Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Кафедра інженерії
програмного забезпечення
в енергетиці (ІПЗЕ)

ПАРАЛЕЛЬНЕ ПРОГРАМУВАННЯ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	***Перший (бакалаврський)***
Галузь знань	*12 Інформаційні технології*
Спеціальність	*121 Інженерія програмного забезпечення*
Освітня програма	*Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці*
Статус дисципліни	*Вибіркова*
Форма навчання	*очна(денна)*
Рік підготовки, семестр	*3 курс, осінній семестр*
Обсяг дисципліни	*4кред/120год (лекцій 36год., комп.прк.18 год., 66 СРС)*
Семестровий контроль/ контрольні заходи	Залік/МКР
Розклад занять	http://rozklad.kpi.ua/
Мова викладання	*Українська*
Інформація про керівника курсу / викладачів	Лектор: *к.т.н., Варава Іван Андрійович, varava_ivan@lll.kpi.ua, @Varavalvan (у робочий час)* Комп'ютерні практикуми: *к.т.н., Варава Іван Андрійович, varava_ivan@lll.kpi.ua, @Varavalvan (у робочий час)*
Розміщення курсу	Кампус

Програма навчальної дисципліни

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

Силабус навчальної дисципліни «Паралельне програмування» складено відповідно до освітньої програми підготовки бакалаврів «Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці» спеціальності 121 – Інженерія програмного забезпечення.

***Метою навчальної дисципліни** є оволодіння студентами теоретичних знань та практичних навичок створення програмного забезпечення з використанням паралельних обчислень, методів та засобів організації взаємодії потоків, засобів синхронізації потоків, формування та закріплення наступних компетентностей:*

(ЗК 2) Здатність застосовувати знання у практичних ситуаціях. (ФК 3) Здатність розробляти архітектури, модулі та компоненти програмних систем. (ФК 7) Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних. (ФК8) Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення. (ФК 14) Здатність до алгоритмічного та логічного мислення. (ФК19) Здатність застосовувати транслятори мов програмування при реалізації програмних систем.

***Предмет навчальної дисципліни** – сучасні програмні пакети, сервіси, бібліотеки для візуалізації даних.*

Програмні результати навчання, на формування та покращення яких спрямована дисципліна: *(ПРН 6) Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення; (ПРН 13) Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань. (ПРН 15) Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.*

Після засвоєння навчальної дисципліни студенти мають продемонструвати такі результати навчання:

ЗНАННЯ:

- *ключових понять і принципів організації паралельних обчислень;*
- *засобів програмування процесів за допомогою сучасних мов та бібліотек паралельного програмування;*
- *організації пам'яті та методів взаємодії процесів;*

УМІННЯ:

- *проектувати паралельні алгоритми;*
- *розраховувати характеристики ефективності паралельного алгоритму;*
- *організовувати ефективну взаємодію процесів;*
- *виконувати відлагодження паралельних програм.*

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

Пререквізити дисципліни*. Знання, отримані при вивченні дисциплін: «Основи програмування», «Алгоритми та структури даних», «Компоненти програмної інженерії. Частина 3. Архітектура програмного забезпечення».*

Постреквізити дисципліни*. Компетенції, знання та уміння, одержані в процесі вивчення кредитного модуля «Паралельне програмування» можуть бути використані при вивченні дисциплін «Розробка програмного забезпечення мобільних пристроїв», «Побудова масштабованих систем обробки даних у реальному часі», «Переддипломна практика» та «Дипломне проектування».*

3. Зміст навчальної дисципліни

Розділ 1. Основи паралельного програмування

Тема 1.1 Вступ до паралельного програмування

Тема 1.2 Класифікація паралельних комп'ютерних систем

Тема 1.3 Типи паралелізму

Тема 1.4 Характеристики ефективності паралельного алгоритму

Тема 1.5 Класичні задачі паралельного програмування

Тема 1.6 Синхронізації потоків

Розділ 2. Бібліотеки паралельних обчислень

Тема 2.1 POSIX Threads

Тема 2.2 Потоки Windows API

Тема 2.3 Розпаралелювання обчислень на OpenMP.

Тема 2.4 Огляд Intel TBB.

Розділ 3. Паралельне програмування мовою C#

Тема 3.1 Багатопоточність в C#

Тема 3.2 Бібліотека TPL

Розділ 4. Паралельне програмування на графічних процесорах

Тема 4.1 Вступ до технології CUDA

Тема 4.2 Робота з бібліотеками CUDA.

