



ТЕХНОЛОГІЇ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ. КУРСОВА РОБОТА

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення інформаційних систем</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>очна(денна)</i>
Рік підготовки, семестр	<i>3 курс, весняний семестр</i>
Обсяг дисципліни	<i>1 кредит (30 год)</i>
Семестровий контроль/ контрольні заходи	<i>Залік (захист курсової роботи)</i>
Розклад занять	
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Керівники: д.т.н., професор, Стеценко Інна Вячеславівна, inna.stetsenko-fiot@ill.kpi.ua ст.викл. Дифучин Антон Юрійович, difuchin@gmail.com ст.викл. Дифучина Олександра Юріївна sashadif@gmail.com , ас. Нестеренко Костянтин Петрович k.nesterenko@kpi.ua
Розміщення курсу	Moodle: <i>Технології паралельних обчислень</i> (uc80yt) https://do.ipk.kpi.ua/course/view.php?id=2504

Програма навчальної дисципліни

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

Вивчення дисципліни спрямовано на оволодіння практичними навичками з розробки ефективних алгоритмів паралельних обчислень з використанням технологій багатопотокових (Java Multithreading, CUDA C++, OpenMPI, MPJ Express, Java Sockets, Java RPC). Увага приділяється не тільки розробці алгоритму з використанням вказаних технологій, але й експериментальному дослідженню ефективності алгоритму. Дисципліна формує навички розробки паралельних алгоритмів, розвиває навички реалізації паралельних обчислень в одно- та багатопроцесорних обчислювальних системах, розвиває навички реалізації розподілених обчислень, а також формує базові дослідницькі навички з оцінювання швидкодії обчислень.

Предмет навчальної дисципліни – методи та технології розробки паралельних програм для багатоядерних та багатопроцесорних комп'ютерних систем.

Метою дисципліни є розуміння студентами принципів та способів розробки паралельних алгоритмів з високою швидкодією, вивчення сучасних технологій для реалізації паралельних та розподілених обчислень. За результатами вивчення дисципліни студент повинен набути **досвід** з розробки паралельних алгоритмів та їх реалізації з використанням технологій паралельних та розподілених обчислень Java Multithreading, Java MPI, Java RMI.

Програмні результати навчання студента.

Студент після засвоєння навчальної дисципліни повинен **знати**:

- базові принципи паралельних обчислень,
- архітектуру багатоядерних та багатопроцесорних обчислювальних систем;
- методи розробки паралельних алгоритмів для прикладних задач;
- технології багатопотокового програмування;
- технологію Message Passing Interface (MPI);
- методи оцінювання ефективності паралельних обчислень.

Студент повинен **вміти**:

- розробляти багатопотокові програми, використовуючи низько- та високо-рівневі засоби паралельних обчислень мови Java;
- виконувати дослідження ефективності розроблених паралельних програм експериментальними методами;
- програмувати розподілені обчислення
- розробляти алгоритми для виконання обчислень в розподілених системах з використанням технології MPI.

Згідно з вимогами освітньої програми вивчення дисципліни спрямоване на оволодіння студентом таких **компетентностей**:

ЗК 2. Здатність застосовувати знання у практичних ситуаціях.

ФК 17. Здатність розробляти інформаційні системи з використанням паралельних обчислень.

Програмні результати вивчення дисципліни забезпечують такі програмні результати освітньої програми:

ПРН 12. Застосовувати на практиці ефективні підходи щодо проєктування програмного забезпечення.

ПРН 31. Знати технології паралельних обчислень, віртуалізації серверних систем.

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

Вивчення дисципліни спирається на знання, отримані студентами при вивченні дисциплін «Основи програмування», «Алгоритми та структури даних», «Проєктування алгоритмів», «Компоненти програмної інженерії. Частина 4. Якість та тестування програмного забезпечення», «Програмування веб-застосовувань». Знання та навички, набуті студентом при вивченні дисципліни, використовуються в розробці дипломних проєктів здобувачів першого (бакалаврського) рівня вищої освіти за освітньою програмою «Інженерія програмного забезпечення інформаційних систем».

3. Зміст навчальної дисципліни

Передбачено виконання курсової роботи за темою «Алгоритм _(назва вибраного алгоритму)_ та його паралельна реалізація _(назва технології чи програмного забезпечення)_». Виконання курсової роботи передбачає розробку паралельного алгоритму у відповідності до обраного варіанту та експериментальне дослідження ефективності розробленого алгоритму. Алгоритм, який розпаралелюється, та засоби його реалізації вибираються разом з викладачем у відповідності до обраного студентом рівня складності. Варіанти, в яких вказані алгоритм та програмне забезпечення, поділені за рівнем складності завдання на групи: до 100 балів, до 75 балів, до 65 балів.

