



Технології паралельних обчислень

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	
Форма навчання	<i>заочна</i>
Рік підготовки, семестр	<i>3 курс, весняний семестр</i>
Обсяг дисципліни	<i>5,5 кредитів (165 год)</i>
Семестровий контроль/ контрольні заходи	<i>Екзамен / Домашня контрольна робота</i>
Розклад занять	
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Лектор: ст. викл., Дифучина Олександра Юріївна, sashadif@gmail.com Лабораторні: ст. викл., Дифучина Олександра Юріївна
Розміщення курсу	Google Classroom https://classroom.google.com/c/NzQ2NTUwNjYxNDgw?cjc=tmdl3fb

Програма навчальної дисципліни

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

Вивчення дисципліни спрямовано на оволодіння технологіями паралельних обчислень (Java Multithreading, Java MPI). Увага приділяється розробці ефективних паралельних алгоритмів з використанням вказаних технологій. 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 швидкістю, вивчення сучасних технологій для реалізації паралельних обчислень. За результатами вивчення дисципліни студент повинен набути **досвід** з розробки паралельних алгоритмів та їх реалізації з використанням технологій паралельних обчислень Java Multithreading, Java MPI.

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 багатопоточні програми, використовуючи низько- та високо-рівневі засоби паралельних обчислень мови Java;
- виконувати дослідження ефективності розроблених паралельних програм експериментальними методами;
- розробляти алгоритми для виконання паралельних обчислень з використанням технології MPI.

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

КС 4 Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).

КС Х Здатність розробляти інформаційні управляючі системи з використанням розподілених та паралельних обчислень

Програмні результати вивчення дисципліни забезпечують такі програмні результати освітньої програми:

ПР 3 Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій.

ПР 6 Демонструвати знання сучасного рівня технологій інформаційних систем, практичні навички програмування та використання прикладних і спеціалізованих комп'ютерних систем та середовищ з метою їх запровадження у професійній діяльності.

ПР 7 Обґрунтовувати вибір технічної структури та розробляти відповідне програмне забезпечення, що входить до складу інформаційних систем та технологій.

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 «Об'єктно-орієнтоване програмування», «Теорія алгоритмів», «Веб-орієнтована розробка програмного забезпечення», «Комп'ютерні мережі», «Моделювання систем». Знання та навички, набуті студентом при вивченні дисципліни, використовуються в розробці дипломних проектів бакалавра за освітньою програмою «Інженерія програмного забезпечення інформаційних систем».

3. Зміст навчальної дисципліни

Розділ 1 Паралельні алгоритми та обчислення в багатоядерних системах.

Тема 1.1. Поняття паралельних, псевдопаралельних та розподілених обчислень.

- Огляд сучасних паралельних обчислювальних систем
- Класифікація паралельних обчислювальних систем.
- Два рівні розпаралелювання.
- Способи розпаралелювання.
- Завдання, розв'язувані суперкомп'ютерами.
- Способи підвищення продуктивності комп'ютерів.
- Проблеми використання багатопроцесорних систем.

Тема 1.2. Багатопоточна технологія Java. Створення та запуск потоків в Java.

- Організація обчислень у багатопроцесорних системах.

- Поняття потоку обчислень.
- Стани потоку та переходи потоку з одного стану в інший.
- Властивості потоку: пріоритети.
- Клас Thread.

Тема 1.3. Управління потоками в Java. Синхронізація. Блокований об'єкт.

- Призупинка та відновлення роботи потоку.
- Переривання дії потоку.
- Синхронізовані методи.
- Поняття блокованого об'єкта.
- Проблеми управління потоками: дедлок та інші небажані стани паралельної програми.

Тема 1.4. Високорівневі засоби управління потоками з використанням бібліотеки паралельних обчислень `java.util.concurrent`. Локери.

- Призначення високорівневих способів управління потоками.
- Локери: призначення, особливості використання.
- Приклади використання локерів при розробці паралельних програм.
- Приклади взаємного блокування потоків та подолання цього небажаного стану.

Тема 1.5. Високорівневі засоби управління потоками з використанням бібліотеки паралельних обчислень `java.util.concurrent`. Пули потоків.

