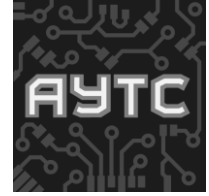




Силабус освітнього компонента Програма навчальної дисципліни



Програмування спеціалізованих обчислювальних систем

Шифр та назва спеціальності

174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка

Інститут

ННІ Комп'ютерного моделювання, прикладної фізики та математики

Освітня програма

Комп'ютерні технології та програмування в автоматизованих системах керування

Кафедра

Автоматика та управління в технічних системах (172)

Рівень освіти

Магістр

Тип дисципліни

Вибіркова, профільна підготовка

Семестр

2

Мова викладання

Українська

Викладачі, розробники



Сальніков Дмитро Валентинович

dmytro.salnikov@khp.edu.ua

к. т. н., старший викладач кафедри "Автоматика та управління в технічних системах", НТУ "ХПІ"

Досвід роботи – 9 років. Автор та співавтор понад 20 наукових та навчально-методичних праць. Основні професійні та наукові інтереси: дослідження ефективності та вдосконалення методів цифрової обробки сигналів, цифрова фільтрація та відокремлення шумів, використання нейронних мереж для цифрової фільтрації.

Провідний лектор з ряду фахових дисциплін.

[Детальніше про викладача на сайті кафедри](#)

Загальна інформація

Анотація

Курс призначений для студентів, які бажають здобути глибокі знання в області програмування, зокрема у створенні спеціалізованих програм, що вимагають високої паралельності обчислень, та оптимізації програм за швидкодією. Він охоплює як теоретичні, так і практичні елементи розробки програмного забезпечення, що працює на сучасних процесорах і графічних акселераторах компанії NVIDIA. Студенти дізнаються про методи профілювання програм, аналізу теоретичної продуктивності алгоритмів та шляхи досягнення максимальної продуктивності.

Мета та цілі дисципліни

Мета курсу — надати студентам необхідні теоретичні знання та практичні навички для створення ефективного програмного забезпечення, призначеного для графічних прискорювачів NVIDIA. Студенти вивчають, як розробляти програми, що використовують паралельні обчислення для підвищення продуктивності, а також методи оцінки та оптимізації швидкості виконання програмного коду.

Формат занять

Лекції та лабораторні роботи, консультації. Передбачається самостійне опрацювання лекційного матеріалу, підготовка та виконання лабораторних робіт. Проводиться проміжний модульний контроль у вигляді контрольних робіт або електронного тестування. Підсумковий контроль — іспит.

Компетентності

Вивчення даної дисципліни забезпечує формування у студентів програмних компетентностей ЗК3, СК2, та СК7 згідно освітньої програми, а саме: здатність до абстрактного мислення, аналізу та синтезу, здатність проектувати та впроваджувати високонадійні системи автоматизації та їх прикладне програмне забезпечення, для реалізації функцій управління та опрацювання інформації, здійснювати захист прав інтелектуальної власності на нові проектні та інженерні рішення, здатність застосовувати спеціалізоване програмне забезпечення та цифрові технології для розв'язання складних задач і проблем автоматизації та комп'ютерно-інтегрованих технологій.

Результати навчання

Вивчення даної дисципліни передбачає досягнення програмних результатів навчання РН09 та РН10, згідно освітньої програми, а саме: розробляти функціональну, організаційну, технічну та інформаційну структури систем автоматизації складними технологічними та організаційно-технічними об'єктами, розробляти програмно-технічні керуючі комплекси із застосуванням мережеских та інформаційних технологій, промислових контролерів, мехатронних компонентів, робототехнічних пристроїв, засобів людино-машинного інтерфейсу та з урахуванням технологічних умов та вимог до управління виробництвом, розробляти і використовувати спеціалізоване програмне забезпечення та цифрові технології для створення систем автоматизації складними організаційно-технічними об'єктами, професійно володіти спеціальними програмними засобами.