Завдання до курсової роботи:

- 1) Виконати огляд існуючих реалізацій алгоритму, послідовних та паралельних, з відповідними посиланнями на джерела інформації (статті, книги, електронні ресурси). Зробити висновок про актуальність дослідження.
- 2) Виконати розробку послідовного алгоритму у відповідності до варіанту завдання та обраного програмного забезпечення для реалізації. Дослідити швидкість алгоритму при зростанні складності обчислень та зробити висновки про необхідність паралельної реалізації алгоритму.

- 3) Виконати розробку паралельного алгоритму у відповідності до варіанту завдання та обраного програмного забезпечення для реалізації. Забезпечити зручне введення даних для початку обчислень.
- 4) Виконати тестування алгоритму, що доводить коректність результатів обчислень.
- 5) Виконати дослідження швидкодії алгоритму при зростанні кількості даних для обчислень.
- 6) Виконати експериментальне дослідження прискорення розробленого алгоритму при зростанні кількості даних для обчислень. Реалізація алгоритму вважається успішною, якщо прискорення більше 1.
- 7) Дослідити вплив параметрів паралельного алгоритму на отримуване прискорення.
- 8) Зробити висновки про переваги паралельної реалізації обчислень для алгоритму, що розглядається у курсовій роботі, та програмних засобів, які використовувались.

У розділах пояснювальної записки до курсової роботи мають бути описані усі етапи розробки паралельного алгоритму та його дослідження, а також відомості про обране програмне забезпечення для виконання завдання.

Курсова робота має включати такі розділи:

Анотація (не менше 650 символів)

Ключові слова (не менше 3 словосполучень)

Вступ (про актуальність паралельних обчислень та завдання, яке виконується в рамках курсової роботи).

Розділ 1 Опис алгоритму та його відомих паралельних реалізацій (якщо такі є).

Розділ 2 Розробка послідовного алгоритму та аналіз його швидкодії

Розділ 3 Вибір програмного забезпечення для розробки паралельних обчислень та його короткий опис.

Розділ 4 Розробка паралельних обчислень алгоритму з використанням обраного програмного забезпечення: проєктування, реалізація, тестування.

Розділ 5 Дослідження ефективності паралельних обчислень алгоритму (порівняльний аналіз швидкості обчислень).

Висновки (про переваги використання паралельних обчислень та обраного програмного забезпечення).

Список використаних джерел.

Додатки (схема взаємодії паралельних процесів, лістинг коду, скріншоти).

Курсова робота має бути виконана та захищена до кінця семестру. Захист робіт з високою якістю виконання (більше 75 балів) відбувається з презентацією цієї роботи на одному з занять комп'ютерного практикуму (5-7 хвилин).

4. Навчальні матеріали та ресурси

Базова література

1. The Java Tutorials Lesson:Concurrency [Електронний ресурс] – Режим доступу: <https://docs.oracle.com/javase/tutorial/essential/concurrency/index.html>
2. Стіренко С. Г. Засоби паралельного програмування / С. Г. Стіренко. Д. В. Грибенко. О. І. Зіненко. А. В. Михайленко – Київ. 2012. – 183 с. [Електронний ресурс] – Режим доступу: <http://hpcc.kpi.ua/hpc-book/>
3. Lea D. Concurrent programming in Java: design principles and patterns / D. Lea – Addison-Wesley Professional. 2000. – 411p.
4. Foster I. Designing and Building Parallel Programs [Електронний ресурс] – Режим доступу: <http://www.mcs.anl.gov/~itf/dbpp/text/book.html>

Допоміжна література

1. Аксак Н.Г. Паралельні та розподілені обчислення: підруч./ Н.Г.Аксак. О.Г. Руденко. А.М.Гуржій. – Х.:Компанія СМІТ. 2009. – 480с.
2. Семеренко В.П. Технології паралельних обчислень : навчальний посібник / Семеренко В. П. – Вінниця : ВНТУ, 2018. – 104 с.