Тема 4.3 Паралельне програмування гетерогенних систем на OpenCL

4. Навчальні матеріали та ресурси

Основна література

1. *Минайленко Р.М. Паралельні та розподілені обчислення : навч. посіб. — Кропивницький: Видавець Лисенко В. Ф., 2021. – 153 с.*
2. *Малашонок Г. І., Сідько А. А. Паралельні обчислення на розподіленій пам'яті: OpenMPI, Java, Math Partner : підручник. – Київ : НаУКМА, 2020. – 266 с.*
3. *Rainer Grimm Concurrency with Modern C++. What every professional C++ programmer should know about concurrency Packt Publishing, 2019. – 727p.*
4. *Shakti Tanwar Hands-On Parallel Programming with C# and .NET Core. Packt Publishing, 2019, 346 p.*
5. *Richard Ansorge Programming in Parallel with CUDA: A Practical Guide. Cambridge University Press, 2022. – 395 p.*

Додаткова література

6. *Pas, Ruud van der Using OpenMP-The Next Step // Ruud van der Pas, Eric Stotzer, Christian Terboven / Cambridge, MA: The MIT Press, 2017. – 381p.*
7. *Michael Voss. Pro Tbb: C++ Parallel Programming with Threading Building Blocks // Michael Voss, Rafael Asenjo, James Reinders // Apress Berkeley, CA, 2019. – 754p.*
8. *NVIDIA CUDA – NVIDIA Docs [Електронний ресурс]. – Режим доступу: https://docs.nvidia.com/cuda/doc/index.html. – Назва з екрану*
9. *OpenCL Overview - The Khronos Group Inc [Електронний ресурс]. – Режим доступу: https://www.khronos.org/opencl/. – Назва з екрану.*

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

№ з/п	*Назва теми лекції та перелік основних питань (перелік дидактичних засобів, посилання на інформаційні джерела)*
1	***Тема Вступ до паралельного програмування.*** *Основні питання: Основні поняття та визначення. Огляд паралельних обчислювальних систем. Області застосування паралельних обчислень. Переваги та недоліки паралельного програмування. Представлення паралельних алгоритмів. Принципи розробки паралельних алгоритмів.*
2	***Тема Класифікація паралельних комп'ютерних систем.*** *Основні питання: Класифікація Флінна. Класифікація Шора. Класифікація Хокні. Класифікація Скіллікорна. Класифікація Дункана.*
3	***Тема Типи паралелізму.*** *Основні питання: Паралелізм на рівні бітів. паралелізм інструкцій. Паралелізм даних. Паралелізм задач. Загальна та розподілена пам'ять (SMP, MPP). Архітектура NUMA.*
4	***Тема Характеристики ефективності паралельного алгоритму*** *Основні питання: Прискорення. Продуктивність. Ефективність. Закон Амдаля. Закон Густафсона-Барсіса.*
5	***Тема Класичні задачі паралельного програмування***

