

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Львівський національний університет імені Івана Франка**  
**Факультет прикладної математики та інформатики**  
**Кафедра програмування**

**Затверджено**

На засіданні кафедри програмування  
факультету прикладної математики  
Львівського національного університету  
імені Івана Франка  
(протокол № 1 від 29 серпня 2025 р.)



Зав. кафедри к. ф.-м. н., доц. Ярошко С. А.

**Силабус навчальної дисципліни**  
**«Паралельні та розподілені обчислення»,**  
**що викладається в межах ОПП “Середня освіта (Інформатика)”**  
**першого (бакалаврського) рівня вищої освіти**  
**для здобувачів зі спеціальності А4.09 Середня освіта (Інформатика)**

Львів 2025 р.

<b>Назва дисципліни</b>	Паралельні та розподілені обчислення
<b>Адреса викладання дисципліни</b>	Львівський національний університет імені Івана Франка, вул. Університетська 1, м. Львів, Україна, 79000
<b>Факультет та кафедра, за якою закріплена дисципліна</b>	Факультет прикладної математики та інформатики, кафедра програмування
<b>Галузь знань, шифр та назва спеціальності</b>	Галузь знань: А Освіта / Педагогіка Спеціальність: А4.09 Середня освіта (Інформатика)
<b>Викладачі дисципліни</b>	Гошко Богдан Мирославович, к. ф.-м. н., доцент, доцент кафедри програмування; Пасічник Тимофій Васильович, к. ф.-м. н., доцент, доцент кафедри програмування
<b>Контактна інформація викладачів</b>	Електронна пошта: <a href="mailto:bohdan.hoshko@lnu.edu.ua">bohdan.hoshko@lnu.edu.ua</a> , <a href="mailto:tymofiy.pasichnyk@lnu.edu.ua">tymofiy.pasichnyk@lnu.edu.ua</a> веб-сторінки: <a href="https://ami.lnu.edu.ua/employee/hoshko">https://ami.lnu.edu.ua/employee/hoshko</a> <a href="https://ami.lnu.edu.ua/employee/pasichnyk">https://ami.lnu.edu.ua/employee/pasichnyk</a>
<b>Консультації з питань навчання по дисципліні відбуваються</b>	Консультації проводять раз на тиждень згідно з оприлюдненим розкладом консультацій викладача. Можливі он-лайн консультації через Zoom чи Microsoft Teams. Для погодження часу он-лайн консультацій слід писати на електронну пошту викладача.
<b>Сторінка курсу</b>	<a href="https://ami.lnu.edu.ua/course/paralelni-ta-rozpodileni-obchyslennia-so">https://ami.lnu.edu.ua/course/paralelni-ta-rozpodileni-obchyslennia-so</a>
<b>Інформація про дисципліну</b>	Курс «Паралельні та розподілені обчислення» є вибірковою дисципліною, яку викладають у 3-му семестрі в обсязі 4 кредитів (за Європейською кредитно-трансферною системою ECTS)
<b>Коротка анотація дисципліни</b>	Курс «Паралельні та розподілені обчислення» призначений для набуття теоретичних і практичних знань в області паралельних та розподілених обчислень, оволодіння концепціями сучасного програмування в рамках парадигм паралельного та розподіленого програмування. Основу вивчення курсу складають підходи до програмування в багатопотокових системах, розподілених системах, системах синхронних паралельних розрахунків, розглядаються проблеми сумісної роботи процесів паралельної програми та їх синхронізації. Реалізація синхронних або асинхронних паралельних процесів з використанням бібліотек MPI / OpenMP, стандартними засобами мов програмування C++, C# для розподілених обчислень.
<b>Мета та цілі дисципліни</b>	Метою нормативної дисципліни «Паралельні та розподілені обчислення» є формування необхідного рівня теоретичної і практичної підготовки студентів для грамотного використання ними знань основ паралельних та розподілених обчислень при освоєнні суміжних дисциплін та у майбутній професійній діяльності, а також формування знань для організації ефективного розв'язку великих задач на комп'ютерах з паралельною архітектурою.
<b>Література для вивчення дисципліни</b>	<ol style="list-style-type: none"> <li>1. Основна література</li> <li>2. Аксак Н.Г. Паралельні та розподілені обчислення : підруч./ Н. Г. Аксак, О. Г. Руденко, А. М. Гуржій. – Х. :Компанія СМІТ, 2009. – 480с.</li> <li>3. Robey R., Zamora Y. Parallel and High Performance Computing., Manning Publications, 2021. – 704 p.</li> <li>4. Introduction to Parallel Computing. From Algorithms to Programming on State-of-the-Art Platforms. / Roman Trobec , Boštjan Slivnik, Patricio Bulić, Borut Robič / Springer Books, 2018. – 256 p.</li> <li>5. J. Sanders, E. Kandrot. CUDA by Example. An Introduction to General-Purpose GPU Programming. Addison-Wesley - Boston. July 2010.</li> <li>6. R. Hochberg. Matrix Multiplication with CUDA. A basic introduction to the CUDA programming model. Cambridge University Press, August 11, 2012.</li> </ol>

