



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	<i>Нормативна</i>
Форма навчання	<i>очна(денна)</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 / Модульні контрольні роботи</i>
Розклад занять	
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Лектор: <i>д.т.н., професор, Стеценко Інна Вячеславівна, inna.stetsenko-fiot@ill.kpi.ua</i> Практичні / Семінарські: <i>ст.викл., Ph.D. Дифучин Антон Юрійович, difuchin@gmail.com,</i> <i>ст.викл., Ph.D Дифучина Олександра Юріївна, sashadif@gmail.com,</i> <i>ас. Нестеренко Костянтин Павлович, k.nesterenko@kpi.ua</i> Лабораторні: <i>ст.викл., Ph.D. Дифучин Антон Юрійович, difuchin@gmail.com,</i> <i>ст.викл., Ph.D Дифучина Олександра Юріївна, sashadif@gmail.com,</i> <i>ас. Нестеренко Костянтин Павлович, k.nesterenko@kpi.ua</i>
Розміщення курсу	<i>Moodle: Технології паралельних та розподілених обчислень (uc80yt)</i> <i>https://do.ipk.kpi.ua/course/view.php?id=2504</i>

Програма навчальної дисципліни

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

Вивчення дисципліни спрямовано на оволодіння технологіями паралельних обчислень (Java Multithreading, CUDA C++, OpenMPI, MPJ Express, Java Sockets, Java RPC). 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 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 Sockets, Java RMI, Java IDL.

Програмні результати навчання студента.

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

- базові принципи паралельних та розподілених обчислень,
- архітектуру багатоядерних та багатопроцесорних обчислювальних систем;
- методи розробки паралельних алгоритмів для прикладних задач;
- технології багатопотокового програмування;
- технологію Message Passing Interface (MPI);
- застосування розподілених систем у різних областях;
- методи оцінювання ефективності паралельних та розподілених 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**:

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ФК17. Здатність розробляти інформаційні системи з використанням паралельних обчислень.

Програмні результати вивчення дисципліни забезпечують такі програмні результати освітньої програми:

ПРН 15. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки 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», «Проектування 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

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. Алгоритми паралельного множення матриць.

- Стрічковий алгоритм 1.
- Стрічковий алгоритм 2.
- Алгоритм Форда.

Тема 1.4. 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 1.5. 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 1.6. 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 1.7. 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 1.8. Розробка високоефективних паралельних програм з використанням `ForkJoinFramework`

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

Тема 1.9. Моделювання паралельних обчислень

- Модель обчислень у вигляді графа «операції—операнди».
- Моделювання паралельних обчислень на прикладі паралельного сумування масиву значень.
- Показники ефективності паралельного алгоритму.
- Закон Амдала.
- Ефект Амдала.

Тема 1.10. Проектування паралельних програм.

- Шаблони проектування.
- Моделювання паралельних програм мережею Петрі.

Розділ 2 Паралельні обчислення в розподілених системах

Тема 2.1. Архітектура розподілених систем.

- Таксономія Фліна.
- Архітектура пам'яті паралельних обчислень: колективна та розподілена пам'ять.
- Моделі пам'яті паралельних обчислень: модель колективної пам'яті, модель потоків, модель розподіленої пам'яті, паралельна модель даних, гібридна модель, `SPMD` та `MPMP`.

Тема 2.2. Обчислення в розподілених системах.

- Обчислення в розподілених системах.
- Стандарти та технології обміну повідомленням в розподілених системах.
- Стандарт `Message Passing Interface (MPI)`.

Тема 2.3. Методи обміну повідомленнями стандарту MPI один до одного.

- 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 2.4. 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 2.5. Технологія MPI

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

Тема 2.6. Розробка ефективних паралельних алгоритмів в MPI.

- Алгоритми паралельного сортування.
- Алгоритми паралельного множення матриць.
- Паралельна реалізація алгоритму МГУА.
- Оцінювання продуктивності паралельних алгоритмів в розподілених системах.

Тема 2.7. Технологія Remote Method Invocation (RMI) та інші прогресивні технології розподілених обчислень.

- Клієнт-серверна модель розподілених обчислень.
- Модель розподілених об'єктів та технологія RMI.
- Розробка розподілених додатків з використанням RMI.
- Архітектура CORBA та розробка CORBA-застосувань з використанням Java IDL.

Тема 2.8. Grid технології.

- Базові складові Grid та ресурси.
- Зв'язок Grid і веб-технологій.
- Програмне забезпечення Grid.
- Організація і управління розподіленими грід-ресурсами.
- Grid і бази даних.
- Безпека файлової системи.
- Сертифікат відкритих ключів.
- Система підтримки функціонування: послуга протоколювання процесу виконання завдань.

