

**ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І.І. МЕЧНИКОВА
ФАКУЛЬТЕТ МАТЕМАТИКИ, ФІЗИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

КАФЕДРА ОПТИМАЛЬНОГО КЕРУВАННЯ ТА ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ

СИЛАБУС КУРСУ «МОДЕЛІ ТА ЗАСОБИ БАГАТОПОТОЧНОГО ПРОГРАМУВАННЯ»

Обсяг	загальна кількість: кредитів –5; годин – 150; змістовних модулів – 2
Семестр	осінній
Дні, Час, Місце	за розкладом занять
Викладач(і)	Вербіцький Віктор Васильович, к. фіз.-мат. наук, доцент, доцент кафедри оптимального керування та економічної кібернетики
Контактний телефон	(048)7340723
E-mail	v.verbitskyi@onu.edu.ua
Робоче місце	кафедра оптимального керування та економічної кібернетики
Консультації	очні консультації: за розкладом on-line консультації: ZOOM за розкладом (посилання генерується на початку занять)

КОМУНІКАЦІЯ

Комунікація зі студентами буде здійснюватися в аудиторії, та також з використанням додатків Google Клас, Google Meet, ZOOM.

АНОТАЦІЯ КУРСУ

Предметом вивчення курсу є основні парадигми паралельного багатопоточного програмування та технології створення паралельних додатків для сучасних комп'ютерних архітектур з пам'яттю що поділяється.

Пререквізити курсу

Матеріал курсу ґрунтується на раніше отриманих студентами знаннях, практичних вміннях та навичках з тем та напрямів щодо програмування, алгоритмів, структур даних, операційних систем та системного програмування. Відповідні курси викладаються у межах освітньої програми першого (бакалаврського) рівня вищої освіти за спеціальністю 123 «Комп'ютерна інженерія».

Постреквізити курсу

Цей курс доповнює дисципліни «Програмування», «Архітектура комп'ютерів та низькорівневе програмування», «Введення в сучасні операційні системи і середовища», «Структури даних та алгоритми» в сфері розробки паралельних та розподілених програм на основі паралельних алгоритмів для комп'ютерів з сучасною архітектурою і є базою для засвоєння наступних дисциплін освітньо-професійної програми підготовки бакалаврів за спеціальністю 123 «Комп'ютерна інженерія»: «Проектно-технологічна практика», «Преддипломна практика», «Дипломне проектування».

Метою курсу є вивчення основних принципів побудови паралельних багатопоточних програмних додатків для комп'ютерних систем з пам'яттю що поділяється, а також придбання практичних навичок щодо створення, тестування та експлуатації паралельного програмного продукту з використанням сучасних пакетів та стандартів паралельного програмування за допомогою потоків.

Зміст курсу

Розглядаються:

- Процес проектування паралельних багатопотокових програм: декомпозиція, зв'язок, синхронізація. Середовища для паралельного багатопотокового програмування.
- Класифікація паралельних обчислювальних систем: SMP-системи, кластери, MPP-системи. Системи із загальною й розподіленою пам'яттю. Архітектура NUMA та ccNUMA. Організація когерентності багаторівневої ієрархічної пам'яті в SMP-системах.
- Головні парадигми паралельного програмування з використанням потоків. Ітеративний паралелізм (множення матриць). Рекурсивний паралелізм (адаптивна квадратура). «Виробники та споживачі» (конвеєри). Проблеми паралельного і розподіленого програмування: «гонка» даних, нескінченна відстрочка, взаємоблокування, труднощі організації зв'язку.
- Паралельне програмування з використанням потоків стандарту POSIX. Створення потоку. Атрибути потоку. Очікувані й від'єднанні потоки. Скасування потоку: асинхронне скасування, синхронне скасування, потоки, які не можна скасувати. Потоківі дані. Оброблювач очищення. Очищення поточкових даних у C++. Поточкові семафори. Мютекси. Умовні змінні POSIX.
- Технологія паралельного програмування OpenMP. Модель OpenMP-додатка. Директива паралельної обробки `parallel`. Директива розподілення роботи `for`. Директиви розподілення роботи `sections` та `section`. Директиви `single` та `master`. Директиви `tasks` та `taskwait`. Директиви синхронізації `barrier`, `ordered`, `critical`, `atomic`. Спільні та приватні змінні. Функції середовища виконання. Функції блокування та синхронізації. Змінні оточення. Алгоритми планування паралельного виконання циклів (`static`, `dynamic`, `guided`, `runtime scheduling`).
- Паралелізм на основі потоків за допомогою Python. Модуль потоків Python. Визначення поточкового потоку. Визначення підкласу потоку. Синхронізація потоку з блокуванням. Синхронізація потоків з `RLock`. Синхронізація потоків із семафорами. Синхронізація потоку з умовою. Синхронізація потоку з подією. Синхронізація потоку з бар'єром. Поточкове спілкування за допомогою черги.

ОЧІКУВАНІ РЕЗУЛЬТАТИ

У результаті вивчення курсу студент повинен

знати: основні парадигми паралельного програмування з використанням потоків; основні прийоми написання паралельних програм з використанням потоків стандарту POSIX; технологію OpenMP написання паралельних програм з використанням потоків.

вміти: створювати паралельні алгоритми для розв'язування різноманітних задач з використанням потоків; аналізувати ефективність паралельного алгоритму; реалізувати паралельний алгоритм для паралельних комп'ютерних систем з пам'яттю що поділяється;

Компетентності, які отримує студент у результаті вивчення курсу:

- Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності.

Результати навчання: по завершенню курсу студент матиме навички

- здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії.
- виконувати експериментальні дослідження за професійною тематикою.
- оцінювати отримані результати та аргументовано захищати прийняті рішення.

ФОРМИ І МЕТОДИ НАВЧАННЯ

Курс буде викладений у формі лекцій (36 год.) та лабораторних занять (18 год.), організації самостійної роботи студентів (136 год.).

Основна підготовка студентів здійснюється на лекційних та лабораторних заняттях.

Під час викладання курсу використовуються такі **методи навчання:** словесні (лекція, пояснення); наочні (презентація); практичні (лабораторні роботи); робота з літературними джерелами (самостійна робота студентів).