Обсяг дисципліни

Загальний обсяг дисципліни 150 год. (5 кредитів ECTS): лекції – 32 год., лабораторні роботи – 32 год., самостійна робота – 86 год.

Передумови вивчення дисципліни (пререквізити)

Студент повинен мати базові навички та знання впевненого користувача ПК, вміти здійснювати інформаційний пошук матеріалів за заданою темою, мати навички програмування та принципів роботи процесорів, бажано володіння англійською мовою. Перед початком вивчення даної дисципліни студент повинен мати навички використання будь-якої мови програмування (бажано – мови C++), знання про алгоритми та структури даних, принципи організації обчислювальних систем.

Особливості дисципліни, методи та технології навчання

Усі навчальні матеріали надаються в електронному вигляді, що дозволяє проведення занять як у аудиторному, так і у дистанційному форматі. Навчальні матеріали адаптовано до проведення занять з використанням Microsoft Teams. Для виконання лабораторних робіт необхідно використовувати персональний комп'ютер з ОС Windows/Linux/macOS (від 4 ГБ ОЗП, 2 ГБ дискового простору, доступ в Інтернет), лабораторні роботи виконуються в онлайн середовищі Google Colab або на персональному комп'ютері студента за наявності графічного прискорювача NVIDIA.

Програма навчальної дисципліни

Теми лекційних занять

Тема 1. Вступ до паралельного програмування

Вступ до паралельного програмування. Переваги та недоліки паралельних обчислювальних засобів. Мови програмування що використовують для програмування паралельних обчислювачів.

Тема 2. Апаратна структура сучасних обчислювальних засобів

Мікропроцесорні засоби паралельних обчислень (SIMD multicore CPUs). Конвеєр мікропроцесору. Графічні прискорювачі (GPUs).

Тема 3. Основи програмування CUDA

Модель програмування CUDA: потоки, блоки та сітки. Функції мови CUDA C. Написання першої програми CUDA. Запуск ядер.

Тема 4. Керування пам'яттю в програмах CUDA

Розуміння ієрархії пам'яті в архітектурі GPU. Глобальна, постійна, спільна та текстурована пам'ять. Стратегії керування пам'яттю: об'єднаний і необ'єднаний доступ, доповнення та вирівнювання.

Тема 5. Передача даних між процесором і графічним процесором

Розподіл пам'яті на хості та пристрої. Передача даних за допомогою `cudaMemcpy()` та його варіацій. Асинхронна передача даних

Тема 6. Векторні та матричні операції

Множення матриць. Пошук та сортування.

Тема 7. Використання бібліотек Nvidia Cuda

Огляд Pytorch/Tensorflow/cuDNN/JAX

Теми практичних занять

Практичні заняття не передбачені навчальним планом ОП.

Теми лабораторних робіт

Лабораторна робота 1. Методи оптимізації ПЗ

Оптимізація алгоритмів за допомогою інтринсик функцій. Спеціалізовані інструкції процесорів.

Лабораторна робота 2. Введення в CUDA

Розподіл та види пам'яті. Ядра CUDA. Компіляція та запуск програм CUDA.

Лабораторна робота 3. Вимірювання швидкості програмних засобів

Профілювання за допомогою NVIDIA Nsight Systems

Лабораторна робота 4. Синхронізація в CUDA.

Дослідження алгоритмів сортування.

Лабораторна робота 5. Техніки оптимізації виконання програм (Частина 1)

Дослідження алгоритму множення матриць. Патерни використання пам'яті.

Лабораторна робота 6. Техніки оптимізації виконання програм (Частина 2)

Дослідження алгоритму множення матриць. Спільна та кеш пам'ять.

Лабораторна робота 7. Техніки оптимізації виконання програм (Частина 3)

Дослідження алгоритму множення матриць. Векторизація.

Лабораторна робота 8. Використання бібліотек CUDA

Створення програмних засобів з використанням бібліотек CuDNN, CuBLAS.