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.
3. Ukrainian National Grid [Електронний ресурс] – Режим доступу:<http://grid.kpi.ua/>

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 середовищі Moodle платформи дистанційного навчання «Сікорський» (скорочена назва дисципліни uc80yt, сертифікат ДК № 0098, затверджений Методичною радою КПІ ім. Ігоря Сікорського, протокол № 8 від 02.06.2023 р.). Контент платформи доступний із будь-якого місця в мережі Інтернет зареєстрованому на курс студенту. Навчальний курс в середовищі Moodle складається з розділів, кожний 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.

Лекції по дисципліні проводяться викладачем із використанням сучасних мультимедійних презентаційних технологій. На лекціях проводяться експрес-опитування, які дають можливість лектору отримати інформацію про якість засвоєння матеріалу та, за необхідності, розглянути більш детально складний матеріал. Обговорення лекційного матеріалу відбувається також в чаті дисципліни в середовищі Moodle.

Практичні заняття спрямовані на оволодіння навичками проектування паралельної програми, моделювання паралельної програми та проведення експериментального дослідження ефективності паралельної програми.

Перелік практичних занять:

Практичне заняття 1 «Проектування паралельної програми та її моделювання з метою досягнення найбільшої ефективності»

Практичне заняття 2 «Проведення експериментального дослідження ефективності паралельної програми»

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. Для виконання кожного завдання комп'ютерного практикуму студенту надається 2 тижні. Студент на початку семестру отримує рекомендований календарний план початку та завершення захисту завдань комп'ютерного практикуму. Захистити та доопрацювати завдання студент може протягом семестру.

Перелік завдань комп'ютерного практикуму:

Завдання до комп'ютерного практикуму 1 «Розробка потоків та дослідження пріоритету запуску потоків» (тема 1.2)

Завдання до комп'ютерного практикуму 2 «Розробка паралельних алгоритмів множення матриць та дослідження їх ефективності» (тема 1.3)

Завдання до комп'ютерного практикуму 3 «Розробка паралельних програм з використанням механізмів синхронізації: синхронізовані методи, локери, спеціальні типи» (теми 1.4, 1.5)

Завдання до комп'ютерного практикуму 4 «Розробка паралельних програм з використанням пулів потоків, екзекуторів та ForkJoinFramework» (теми 1.6-1.8)

Завдання до комп'ютерного практикуму 5 «Застосування високорівневих засобів паралельного програмування для побудови алгоритмів імітації та дослідження їх ефективності» (теми 1.6-1.8)

Завдання до комп'ютерного практикуму 6 «Розробка паралельного алгоритму множення матриць з використанням MPI-методів обміну повідомленнями «один-до-одного» та дослідження його ефективності». (тема 2.3)

Завдання до комп'ютерного практикуму 7 «Розробка паралельного алгоритму множення матриць з використанням MPI-методів колективного обміну повідомленнями («один-до-багатьох», «багато-до-одного», «багато-до-багатьох») та дослідження його ефективності» (тема 2.4).

Завдання до комп'ютерного практикуму 8 «Розробка паралельного алгоритму для клієнт-серверній архітектури з використанням сокетів» (тема 2.7)

Модульні контролі проводяться за результатами виконання комп'ютерних практикумів та модульних контрольних робіт. Модульні контрольні роботи містять завдання з перевірки як теоретичних знань, так і практичних навичок.

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 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». 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 та тьюторіалом програмного забезпечення, а також опрацювання лекційного та додаткового теоретичного матеріалу за наданими презентаціями лекцій, навчальним посібником та додатковою літературою. На самостійну роботу студент має витрати кількість годин, що дорівнює кількості годин, проведених ним на аудиторних заняттях.

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 2 тижнів 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 контролі проводяться за результатами виконання комп'ютерних практикумів та модульних контрольних робіт. Файли з виконаними завданнями розміщуються студентами у відповідному розділі дисципліни у мудл платформи «Сікорський».

Дві модульні контрольні роботи проводяться в семестрі з використанням середовища Moodle платформи дистанційного навчання «Сікорський». Завдання та запитання модульної контрольної роботи – це тренування студента до запитань та завдань екзаменаційного білету. Дату та час проведення робіт оголошує лектор на початку семестру, відмічає її у журналі та нагадує про проведення роботи на лекційних заняттях. Переписати модульну контрольну роботу неможливо.

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: *експрес-опитування, МКР, завдання комп'ютерного практикуму*

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

Семестровий контроль: *екзамен*

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 модульних контрольних робіт;
- 2) виконання лабораторних комп'ютерних практикумів;
- 3) відповідей на екзаменаційні питання;
- 4) виконання практичного завдання екзаменаційного білету.

Враховуються також додаткові бали за правильні відповіді на запитання під час лекційних занять.