	<p>7. David B. Kirk, Wen-mei Hwu. Programming Massively Parallel Processors. A Hands-on Approach. - Published by Elsevier Inc. - 2010. Допоміжна література</p> <p>8. Качко О.Г. Паралельне програмування. – Харків. нац. ун-т радіоелектроніки. – Харків : ХНУРЕ, 2016. – 403 с.</p> <p>9. Жуков І., Корочкін О. Паралельні та розподілені обчислення – К.:Корнійчук, 2005. – 226 с.</p>
<b>Обсяг курсу</b>	4 кредити ЄКТС – 120 годин. З них 32 години лекцій, 32 години лабораторних занять та 56 годин самостійної роботи
<b>Очікувані результати навчання</b>	<p>Після завершення цього курсу студент буде:</p> <p><i>знати</i></p> <ul style="list-style-type: none"> <li>● засоби та можливості середовища програмування, зокрема, щодо паралельних обчислень з використанням багатьох потоків</li> <li>● засади паралельного програмування з використанням обчислювальних можливостей графічних процесорів;</li> <li>● призначення та влаштування базових структур даних і алгоритмів для опрацювання даних з технологіями CUDA та OpenCL.</li> </ul> <p><i>вміти</i></p> <ul style="list-style-type: none"> <li>● писати та налагоджувати ефективні комп'ютерні програми мовою C++ та іншими мовами програмування з використанням паралельних обчислень для підвищення швидкодії відомих алгоритмів опрацювання даних;</li> <li>● оголошувати та використовувати масиви потоків чи завдань(Thread, Task);</li> <li>● використовувати та доповнювати засоби бібліотеки STL;</li> <li>● реалізовувати алгоритми розподілених обчислень на обчислювальному кластері ЛНУ.</li> </ul>
<b>Компетентності</b>	<p><i>Інтегральна:</i> Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачають застосування теорій та методів інформаційних технологій і характеризуються комплексністю та невизначеністю умов.</p> <p><i>Загальні (ЗК):</i></p> <p>ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.  ЗК2. Здатність застосовувати знання у практичних ситуаціях.  ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.  ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.  ЗК8. Здатність генерувати нові ідеї (креативність).  ЗК9. Здатність працювати в команді.  ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.</p> <p><i>Спеціальні (фахові, предметні) компетентності (СК):</i></p> <p>СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.  СК7. Здатність застосовувати теоретичні та практичні основи методології та технології моделювання для дослідження характеристик і поведінки складних об'єктів і систем, проводити обчислювальні експерименти з обробкою й аналізом результатів.  СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням паралельного та розподіленого програмування: з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.  СК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.</p>
<b>Програмні результати навчання</b>	<p>ПР1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.</p>

	<p>PR5. Проектувати, розробляти та аналізувати паралельні алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.</p> <p>PR9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.</p> <p>PR11. Розробляти програми з використанням технології CUDA та OpenCL для програмування з використанням можливостей графічних процесорів.</p> <p>PR15. Застосовувати знання технології MPI/OpenMP - засобів проведення складних розподілених обчислень, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.</p>		
<b>Ключові слова</b>	Паралельні обчислення, розподілені обчислення, кластер, паракомп'ютер, програма мовою C++, прискорення та ефективність паралельного алгоритму, ґрид, мультипроцесор, мультикомп'ютер, закон Амдалла.		
<b>Формат курсу</b>	Очний: проведення лекцій, лабораторних робіт та консультацій в приміщеннях університету, а в умовах карантину – онлайнний на платформі Microsoft Teams		
<b>Теми</b>		<b>Лекції</b>	
	Тиж-день	Номер, назва і зміст теми	К-сть год.
	1	Мета, задачі та цілі паралельної обробки даних	2
	2	Принципи побудови паралельних структур обчислювальних систем. Способи розпаралелювання та організація обчислень у багатопроцесорних системах.	2
	3	Приклади паралельних обчислювальних систем. Суперкомп'ютери. Кластери. Класифікація обчислювальних систем. Мультипроцесори. Мультикомп'ютери	2
	4	Закон Амдала. Мережний закон Амдала. Класифікація багатопроцесорних систем. Класи багатопроцесорних систем. Технічна реалізація багато-процесорних систем. Системи з масовим паралелізмом.	2
	5	Моделювання паралельних алгоритмів Алгоритм і його представлення. Форми запису алгоритму. Граф алгоритму. Топологічне сортування. Модель обчислень у вигляді графа «операції—операнди». Модель паралельного алгоритму. Показники ефективності паралельного алгоритму. Модель паралельного програмування. Розробка паралельних алгоритмів. Проектування комунікацій.	2
	6	Оцінка комунікаційної трудомісткості паралельних алгоритмів	2
	7	Загальна характеристика механізмів передачі даних	2
	8	. Паралельні алгоритми сортування. Обробка графів. Паралельне узагальнення базової операції сортування. Бульбашкове сортування. Сортування Шелла. Швидке сортування. Знаходження мінімально охоплюючого дерева. Пошук найкоротших шляхів. Класи задач, які можуть бути ефективно векторизовані або розпаралелені	2