	Основні питання: Задача «Виробники-споживачі». Задача «Письменники — читачі» та її варіанти. Задача «Обідаючі філософи». Задача про сплячого перукаря.
6	***Тема Засоби синхронізації потоків*** *Основні питання: Проблеми конкуренції потоків та взаємного блокування. Критичні секції. Бар'єр, Семафор. М'ютекс*
7	***Тема Засоби синхронізації потоків-2*** *Основні питання: Монітори. Реалізація моніторів. CAS-операції та атоміки.*
8	***Тема Алгоритми синхронізації*** *Основні питання: Види синхронізації: оптимістична, лінива, неблокуюча. Проблема ABA. Класи алгоритмів: lock-free, wait-free. lock-free структури даних: стек та черга*
9	***Тема POSIX Threads*** *Створення та завершення потоків. Атрибути потоків. Приєднувані та від'єднувані потоки. Модель планування потоків. Очікування потоків. Синхронізація потоків*
10	***Тема Потоки Windows API*** *Основні питання: Функції CreateThread, WaitForMultipleObjects. CloseHandle. Приклад багатопоточної програми.*
11	***Тема Розпаралелювання обчислень на OpenMP.*** *Основні питання: Модель виконання OpenMP-програми «Fork-Join». Директиви компілятора. Директиви синхронізації потоків. Директива блокування. Змінні середовища*
12	***Тема Розпаралелювання обчислень на MPI.*** *Основні питання: Інтерфейс передачі повідомлень MPI. Комунікаційна модель MPI Групи і комунікатори MPI. Виклик mpi-програми на виконання.*
13	***Тема Огляд Intel TBB.*** *Основні питання: Паралельні алгоритми. Планувальник задач. Примітиви синхронізації. Контейнери. Аллокатори.*
14	***Тема Багатопоточність в C#*** *Основні питання: Створення потоків Делегат ThreadStart. Потоки з параметрами та ParameterizedThreadStart. Синхронізація потоків. Оператор lock. Клас System.Threading.Monitor. Клас AutoResetEvent. Клас Thread.*
15	***Тема Бібліотека Task Parallel Library*** *Основні питання: Клас Task. Делегат Action. Методи очікування виконання задач. Wait. Клас Parallel. Конструкції Parallel.For та Parallel.ForEach*
16	***Тема Вступ до технології CUDA*** *Основні питання: Модель програмування CUDA, ядра, потоки, пам'ять в CUDA, Хост. Пристрій. Сітка. Блок. Нитка. Варпи. Вбудовані змінні. Атомарні операції в CUDA. Відмінність архітектури GPU від CPU. Компілятор NVCC.*
17	***Тема Робота з бібліотеками CUDA*** *Основні питанні: CUDA Toolkit. CUDA THRUST, CUDA BLAS. CUDA LAPACK. CUDA cuFFT. Приклади використання бібліотек CUDA.*
18	***Тема Паралельне програмування гетерогенних систем на OpenCL*** *Основні питання: Визначення наявності платформ та пристроїв. Контекст. Черги команд.* ***Залік*** *На заліку оголошується кінцева оцінка, яка ставиться у заліково-екзаменаційну відомість. Студенти, що не набрали 60 балів, а також, ті хто хочуть підвищити свою оцінку виконують на занятті залікову контрольну роботу. Студенти, що не допущені до заліку можуть здавати на занятті заборгованості. Якщо недопущений студент зміг протягом заняття отримати допуск та має більш*

	ніж 60 балів, він отримує залікову оцінку на цьому ж занятті. Якщо студент допустився, але 60 балів не набрав, він також має право написати залікову тестову роботу. Недопущені на занятті студенти, а також ті, хто не з'явився на залік і не мають допуску отримують у відомості «не допущений» та відправляються на додаткову сесію. Студенти, що отримали заздалегідь допуск та погоджуються зі своєю оцінкою, можуть не бути присутні на заліковому занятті.

Практичні заняття

№ з/п	*Назва теми заняття та перелік основних питань*
1	***Практичне заняття №1. Паралельний розрахунок інтеграла з використанням phthreads*** *Основні питання: Функція pthread_create(); передача аргументів потоковій функції; функція pthread_exit(); функція pthread_join(); інформація про потік.*
2	***Практичне заняття №2. Програмна реалізація класичної задачі паралельного програмування*** *Основні питання: Задача «Виробники — споживачі», задача «Письменники — читачі» та її варіанти, задача «Обідаючі філософи», задача «Сплячий перукар», механізми синхронізації:*
3	***Практичне заняття №3. Створення додатку з використанням OpenMP*** *Основні питання: Директиви*
4	***Практичне заняття №4. Створення додатку з використанням MPI*** *Основні питання: Ініціалізація підсистеми MPI; види комунікацій в MPI; блочне перемноження матриць (алгоритм Фокса).*
5	***Практичне заняття №5. Розробка багатапоточного додатка з використанням Windows API*** *Основні питання: Створення потоків CreateThread( ), ідентифікатор потоку GetCurrentThreadId(), завершення потоків ExitThread() або TerminateThread(), відновлення потоку ResumeThread(), функції очікування WaitForSingleObject() або WaitForMultipleObjects().*
6	***Практичне заняття №6. Розробка багатопоточного додатка мовою C#*** *Основні питання: пріоритет потоків; методи класу Thread: Sleep(), Join(), Interrupt(), Abort(), ResetAbort(), Suspend(), Resume();*
7	***Практичне заняття №7. Робота з паралельними колекціями в C#*** *Основні питання: клас Task, простір імен System.Collection.Concurrent, класи ConcurrentQueue<T> та ConcurrentStack<T>*
8	***Практичне заняття №8. Розв'язок задач лінійної алгебри за допомогою CUDA*** *Основні питання: Глобальна та розділювана пам'ять, специфікатори змінних в CUDA, функції копіювання даних, бібліотека CUBLAS, паралельні алгоритми лінійної алгебри.*
9	***Практичне заняття №9. Розв'язок задач математичної фізики за допомогою CUDA*** *Основні питання: Grid, Kernel, CUDA-версії математичних функцій, рівняння математичної фізики.*