- Призначення пулів потоків.
- Приклади використання пулів потоків при розробці паралельних програм.
- Переваги використання пулів потоків.

Тема 1.6. Високорівневі засоби управління потоками з використанням бібліотеки паралельних обчислень `java.util.concurrent`. Екзекутори.

- Призначення екзекуторів.
- Приклади використання екзекуторів при розробці паралельних програм.
- Переваги використання екзекуторів.

Тема 1.7 Розробка високоефективних паралельних програм з використанням `ForkJoinFramework`

- Призначення `ForkJoinFramework`.
- Приклади використання `ForkJoinFramework` при розробці паралельних програм.
- Переваги використання `ForkJoinFramework`.

Розділ 2 Паралельні обчислення в розподілених системах

Тема 2.1. Стандарт `Message Passing Interface (MPI)`.

- Призначення методів стандарту MPI один до одного.
- Основні методи стандарту MPI один до одного.
- Застосування методів обміну повідомлення один до одного для розробки паралельних програм.

Тема 2.2. Колективні методи обміну повідомленнями стандарту MPI.

- Призначення колективних методів стандарту MPI.
- Основні колективні методи стандарту MPI.
- Застосування колективних методів обміну повідомлення для розробки паралельних програм.

Тема 2.3. Технологія MPI

- `MPJ Express` — реалізація стандарту MPI.
- Побудова MPI програми, особливості її виконання в розподіленій системі.
- Методи MPI.

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.
4. Центр суперкомп'ютерних обчислень НТУУ «Київський політехнічний інститут імені Ігоря Сікорського» [Електронний ресурс] – Режим доступу: http://kpi.ua/web_hpcc

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Матеріали для вивчення дисципліни розміщені викладачем в електронному вигляді в Google classroom (посилання на <https://classroom.google.com/c/NzQ2NTUwNjYxNDgw?cjc=tmdl3fb> , код курсу tmdl3fb).

Лекції по дисципліні проводяться викладачем із використанням сучасних мультимедійних презентаційних технологій.

Лабораторні заняття (Комп'ютерні практикуми) виконуються з використанням платформи Java (SE та EE). Під час їх проведення використовуються методичні вказівки до комп'ютерних практикумів по дисципліні. Завдання кожного практикуму містить кілька підзавдань різної складності, які мають окреме оцінювання. Таке розбиття надає можливість більш об'єктивно оцінити рівень умінь студента і, водночас, адаптувати завдання до рівня знань та навичок студента.

Домашня контрольна робота містить завдання з перевірки як теоретичних знань, так і практичних навичок.

6. Самостійна робота студента/аспіранта

Матеріали для вивчення дисципліни розміщені викладачем в електронному вигляді в Google classroom (посилання <https://classroom.google.com/c/NzQ2NTUwNjYxNDgw?cjc=tmdl3fb> , код курсу tmdl3fb). Контент платформи доступний із будь-якого місця в мережі Інтернет. До самостійної роботи студента відноситься, в основному, виконання завдання комп'ютерного практикуму, робота з документацією програмного забезпечення, а також опрацювання лекційного та додаткового теоретичного матеріалу за наданими презентаціями лекцій, навчальним посібником та додатковою літературою.

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 модульної контрольної роботи. 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 завданні у Google Classroom.

Модульні контролі проводяться за результатами виконання комп'ютерних практикумів та модульної контрольної роботи. Файли з виконаними завданнями розміщуються студенти у відповідному завданні у Google Classroom.

Оцінювання студентів здійснюється згідно рейтингової оцінки рівня підготовки студентів з дисципліни. Рейтингова система оцінювання з кредитного модуля описана у наступному розділі робочої програми.

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

Поточний контроль: ДКР, завдання комп'ютерного практикуму

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

Семестровий контроль: залік.

Умови допуску до семестрового контролю: 60 балів зі 100 можливих за захисти завдань комп'ютерного практикуму та семестровий рейтинг більше 50 балів (зі 100 можливих).

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

- 1) виконання модульної контрольної роботи;
- 2) виконання лабораторних комп'ютерних практикумів;
- 3) відповідей на залікові питання.

