [1, с. 99, 126].</p> <p>СРС: Реалізація та застосування моніторів в мові Python [16, с. 131-167].</p>
12	<p>Програмування для КС з спільною пам'яттю. Монітори. Мова Java Монітори в Java. Модифікатори <code>private</code>, <code>synchronized</code>. <code>wait()</code>, <code>notify()</code>, <code>notifyAll()</code>, [1, с. 99, 126]</p>

	CPS: Реалізація та застосування моніторів в мові C# [16, с. 131-167].
13	Програмування для КС з спільною пам'яттю. Бібліотека OpenMP Прагми <i>barrier, shared, lock</i> . Застосування . [1, с. 82-90]. CPS: Реалізація механізму критичних секцій в бібліотеці OpenMP [1, с. 127]
14	Програмування для КС з спільною пам'яттю . Atomic змінні Загальна концепція семафорів. Семафорний тип. Операції з семафорами. Застосування семафорів. Реалізація. Приклади. [1, с. 91-100]. CPS: <i>atomic</i> змінні в Python [1, с.76-79]
15	Програмування для КС з розподіленою пам'яттю. Мова Ада Загальна концепція механізму рандеву. Кострукції <i>entry, ассер</i> [1, с.102-104]. CPS: Використання оператора <i>select</i> [1, с.76-79]
16	Програмування для КС з розподіленою пам'яттю. Бібліотека MPI Реалізація моделі повідомлень. [1, с. 104-110] CPS: Завдання Завод - Магазин [1, с.76-79]
17	Програмування для РКС .Сокети Концепція сокетів. Види сокетів. Передача даних за допомогою сокетів. [1, с.152-177] CPS: Реалізація сокетів в мові Java [1, с.168-177]
18	Програмування для РКС .Віддалені процедури (RPC) Концепція віддалених методів.Реалізація сокетів в бібліотеці [1, с.153-176] CPS: Бібліотека PVM [1, 162-177] , [5, 117-122]

Лабораторні заняття

Основні завдання циклу лабораторних занять - придбання студентами необхідних практичних навиків розробки паралельних (розподілених) програм, що базуються на концепції потоків, що взаємодіють, з застосуванням механізмів організації взаємодії потоків сучасних мов і бібліотек паралельного програмування Java, Ада, C#, WinAPI, OpenMP, MPI на базі сучасного комп'ютерного обладнання у вигляді потужних багатоядерних систем.

Очна форма

№ з/п	Назва лабораторної роботи	Кількість ауд. годин
1	Програмування обчислень для ПКС СП. Використання низкорівневих засобів взаємодії потоків	3
2	Програмування обчислень для ПКС СП Використання моніторів	3
3	Програмування обчислень для ПКС СП Бібліотека OpenMP	3
4	Програмування обчислень для ПКС ЛП Модель повідомлень	4
5	Програмування обчислень для РКС	5
	Разом:	18

Заочна форма

	Назва лабораторної роботи	Кількість
--	---------------------------	-----------

№ з/п		ауд. годин
1	Програмування обчислень для ПКС СП. Використання низкорівневих засобів взаємодії потоків	3
2	Програмування обчислень для ПКС СП Використання моніторів	3
3	Програмування обчислень для ПКС СП Бібліотека OpenMP	3
4	Програмування обчислень для ПКС ЛП Модель повідомлень	4
5	Програмування обчислень для РКС	5
	Разом:	18

6. Самостійна робота студента

Очна форма

№ з/п	Назва теми, що виноситься на самостійне опрацювання	Кількість годин СРС
1	Розробка паралельних та розподілених програм. Мова Python[11. с 153]	22
2	Розробка розподілених програм.Мова C#	22
3	Розробка розподілених програм Ada [5. с. 30-34, 117-123.]	22

Заочна форма

№ з/п	Назва теми, що виноситься на самостійне опрацювання	Кількість годин СРС
1	Розробка паралельних та розподілених програм. Мова Python[11. с 153]	17
2	Розробка розподілених програм.Мова C#	17
3	Розробка розподілених програм Ada [5. с. 30-34, 117-123.]	17
4	Програмування для КС з розподіленою пам'яттю. Сокети.WinAPI	17
5	Програмування для КС з розподіленою пам'яттю.Віддалені методи. Java	17
6	Програмування для КС з розподіленою пам'яттю.MPI	19

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

При зарахування та оцінювання лабораторних робіт беруться до уваги наступні чинники:

- Повнота виконання завдання на лабораторну роботу за індивідуальним варіантом;
- Своєчасність виконання лабораторної роботи згідно графіку;
- Самостійність виконання лабораторної роботи та відсутність ознак плагиату;
- Відповіді на питання щодо змісту лабораторної роботи під час її захисту.