6. Самостійна робота студента/аспіранта

№ з/п	Вид самостійної роботи	Кількість годин СРС
1	Підготовка до практичних занять	54
2	Підготовка до МКР	6
3	Підготовка до заліку	6

6. Контрольні роботи

Метою контрольних робіт є закріплення та перевірка теоретичних знань із кредитного модуля, набуття студентами практичних навичок самостійного вирішення задач.

Одна модульна контрольна робота (МКР) розбивається на дві контрольні роботи тривалістю в одну годину кожна. Контрольні роботи проводяться за допомогою Google Forms під час календарних контролів (атестацій).

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Система вимог, які викладач ставить перед студентом:

- *правила відвідування занять: відповідно до Наказу 1-273 від 14.09.2020 р. заборонено оцінювати присутність або відсутність здобувача на аудиторному занятті, в тому числі нараховувати заохочувальні або штрафні бали. Відповідно до РСО даної дисципліни бали нараховують за відповідні види навчальної активності на лекційних та практичних заняттях.*
- *правила поведінки на заняттях: студент має можливість отримувати бали за відповідні види навчальної активності на лекційних та практичних заняттях, передбачені РСО дисципліни. Використання засобів зв'язку для пошуку інформації на гугл-диску викладача, в інтернеті, в дистанційному курсі на платформі Сікорський здійснюється за умови вказівки викладача;*
- *політика дедлайнів та перескладань: якщо студент не проходив або не з'явиться на МКР (без поважної причини), його результат оцінюється у 0 балів. Перескладання результатів МКР не передбачено;*
- *політика щодо академічної доброчесності: Кодекс честі Національного технічного університету України «Київський політехнічний інститут» https://kpi.ua/files/honorcode.pdf встановлює загальні моральні принципи, правила етичної поведінки осіб та передбачає політику академічної доброчесності для осіб, що працюють і навчаються в університеті, якими вони мають керуватись у своїй діяльності, в тому числі при вивченні та складанні контрольних заходів з дисципліни «Інфографіка в програмному забезпеченні»;*
- *при використанні цифрових засобів зв'язку з викладачем (мобільний зв'язок, електронна пошта, переписка на форумах та у соцмережах тощо) необхідно дотримуватись загальноприйнятих етичних норм, зокрема бути ввічливим та обмежувати спілкування робочим часом викладача.*

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

***Поточний контроль**: тестування, МКР, виконання завдань до практичних занять.*

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

Семестровий контроль: *залік.*

***Умови допуску до семестрового контролю**: семестровий рейтинг більше 40 балів.*

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
95-100	Відмінно
85-94	Дуже добре
75-84	Добре
65-74	Задовільно

60-64	*Достатньо*
Менше 60	*Незадовільно*
Менше 30	*Не допущено*

Загальна рейтингова оцінка студента після завершення семестру складається з балів, отриманих за:

- *тестування по кожному лекційному заняттю;*
- *виконання завдань до практичних занять;*
- *виконання модульних контрольних робіт (МКР);*
- *виконання додатковий завдань.*

Тестування по лекціям	*Практичні заняття*	*МКР 1*	*МКР 2*	*Додаткові бали*
18	*54*	*10*	*10*	*8*

Тестування по матеріалам лекційних занять

Ваговий бал 1. *Максимальна кількість балів за тестування – 1 бал * 18 лекцій = 18 балів.*

Тестування проводиться за допомогою Google Forms наприкінці поточної лекції. У деяких випадках термін проходження тестування може бути продовжений лектором. Тривалість проходження одного тестування – 5 хвилин. Кількість спроб – одна. У деяких випадках, що пов'язані з технічними проблемами студентів, може надатися повторна спроба на окремі тестування.

Кожне тестування містить 5 запитань різного формату (вибір правильного варіанту з переліку; вибір правильних варіантів з переліку, вірно/невірно; визначити відповідність; чисельна відповідь).