Система рейтингових балів

Усі завдання студента оцінюються за 100-бальною шкалою. Оцінки за завдання враховуються в семестрову оцінку студента з ваговим коефіцієнтом.

Теоретичні знання студента оцінюють за результатами письмової модульної контрольної роботи (Т). Практичні навички студента оцінюються за результатами захисту виконаних завдань комп'ютерного практикуму. Сумарна оцінка за виконання завдань комп'ютерного практикуму визначається за формулою:

$$P = 1/4 \sum D_i,$$

де D_i – оцінка за i -тий комп'ютерний практикум.

Зауваження: необхідною умовою допуску до заліку є кількість балів, набраних за виконання завдань комп'ютерного практикуму, не менша за 60 (із 100 можливих). Тобто, якщо студент не набрав 60 балів за виконання завдань комп'ютерного практикуму ($35 \leq P < 60$), він отримує підсумкову оцінку «Незадовільно». Якщо студент за виконання завдань комп'ютерного практикуму протягом семестру отримав менше 35 балів, він отримує підсумкову оцінку «Не допущено».

Сума балів, набраних студентом протягом семестру, складається з сумарної оцінки за комп'ютерний практикум за формулою за умови, що набрана кількість балів за виконання завдання комп'ютерного практикуму не менша за 60 (із 100 можливих):

$$Z = 0,5 \cdot P + 0,5 \cdot T, \text{ якщо } P \geq 60,$$

$$Z = P, \text{ якщо } P < 60$$

де P – оцінка практичних навичок студента, T – оцінка його теоретичних знань.

Підсумкова оцінка формується за результатами оцінювання знань та навичок студента в семестрі:

$$S = Z$$

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 питань, які виносяться на семестровий контроль

1. Поняття паралельних, псевдопаралельних та розподілених обчислень.
2. Класифікація паралельних обчислювальних систем.
3. Способи розпаралелювання.
4. Системи з загальною та розподіленою пам'яттю.
5. Способи підвищення продуктивності комп'ютерів. Суперкомп'ютери.

6. Багатопоточна технологія Java.
7. Поняття потоку обчислень. Створення та запуск потоків в Java.
8. Стани потоку та переходи потоку з одного стану в інший.
9. Клас Thread.
10. Призупинка та відновлення роботи потоку. Переривання дії потоку.
11. Управління потоками в Java.
12. Синхронізація в паралельних обчисленнях.
13. Блокування потоку.
14. Блокування об'єкту.
15. Синхронізовані методи.
16. Проблеми управління потоками: дедлок та інші небажані стани паралельної програми.
17. Високорівневі способи управління потоками з використанням бібліотеки паралельних обчислень `java.util.concurrent`.
18. Локери та управління потоками.
19. Пули потоків.
20. Інтерфейс `Executor` та бібліотечні класи, які його реалізують.
21. Розробка високоефективних паралельних програм з використанням `ForkJoinFramework`.
22. Показники ефективності паралельного алгоритму.
23. Алгоритми паралельного сумування та оцінка їх ефективності.
24. Закон Амдала. Ефект Амдала.
25. Експериментальне дослідження ефективності паралельних обчислень.
26. Оцінка ефективності паралельних обчислень.
27. Проектування паралельних програм.
28. Стандарти та технології обміну повідомленням в розподілених системах.
29. Стандарт `Message Passing Interface (MPI)`.
30. Методи обміну повідомленнями стандарту `MPI` один до одного та їх застосування для розробки паралельних програм.
31. Колективні методи обміну повідомленнями стандарту `MPI` та їх застосування колективних для розробки паралельних програм.
32. `OpenMPI`: особливості побудови програми та основні методи..
33. Розробка ефективних паралельних алгоритмів в `OpenMPI`.
34. Алгоритми паралельного сортування.
35. Оцінювання продуктивності паралельних алгоритмів в розподілених системах.

Робочу програму навчальної дисципліни (силабус):

Складено професор кафедри ІПІ, д.т.н., професор Стеценко Інна Вячеславівна, ст. викладач кафедри ІПІ, доктор філософії Дифучина Олександра Юріївна

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

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