5. Методика опанування навчальної дисципліни (освітнього компонента)

Матеріали для вивчення дисципліни розміщені викладачем в електронному вигляді на гугл-диску викладача, до якого надано доступ групі студентів та асистентам, які ведуть заняття комп'ютерного практикуму, а також, в середовищі Moodle платформи дистанційного навчання «Сікорський» (скорочена назва дисципліни uc80yt). Контент платформи доступний із будь-якого місця в мережі Інтернет зареєстрованому на курс студенту. Навчальний курс в середовищі Moodle складається з розділів, кожен з яких містить презентації лекцій, завдання та теоретичний матеріал для виконання комп'ютерного практикуму, URL-посилання на документацію з програмного забезпечення та навчальний матеріал для самостійної роботи відповідно до теми, яка вивчається.

В умовах дистанційного навчання усі види занять, у тому числі контрольні заходи, проводяться з використанням Zoom та Google Meet.

6. Самостійна робота студента/аспіранта

Матеріали для самостійного вивчення дисципліни розміщені викладачем в електронному вигляді в середовищі Moodle платформи дистанційного навчання «Сікорський». Контент платформи доступний із будь-якого місця в мережі Інтернет. До самостійної роботи студента відноситься, розробка програми у відповідності до завдання курсової роботи, робота з документацією та тьюторіалом програмного забезпечення, оформлення пояснювальної записки до курсової роботи, а також опрацювання додаткового теоретичного матеріалу та додаткової літератури. На самостійну роботу студент має витрати кількість годин, що дорівнює 1 кредиту.

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Студент має вивчати дисципліну протягом семестру, дотримуючись календарного плану виконання завдань комп'ютерного практикуму, вивчення тем лекційного матеріалу та виконання модульних контрольних робіт. Усі завдання студент має виконувати **самостійно і вчасно**. Завдання вважається виконаним, якщо студент захистив завдання комп'ютерного практикуму у викладача та розмістив звіт з виконання у відповідному розділі дисципліни у мудл платформи «Сікорський». Несвоєчасним вважається виконання завдання з затримкою більше 1 тижня і штрафується втратою 10% оцінки. Затримка у виконанні завдання більше ніж 4 тижні не допускається і можливість захистити завдання студентом втрачається назавжди. Такі обмеження надають можливість організувати систематичне виконання завдань студентами та не допустити значного накопичення незданих робіт на кінець семестру.

Модульні контролю проводяться за результатами виконання комп'ютерних практикумів та модульних контрольних робіт. Файли з виконаними завданнями розміщуються студенти у відповідному розділі дисципліни у мудл платформи «Сікорський».

Якщо студент не задоволений сумарною оцінкою за модульні контрольні роботи, він може переписати їх на семестровій контрольній роботі, що містить завдання з усього курсу.

Заохоченням до своєчасного засвоєння теоретичного матеріалу є бали за правильні відповіді на запитання під час експрес-опитування на лекціях.

Оцінювання студентів здійснюється згідно рейтингової оцінки рівня підготовки студентів з дисципліни. Поточний стан успішності студенти можуть бачити наприкінці кожного лекційного заняття в електронному журналі. Рейтингова система оцінювання з кредитного модуля описана у наступному розділі робочої програми.

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

Поточний контроль: *перевірка дотримання графіку виконання курсової роботи та якості виконаних частин завдання*

Календарний контроль: *провадиться двічі на семестр як моніторинг поточного стану виконання вимог робочої програми.*

Семестровий контроль: *залік*

Умови допуску до семестрового контролю: *50 балів зі 100 за результатами поточного контролю.*

Рейтинг студента з кредитного модуля складається з балів, які він отримує за результатами:

- 1) розробки програмного забезпечення у відповідності до індивідуального завдання;
- 2) оформлення пояснювальної записки до курсової роботи;
- 3) прилюдного захисту з презентацією результатів курсової роботи.

Семестрова 100-бальна оцінка за виконання курсової роботи складається з балів, отриманих за розробку програмного забезпечення у відповідності до індивідуального завдання та оформлення пояснювальної записки. Набрана кількість балів не може перевищувати максимальну кількість балів завдання обраної складності.