Критерії оцінювання

- *запитання типу «вибір правильного варіанту з переліку», «вірно/невірно», «чисельна відповідь» оцінюються однозначно: вірна відповідь – 0,2 бал, невірна відповідь – 0 балів;*
- *запитання, на які немає однієї конкретної відповіді, типу «вибір правильних варіантів з переліку» або «визначити відповідність» оцінюються у відповідності до кількості елементів у тесті (наприклад, якщо потрібно обрати 4 вірні варіанти відповіді із 7, то студент отримає по 0,05 балів за один правильний варіант відповіді, а за всі 4 правильні відповіді отримає відповідно 0,2 бал) – невірна відповідь – 0 балів, частково вірна відповідь – 0,05-0,15 балів, вірна відповідь 0,2 бал.*

Практичні заняття

Ваговий бал – 6*. Максимальна кількість балів за всі практичні заняття – 6 бали * 9 занять= 54 балів.*

На практичних занять студенти розробляють програмний код відповідності до тематики практичного заняття. Після кожного практичного заняття студенти отримують домашнє завдання, яке необхідно вирішити та надати на перевірку викладачу до початку наступного заняття (зазвичай це 2 тижні, однак іноді цей час може бути змінений викладачем у деяких конкретних випадках).

Критерії оцінювання

- *домашнє завдання вирішено вірно та здано протягом 2-х тижнів після практичного заняття – 6 балів;*
- *домашнє завдання вирішено вірно, але здано протягом більш ніж 2-х тижнів після практичного заняття – 5 балів;*
- *домашнє завдання вирішено із незначними помилками та здано протягом 2-х тижнів після практичного заняття – 4 бали;*
- *домашнє завдання вирішено із незначними помилками та здано протягом більш ніж 2-х тижнів після практичного заняття – 3 бали;*

- домашнє завдання вирішено із значними помилками – повертається на доопрацювання.

Модульна контрольна робота

Ваговий бал за одну МКР – 10. Максимальний бал за 2 МКР складає 20 балів.

Критерії оцінювання

Кожна частина модульної контрольної роботи складається із 20 завдань. Окреме завдання оцінюється в 0.5 бали.

Календарний контроль базується на поточній рейтинговій оцінці. Умовою позитивної атестації є значення поточного рейтингу студента не менше 50% від максимально можливого на час атестації.

Додаткові (бонусні) бали

Рейтинговою системою оцінювання передбачені додаткові бали за виконання додаткових завдань. Один студент не може отримати більше ніж 8 бонусних балів у семестрі. При отриманні більш ніж 8 балів, вони обмежуються на рівні 8. Бонусні бали можуть бути отримані за виконання «Завдання до лекцій».

Завдання до лекцій. *Студенти, за бажанням, можуть виконувати додаткові завдання за матеріалами лекцій (розробити макрос чи користувацьку функцію). За одне додаткове завдання нараховується 0,5 бали. Максимальна кількість балів, що можна отримати за завдання до лекцій складає 8 балів.*

Форма семестрового контролю – залік

Максимальна сума балів складає 100. Необхідною умовою допуску до заліку є зараховані обидві частини МКР та здані усі завдання до практичних занять. Для отримання заліку з кредитного модулю «автоматом» потрібно мати рейтинг не менше 60 балів, а також виконані умови допуску до заліку.

Студенти, які наприкінці семестру мають рейтинг менше 60 балів, а також ті, хто хоче підвищити свою оцінку в системі ECTS, виконують залікову контрольну роботу. При цьому набрані бали студентом анулюються, а оцінка за залікову контрольну роботу є остаточною.

Залікова робота. *Залікова робота проводиться на останньому лекційному занятті. Студент проходить тестування за допомогою Google Forms. На тестування пропонується 50 тестових завдань, кожне з яких оцінюється в 2 бали. Для отримання позитивної оцінки необхідно набрати 60 балів і вище. Час тестування зазвичай складає 90 хвилин, але може бути скоригований лектором та (або) викладачам, що приймає залік.*

Робочу програму навчальної дисципліни (силабус) «Паралельне програмування»:

Складено доцентом кафедри інженерії програмного забезпечення в енергетиці, к.т.н. Варавою І.А.

Ухвалено кафедрою інженерії програмного забезпечення в енергетиці (протокол № 34 від 10.05.2024 р.)

Погоджено Методичною комісією ННІАТЕ КПІ ім. Ігоря Сікорського (протокол № 9 від 31.05.2024 р.)