

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ЗАТВЕРДЖУЮ

Декан ФІТ

Говорущенко Т.О.

5 вересня 2024р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Теорія і технології проектування спеціалізованих операційних систем

Назва

Галузь знань 12 – Інформаційні технології

Спеціальність 123 – Комп'ютерна інженерія очна денна форма здобуття освіти

Освітня програма Комп'ютерна інженерія та програмування

Статус дисципліни: обов'язкова, цикл науково-дослідна підготовка

Факультет – Інформаційних технологій

Кафедра – Комп'ютерної інженерії та інформаційних систем

| Форма здобуття освіти | Курс | Семестр | Загальне навантаження | | Кількість годин | | | | | | Форма семестрового контролю | | | |
|-----------------------|------|---------|-----------------------|------------|-------------------|-----------|--------------------|-------------------|-------------------------------|-------------------------------|-----------------------------|----------------|----------|-------|
| | | | Кредити ЕКТС | Години | Аудиторні заняття | | | | Індивідуальна робота студента | Самостійна робота, в т.ч. ІРС | Курсовий проект | Курсова робота | Залік | Іспит |
| | | | | | Разом | Лекції | Лабораторні роботи | Практичні заняття | | | | | | |
| ОД | 1 | 2 | 5,0 | 150 | 54 | 18 | 36 | 0 | | 96 | | | + | |
| Разом | | | 5,0 | 150 | 54 | 18 | 36 | 0 | | 96 | | | 1 | |

Програма складена


Підпис

Денисюком Д.О.

Ініціали, прізвище викладача(ів)

Схвалена на засіданні кафедри комп'ютерної інженерії та інформаційних систем

Протокол 2 від 30.08.2024 р.

Зав. кафедри комп'ютерної інженерії та інформаційних систем

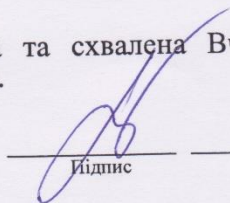
Підпис

Засорнова І.О.

Ініціали, прізвище

Робоча програма розглянута та схвалена Вченою радою факультету інформаційних технологій
Протокол №1 від 05.09.2024 р.

Голова Вченої ради


Підпис

Говорущенко Т.О.

Ініціали, прізвище

ВСТУП

Дисципліна "Теорія і технології проектування спеціалізованих операційних систем" відноситься до обов'язкових дисциплін циклу науково-дослідної підготовки магістрів зі спеціальності «Комп'ютерна інженерія» і вивчає принципи побудови сучасних операційних систем та основи драйверної концепції.

Дисципліна розглядає архітектурні особливості сучасних спеціалізованих обчислювальних систем, зокрема реального часу, Інтернету речей, кіберфізичних, мобільних, робототехнічних, розподілених, а також теорію проектування спеціалізованих ОС для цих систем, приклади та шляхи розв'язку наукових задач з теорії проектування спеціалізованих операційних систем; засади драйверної концепції як основи взаємодії операційної системи (користувача) з пристроями (системними/периферійними, реальними/віртуальними тощо), принципи роботи драйверної концепції та теорію і технології проектування драйверів з використанням цієї концепції; проміжне програмне забезпечення для розподілених систем та його місце в загальній системі системного програмного забезпечення; концепція блокчейн технології; проблемні наукові задачі в системі реалізації паралельних обчислень та шляхи їх розвитку; балансування навантаження та забезпечення відмовостійкості в розподілених мережах.

Метою дисципліни є: 1) ознайомити студентів з поняттям про архітектурні особливості сучасних спеціалізованих обчислювальних систем, зокрема реального часу, Інтернету речей, кіберфізичних, мобільних, робототехнічних, розподілених; 2) сформулювати вміння використання спеціалізованих операційних систем в сучасній промисловості при практичному вирішенні задач; 3) надати глибокі знання принципів функціонування спеціалізованих операційних систем і використання теорії проектування спеціалізованих операційних систем для сучасних спеціалізованих обчислювальних систем, зокрема реального часу, Інтернету речей, кіберфізичних, мобільних, робототехнічних, розподілених, а також шляхів вирішення проблемних наукових задач в галузі проектування спеціалізованих операційних систем.

Предмет дисципліни. Особливості архітектур сучасних спеціалізованих операційних систем, зокрема реального часу, Інтернету речей, кіберфізичних, мобільних, робототехнічних та розподілених; суть драйверної концепції як невід'ємної частини сучасних операційних систем та особливості технології написання драйверів; проміжне програмне забезпечення для розподілених та паралельних обчислювальних систем; блокчейн технології.