При оцінюванні контрольних робіт до уваги приймаються:

- Правильність та повнота виконання завдань;

- Кількість виконаних завдань в умовах обмеженого часу;
- Самостійність виконання завдань та відсутність ознак плагіату;
- Кількість спроб виконання контрольних, які передують тій, що оцінюється.

Для підготовки до контрольних студенти отримують перелік теоретичних питань та зміст типових задач, які будуть у завданнях на контрольних.

При першій та другій атестації до уваги приймається кількість лабораторних робіт та контрольних робіт зарахованих на час проведення атестації.

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

PCO з дисципліни, семестровий контроль з якої передбачений у формі заліку, для очної форми навчання розробляється за типом PCO-1 і включає оцінювання заходів поточного контролю з дисципліни впродовж семестру.

Рейтингова оцінка здобувача складається з балів, отриманих здобувачем за результатами заходів поточного контролю, заохочувальних та штрафних балів.

Здобувачі, які виконали всі умови допуску до заліку та мають рейтингову оцінку 60 і більше балів, отримують відповідну до набраного рейтингу оцінку без додаткових випробувань.

Зі здобувачами, які виконали всі умови допуску до заліку та мають рейтингову оцінку менше 60 балів, а також з тими здобувачами, хто бажає підвищити свою рейтингову оцінку, на останньому за розкладом занятті з дисципліни в семестрі викладач проводить семестровий контроль у вигляді залікової контрольної роботи або співбесіди.

Види контролю з навчальної дисципліни включають:

Лабораторні роботи

Заплановано самостійне виконання від трьох до п'яти лабораторних робіт (за вибором). Темі лабораторних робіт узгоджені у часі та за змістом з темами лекцій. Виконання лабораторних робіт у повному обсязі дозволяє набутти практичних навичок програмування потоків зf допомогою сучасних мов та бібліотек паралельного програмування .

Поточний контроль

Передбачено проведення модульної контрольної роботи (МКР)

Семестровий контроль

Залік проводиться у вигляді співбесіди зі студентом для об'єктивного визначення рівня знань, умінь та практичних навичок, отриманих за семестр

Семестровий рейтинг студента складається з балів, які він отримує за види робіт відповідно до таблиці 1.

Таблиця 1

Оцінювання окремих видів навчальної роботи студента (у балах)

Вид навчальної роботи	Всього за видом роботи
Виконання та захист лабораторної роботи № 1	7..16
Виконання та захист лабораторної роботи № 2	7..16
Виконання та захист лабораторної роботи № 3	7..16
Виконання та захист лабораторної роботи № 4	7..16

Виконання та захист лабораторної роботи № 5		7..16
Виконання лабораторних робіт	Rл	35..80
Модульна контрольна робота (МКР)	Rк	5..20
Усього за семестр	Rп= Rл + Rк	40..100
		100
Залік у вигляді співбесіди зі студентом (додатково за бажанням)		20

Індивідуальний поточний рейтинг студента (**Rп**) складається з балів, які він отримує за виконання лабораторних робіт і МКР. Протягом семестру студенти виконують обрану кількість (3-5) лабораторних робіт. Максимальна кількість балів за кожну лабораторну роботу – 16. Бали нараховуються за:

- теоретична складова – 8 бали,
- практична складова – 8 бали.

Максимальний можливий бал за лабораторну роботу – 16 балів.

Максимальна кількість за всі лабораторні роботи $5 \times 12 = 60$ балів.

Розрахунок розміру шкали (R) рейтингу.

Сума вагових балів контрольних заходів протягом семестру становить:

$$Rп = Rл + Rк,$$

де Rп – семестровий рейтинг студента (МКР, лабораторні роботи).

Необхідною умовою допуску студента до заліку є його індивідуальний семестровий рейтинг (**Rп**), не менший, ніж 40 балів, та відсутність заборгованості з лабораторних робіт та МКР. При невиконанні згаданих вимог студент до заліку не допускається.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
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

9. Додаткова інформація з дисципліни (освітнього компонента)

В рамках вивчення дисципліни «Паралельні та розподілені обчислення» допускається зарахування балів, одержаних в результаті дистанційних курсів на платформі “Coursera”, за умови попереднього погодження програми даного курсу з викладачем та за умови отримання офіційного сертифікату.

Робочу програму навчальної дисципліни (силабус):

Складено доцент, к.т.н, доцент, Корочкін Олександр Володимирович

Ухвалено кафедрою обчислювальної техніки (протокол № 10 від 25.05.2022)

Погоджено Методичною комісією факультету (протокол № 10 від 9.06.2022)

.....