Розподіл балів між частинами завдання:

№ з/п	Назва розділу курсової роботи	Всього	Кількість балів по видах діяльності		
			Розробка програмного забезпечення	Оформлення пояснювальної записки	Захист
1	Вступ	1	-	1	
2	Розділ 1. Опис послідовного алгоритму та його відомих паралельних реалізацій	4	-	4	-
3	Розділ 2. Розробка послідовного алгоритму та аналіз його швидкодії	15	10	5	-
4	Розділ 3. Вибір програмного забезпечення для розробки паралельних обчислень та його короткий опис	5	-	5	-
5	Розділ 4. Розробка паралельного обчислення алгоритму з використанням обраного програмного забезпечення: проектування, реалізація, тестування.	35	30	5	-
6	Розділ 5. Дослідження ефективності паралельних обчислень алгоритму	25	20	5	-
7	Висновки	1	-	1	-
8	Список використаних джерел	2	-	2	-
9	Додатки	2	-	2	-
10	Захист курсової роботи	10	-	-	10
	Всього балів	100	60	30	10

Підсумкова оцінка переводиться до залікової оцінки згідно з таблицею:

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)

- Перелік тем, які пропонуються для виконання:

Алгоритм (назва вибраного алгоритму) та його паралельна реалізація (назва мови, технології чи програмного забезпечення)

Варіанти вибору алгоритму:

Рівень А (до 100 балів)

Алгоритм Дейкстри

Δ-ступінг алгоритм пошуку найкоротших шляхів

Алгоритм векторизації тексту на основі статистичного аналізу слів

Алгоритм контрольованого навчання нейромережі

Алгоритм неконтрольованого навчання нейромережі

Алгоритм розпізнавання об'єкта на графічному зображенні

Алгоритм аналізу колекції графічних зображень на співпадіння

Алгоритм сегментації зображення з використання нейромережі

Алгоритм аналізу колекції текстових фрагментів на співпадіння

Алгоритм перевірки тексту на плагіат

Алгоритм виявлення та вирішення колізій фізичних тіл

Мурашиний алгоритм

Бджолиний алгоритм

Алгоритм PageRank

Алгоритм Косараджу

Алгоритм пошуку компонент зв'язності

Багаторядний алгоритм самоорганізації моделей

Алгоритм імітації мережі масового обслуговування

Алгоритм імітації Петрі-об'єктного моделювання

Рівень В (до 84 балів)

Алгоритм Флойда-Воршелла

Алгоритм Беллмана-Форда

Алгоритм A*

Алгоритм D*

Алгоритм статистичного аналізу тексту

Алгоритм мінімакс

Еволюційний алгоритм пошуку оптимального значення

Алгоритм класифікації графічного зображення

Алгоритм виділення ознак графічного зображення

Алгоритм Крускала

Алгоритм Прима

Алгоритм Борувки

Алгоритм сортування масиву елементів об'єктного типу.

Алгоритм розв'язання задачі про пакування рюкзака методом динамічного програмування

Алгоритм пошуку в ширину

Алгоритм пошуку в глибину

Алгоритм пошуку оптимальних значень методом градієнтного спуску

Алгоритм розв'язання транспортної задачі методом потенціалів

Алгоритм коду Прюфера

Алгоритм відтворення життя популяції

Алгоритм обчислення послідовності Фібоначчі

Алгоритм пошуку розв'язку задачі лінійного програмування за симплекс-методом
Однорядний алгоритм самоорганізації моделей

Рівень С (до 74 балів)

Алгоритм бітонічного сортування
Алгоритм сортування комірками
Алгоритм парного-непарного сортування
Алгоритм пірамідального сортування
Алгоритм Фокса паралельного множення матриць
Алгоритм SUMMA (Scalable Universal Matrix Multiplication Algorithm) множення матриць
Алгоритм Кеннона паралельного множення матриць
Алгоритм розв'язання системи лінійних рівнянь методом Гауса
Алгоритм пошуку регресійного рівняння
Алгоритм пошуку оберненої матриці
Алгоритм передобробки графічного зображення
Алгоритм оптимального планування робіт

Варіанти вибору мов програмування: Java, C++, C#, Python, Go, Scala та інші.

Варіанти вибору програмного забезпечення: MPJExpress, OpenMPI, OpenMP, CUDA C++, PyCUDA та інші.

Робочу програму навчальної дисципліни (силабус):

Складено професор кафедри ІПІ, д.т.н., професор Стеценко Інна Вячеславівна, ст. викладач кафедри ІПІ, доктор філософії Дифучина О.Ю., ст. викладач кафедри ІПІ, доктор філософії Дифучин А.Ю.

Ухвалено кафедрою ІПІ (протокол № 16 від 23.06.2025р.)

Погоджено Методичною комісією факультету (протокол № 11 від 27.06.2025р.)