Самостійна робота

Передбачається виділення часу на самостійне опрацювання лекційного матеріалу (32 год.) та підготовка та виконання лабораторних робіт (32 год.). Більшість лабораторних робіт потребує виділення додаткового самостійного часу на доопрацювання та оформлення звіту. Навчальним планом передбачено виконання курсової роботи (20 год.), перелік тем визначається викладачем та оприлюднюється на початку навчального семестру. Строк здавання готових робіт – за три тижні до кінця навчального семестру. Приклади тем: виконання одночасного розрахунку трансформації графічних об'єктів (отримання координат фігур внаслідок обертання та переміщення), виконання одночасного розрахунку накопичених статистичних даних за різні періоди часу, виконання багатопоточного шифрування файлів даних.

Література та навчальні матеріали

Основна література:

1. Зуєв А.О., Гапон Д.А., Денисенко М.А. Методичні вказівки до виконання самостійних робіт «Основи програмування на мові С++» Х.: 2022, 45с. наказ НТУ «ХПІ» №3 26.10.2022р. поз. 342.
<http://repository.kpi.kharkov.ua/handle/KhPI-Press/60574>
2. Stroustrup, Bjarne. The C++ programming language / Bjarne Stroustrup.— Fourth edition. Published by Pearson Education, Inc. 2013. 1360 p.
3. Грицюк Ю., Рак Т. Програмування мовою С++. Навчальний посібник. / Юрій Грицюк, Тарас Рак. Львів. Вид-во ЛДУ БЖД 2011. 290 с.
4. Jaegeun Han. Learn CUDA Programming / Jaegeun Han, Bharatkumar Sharma. - Published by Packt Publishing Ltd. 2019. 502 p. ISBN 978-1-78899-624-2
5. Bhaumik Vaidya. Hands-On GPU-Accelerated Computer Vision with OpenCV and CUDA - Published by Packt Publishing Ltd. 2018. 380 p. ISBN: 978-1-78934-829-3

Додаткова література:

1. CUDA C++ Programming Guide <https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html>
2. Jason Sanders. CUDA by Example: An Introduction to General-Purpose GPU Programming / Jason Sanders, Edward Kandrot. Published by Addison-Wesley Professional; 1st edition, 2010, 320p. ISBN: 978-0131387683

Система оцінювання

Критерії оцінювання успішності студента та розподіл балів

Підсумкова оцінка формується шляхом підрахунку середнього зваженого балу на базі оцінок за виконання лабораторних робіт та проміжного (модульного) контролю. Якщо студент не погоджується з рейтинговою оцінкою, або було виконано недостатньо завдань для формування підсумкової оцінки (менш ніж 70%), студент має отримати оцінку шляхом складання іспиту. На іспиті студент має дати відповідь на декілька (2-3) теоретичних питань за різними темами курсу та виконати практичне завдання з програмування. Оцінки виставляються за наведеною шкалою та у відповідності до критеріїв, які доступні за [посиланням](#).

Шкала оцінювання

Сума балів	Національна оцінка	ECTS
90–100	Відмінно	A
82–89	Добре	B
75–81	Добре	C
64–74	Задовільно	D
60–63	Задовільно	E
35–59	Незадовільно (потрібне додаткове вивчення)	FX
1–34	Незадовільно (потрібне повторне вивчення)	F

Норми академічної етики і політика курсу

Студент повинен дотримуватися «Кодексу етики академічних взаємовідносин та доброчесності НТУ «ХПІ»: виявляти дисциплінованість, вихованість, доброзичливість, чесність, відповідальність. Конфліктні ситуації повинні відкрито обговорюватися в навчальних групах з викладачем, а при неможливості вирішення конфлікту – доводитися до відома співробітників дирекції інституту. Нормативно-правове забезпечення впровадження принципів академічної доброчесності НТУ «ХПІ» розміщено на сайті: <http://blogs.kpi.kharkov.ua/v2/nv/akademichna-dobrochesnist/>

Погодження

Силабус погоджено

04.04.2024

Завідувач кафедри АУТС
Андрій ЗУЄВ

Гарант ОП
Ігор КРАСНІКОВ