9	Паралельні чисельні алгоритми Обчислення сум послідовності числових значень. Множення матриці на вектор. Матричне множення. Паралельні методи розв'язання СЛАР Макроопераційний аналіз алгоритмів розв'язання задач. Організація паралелізму на основі розподілу даних.	2	
10	Паралельні та мережні технології розв'язання задач лінійного програмування. Основи методу. Метод прямого перебору при розв'язанні задач лінійного програмування. Паралельний аналог «симплекс-методу». Паралельне розв'язання задачі цілочисельного лінійного програмування.	2	
11	Паралельні методи розрахунку транспортної мережі Прямий перебір та аналог «симплекс-методу» для розв'язку транспортної задачі без обмежень пропускної здатності комунікацій. Транспортна задача з обмеженнями пропускної здатності комунікацій. Паралельний алгоритм знаходження максимального потоку в мережі.	2	
12	Паралельні обчислювальні методи Паралельні перетворення математичних виразів. Базові алгоритми паралельних обчислень. Паралельні методи розв'язання систем нелінійних рівнянь. Паралельні методи розв'язання лінійних і диференційних рівнянь. Паралельна реалізація методу скінченних різниць	2	
13	Технології паралельного й розподіленого програмування. Кластерне програмування PVM Історія PVM. Класичні моделі паралелізму, підтримувані системою PVM. Застосування PVM. Бібліотека PVM для мови C++. Компіляція, компонування C++/PVM-програм і її виконання. Вимоги до PVM-програм. Об'єднання динамічної C++-бібліотеки з бібліотекою PVM. Базові механізми PVM. Доступ до стандартного вхідного потоку (stdin) і стандартного вихідного потоку (stdout) з боку PVM-задач.	2	
14	Технологія MPI MPICH — основна реалізація MPI. Конфігурування й засоби запуску в MPICH. Налаштування проекту при роботі з середовищем MPICH.	2	
15	Бібліотека MPE Загальна організація MPI. Похідні типи даних MPI. Запуск та налаштування середовища MPICH.	2	
16	OpenMP: зручний спосіб для розпаралелювання додатків. Функції бібліотеки OpenMP Змінні оточення OpenMP. Компіляція. Налаштування. Продуктивність. Ключові точки. Приклади. Розпаралелювання циклів. Мінімізація додаткових часових витрат на розпаралелювання. Програмування, орієнтоване на продуктивність.	2	
<b>Всього</b> :		32	
<b>Лабораторні роботи</b>			
Номер, назва і зміст теми		К- сть год	ТЗН
1. Способи розпаралелювання та організація обчислень.		2	ПК
2. Керування об'єктами у C#.		2	ПК
3. Багатопотокове програмування у .NET. Керування потоками у .NET		2	ПК
4. Паралельні алгоритми сортування. Бульбашкове сортування, сортування Шелла, швидке сортування.		4	ПК