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 оцінюються за результатами письмової модульної контрольної роботи та, частково, за результатами опитування на лекційних заняттях. Сумарна оцінка визначається за формулою:

$$T=0,5 \cdot K1+0,5 \cdot K2+A,$$

де K1 – оцінка за першу модульну контрольну роботу, K2 – оцінка за другу модульну контрольну роботу, A – додаткові бали за результатами опитування на лекційних заняттях.

Практичні навички студента оцінюються за результатами захисту виконаних завдань комп'ютерного практикуму. На виконання завдання комп'ютерного практикуму надається 2 тижні. Якщо студент здає завдання на третьому тижні з моменту видачі завдання, то завдання вважається зданим з затримкою і оцінка за виконання комп'ютерного практикуму знижується на 10%. Якщо студент не здав завдання у 3-тижневий термін, це завдання вважається студентом невиконаним з оцінкою 0 балів без права перезахисту. Звіт з виконання завдання комп'ютерного практикуму має бути поданий у дистанційний курс на платформі «Сікорський», де він зберігається до кінця навчального року. Захист практикуму відбувається на заняттях з комп'ютерного практикуму, які проводяться онлайн за розкладом. Сумарна оцінка за виконання завдань комп'ютерного практикуму визначається за формулою:

$$P=1/8 \sum D_i,$$

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

Зауваження: в умовах дистанційного навчання, якщо студент завантажив файли з виконанням завданням у мудл платформи «Сікорський» у 3-тижневий термін, то йому надається право захисту цього завдання до кінця семестру.

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.

Екзаменаційна оцінка складається з оцінки за відповіді на два теоретичних питання та два практичних завдання за формулою:

$$E = 0,3 \cdot P1 + 0,3P2 + 0,2 \cdot T1 + 0,2 \cdot T2$$

де P1, P2 – оцінки виконання першого та другого практичного завдання, T1, T2 – оцінки відповідей на перше та друге теоретичні питання відповідно.

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

$$S = 0,5 \cdot Z + 0,5 \cdot E$$

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. Алгоритми паралельного множення матриць.
11. Стрічковий алгоритм паралельного множення матриць.
12. Алгоритм Фокса паралельного множення матриць.
13. Алгоритм Кеннона паралельного множення матриць.
14. Призупинка та відновлення роботи потоку. Переривання дії потоку.
15. Управління потоками в Java.
16. Синхронізація в паралельних обчисленнях.
17. Блокування потоку.
18. Блокування об'єкту.
19. Синхронізовані методи.
20. Проблеми управління потоками: дедлок та інші небажані стани паралельної програми.
21. Високорівневі способи управління потоками з використанням бібліотеки паралельних обчислень `java.util.concurrent`.
22. Локери та управління потоками.
23. Пули потоків.
24. Інтерфейс `Executor` та бібліотечні класи, які його реалізують.
25. Розробка високоефективних паралельних програм з використанням `ForkJoinFramework`.
26. Методи моделювання паралельних обчислень.
27. Модель обчислень у вигляді графа «операції—операнди».
28. Моделювання паралельних програм мережею Петрі.
29. Показники ефективності паралельного алгоритму.
30. Алгоритми паралельного сумування та оцінка їх ефективності.
31. Закон Амдала. Ефект Амдала.

32. Експериментальне дослідження ефективності паралельних обчислень.
 33. Оцінка ефективності паралельних обчислень.
 34. Проектування паралельних програм.
 35. Архітектура розподілених систем. Таксономія Фліна.
 36. Моделі пам'яті паралельних обчислень.
 37. Стандарти та технології обміну повідомленням в розподілених системах.
 38. Стандарт Message Passing Interface (MPI).
 39. Методи обміну повідомленнями стандарту MPI один до одного та їх застосування для розробки паралельних програм.
 40. Колективні методи обміну повідомленнями стандарту MPI та їх застосування колективних для розробки паралельних програм.
 41. OpenMPI: особливості побудови програми та основні методи..
 42. Розробка ефективних паралельних алгоритмів в OpenMPI.
 43. Алгоритми паралельного сортування.
 44. Паралельна реалізація алгоритму МГУА (метод групового урахування аргументів).
 45. Оцінювання продуктивності паралельних алгоритмів в розподілених системах.
 46. Технологія Remote Method Invocation (RMI).
 47. Грід-технології. Поняття спільного віртуального ресурсу.
 48. Базові складові грід-системи.
 49. Програмне забезпечення грід-технологій.
 50. Організація і управління розподіленими грід-ресурсами.
- *перелік програмного забезпечення, оволодіння яким може бути зараховане як бонусні бали за умови доповіді 7-10 хвилин на останньому лекційному занятті: OpenMP, Go, Scala, Hadoop.*

Робочу програму навчальної дисципліни (силабус):

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

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

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