	5. Паралельне програмування роботи з матрицями. Множення матриці на вектор. Матричне множення.	4	ПК
	6. Паралельна реалізація алгоритмів розв'язання СЛАР	2	ПК
	7. Паралельне розв'язання задачі лінійного програмування. Паралельний аналог симплекс методу. Використання Windows Forms	4	ПК
	8. Розв'язання задач на графах(Дейкстра, Флойда, Пріма) паралельними методами.	2	ПК
	9. Основи кластерного програмування. Використання технології MPI.	2	ПК
	10. Програмування на графічних процесорах за технологіями CUDA таOpenCL	2	ПК
	11. Організація передачі/прийому повідомлень між окремими процесами засобами MPI. Колективні операції між процесами у MPI.	2	ПК
	12. Зв'язок Grid і веб-технологій. Доступ до Grid засобами веб. Розроблення Grid-застосувань.	4	ПК
	Всього	32	
<b>Підсумковий контроль, форма</b>	залік в кінці семестру		
<b>Пререквізити</b>	Для вивчення курсу студенти потребують базових знань з дисциплін “Програмування”, “Алгоритми та структури даних”, “Архітектура обчислювальних систем”.		
<b>Навчальні методи та техніки, які використовують під час викладання курсу</b>	Лекції з мультимедійними презентаціями та з демонстрацією прийомів практичного використання середовища програмування; лабораторні заняття у вигляді проектування алгоритмів і програм, виконання практичних завдань, у тому числі у команді з 3-4 осіб; самостійне опрацювання навчальних матеріалів: підручників, конспектів лекцій, додаткових навчальних посібників, розміщених у хмарному сховищі (Moodle, Microsoft Teams). Обговорення теоретичного та практичного матеріалу в онлайн сервісах, формулювання творчих завдань для студентів, виконання яких готує до вивчення нового теоретичного матеріалу.		
<b>Необхідне обладнання</b>	Для проведення лекцій: комп'ютер, проектор, доступ до мережі інтернет. Для проведення лабораторних та виконання завдань: комп'ютер, ОС Windows/Linux, доступ до інтернету, середовище програмування мовою С++ (Microsoft Visual Studio, Code Blocks тощо), доступ до кластера ЛНУ з відповідним програмним забезпеченням. Уся література, яку студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.		
<b>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</b>	<p><b>Оцінювання</b> проводиться за 100-бальною шкалою. 60 балів нараховують за виконання лабораторних завдань і контрольних робіт, ще 40 балів – за виконання теоретичних завдань. Лабораторні завдання можуть бути індивідуальні та командні. Упродовж семестру студент виконує 10 лабораторних робіт, кожен з яких оцінюють 6 балів. Додаткові бали будуть зараховані учасникам Всеукраїнської студентської олімпіади з програмування (відповідно до кількості розв'язаних задач).</p> <p>Вчасно виконані завдання оцінюють так (у відсотках від максимальної оцінки):</p> <ul style="list-style-type: none"> <li>• 100% – умови завдання виконано повністю, алгоритми складено правильно, програма містить належні коментарі, роботу програми перевірено на достатньому наборі тестових даних, автор відповідає на всі запитання щодо використаних підходів, чітко інтерпретує отримані результати, немає ознак недоброчесності;</li> <li>• 80% – наведено логічно правильну послідовність розв'язування, алгоритми складено правильно, бракує окремих коментарів чи тестів, автор не досить повно пояснює використані підходи, немає ознак недоброчесності;</li> <li>• 60% – у правильній послідовності розв'язування допущено окремі</li> </ul>		

	<p>помилки, які автор уміє виправити після зауваження викладача, бракує коментарів чи тестів, на запитання щодо використаних підходів автор відповідає з помилками, немає ознак недоброчесності;</p> <ul style="list-style-type: none"> <li>• 40% – у правильній послідовності розв’язування пропущено окремі етапи, завдання виконано частково, автор не розуміє недоліків поданої роботи, не вміє їх виправити, немає ознак недоброчесності;</li> <li>• 20% – завдання виконано частково, немає тестів, програма працює правильно для окремих наборів вхідних даних, автор не може самостійно інтерпретувати отримані результати, виправити помилки, немає ознак недоброчесності;</li> <li>• 0% – завдання не виконано, написана програма не відповідає умові, або ж виявлено ознаки недоброчесності: запозичення, фрагменти коду, дію яких автор пояснити не може, автор не володіє відповідним теоретичним матеріалом тощо;</li> </ul> <p>Запізнення зменшує максимальну оцінку за завдання: кожного наступного після терміну виконання тижня оцінка зменшується удвічі.</p> <p><b>Відвідування занять</b> є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Активність під час проведення лекцій і лабораторних заохочується балами. У будь-якому випадку студенти зобов’язані дотримуватися усіх строків визначених для виконання усіх видів письмових робіт, передбачених курсом. Виконані роботи завантажують у відповідне хмарне сховище. Альтернативою відвідування лабораторних занять в університеті може бути дистанційна онлайн робота за розкладом проведення занять. Активність на лекціях і лабораторних враховують при оцінюванні відповідного лабораторного завдання.</p> <p><b>Академічна доброчесність:</b> очікується, що роботи студентів будуть їхнім оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів, здавання чужих комп’ютерних програм як своїх становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману.</p>
<b>Опитування</b>	Анкету-оцінку з метою оцінювання якості курсу буде надано після завершення курсу.