Завдання курсу:

- розглянути архітектурні особливості сучасних спеціалізованих обчислювальних систем, зокрема реального часу, Інтернету речей, кіберфізичних, мобільних, робототехнічних, розподілених;
- навчити застосовувати знання основ функціонування сучасних спеціалізованих операційних систем та побудови драйверів при проектуванні сучасних

спеціалізованих операційних систем, зокрема реального часу, Інтернету речей, кіберфізичних, мобільних, робототехнічних, розподілених;

- навчити застосовувати знання технічних характеристик та конструктивних особливостей спеціалізованих обчислювальних систем, зокрема реального часу, Інтернету речей, кіберфізичних, мобільних, робототехнічних, розподілених, при розробленні спеціалізованих операційних систем;
- навчити проектувати проміжне програмне забезпечення для розподілених систем;
- навчити проектувати та створювати спеціалізовані операційні системи, зокрема реального часу, Інтернету речей, кіберфізичних, мобільних, робототехнічних, розподілених систем з урахуванням усіх аспектів їх життєвого циклу.

В ході вивчення дисципліни здійснюються такі види контролю: поточний – захист лабораторних робіт, підсумковий контроль – іспит.

В результаті вивчення курсу «Теорія і технології проектування спеціалізованих операційних систем» студент повинен

знати:

- архітектурні особливості сучасних спеціалізованих обчислювальних систем, зокрема реального часу, Інтернету речей, кіберфізичних, мобільних, робототехнічних, розподілених;
- теорію проектування спеціалізованих ОС для спеціалізованих обчислювальних систем, зокрема реального часу, Інтернету речей, кіберфізичних, мобільних, робототехнічних, розподілених та шляхи розв'язку наукових задач з теорії проектування спеціалізованих операційних систем;
- засади драйверної концепції та теорію і технології проектування драйверів з використанням цієї концепції;
- місце проміжного програмного забезпечення для розподілених систем в загальній системі системного програмного забезпечення;
- концепція блокчейн технології;
- шляхи розв'язку проблемних наукових задач в системі реалізації паралельних обчислень та блокчейн технології;

уміти:

- застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів спеціалізованих обчислювальних систем для вирішення задач проектування спеціалізованих операційних систем, зокрема систем реального часу, Інтернету речей, кіберфізичних, мобільних, робототехнічних та розподілених;
- застосовувати загальні підходи пізнання, методи математики, природничих та інженерних наук до розв'язання складних задач з теорії проектування спеціалізованих операційних систем;
- вирішувати задачі аналізу та синтезу спеціалізованих ОС для спеціалізованих обчислювальних систем;

- розробляти спеціалізовані операційні системи для спеціалізованих обчислювальних систем, зокрема реального часу, Інтернету речей, кіберфізичних, мобільних, робототехнічних, розподілених, а також розробляти драйвери пристроїв;
- аналізувати проблематику, ідентифікувати та формулювати проблеми в галузі наукових задач з теорії проєктування спеціалізованих операційних систем та обирати ефективні методи їх вирішення;
- виконувати наукові дослідження в галузі теорії проєктування спеціалізованих ОС для спеціалізованих обчислювальних систем та блокчейн технології, формулювати і перевіряти гіпотези, обирати методики та інструменти розв'язку наукових задач, аналізувати результати та обґрунтовувати одержані висновки;
- виконувати експериментальні дослідження в галузі теорії проєктування спеціалізованих операційних систем та блокчейн технології;
- проєктувати та розробляти проміжне програмне забезпечення та спеціалізовані операційні системи, включаючи інтерфейси спеціалізованих ОС, враховуючи їх призначення, необхідний рівень якості та зручність використання (usability);

бути здатним:

- класифікувати та описувати роботу спеціалізованих операційних систем та драйверів;
- визначати технічні характеристики, конструктивні особливості, застосування і експлуатацію спеціалізованих обчислювальних систем, спеціалізованих операційних систем та проміжного програмного забезпечення різного призначення;
- розробляти спеціалізовані операційні системи для спеціалізованих обчислювальних систем, зокрема реального часу, Інтернету речей, кіберфізичних, мобільних, робототехнічних, розподілених систем, а також драйвери пристроїв з використанням сучасних методів і мов програмування, а також програмних інтегрованих середовищ розробки;
- обирати ефективні методи розв'язування складних задач з проєктування спеціалізованих операційних систем, реалізації блокчейн технологій, критично оцінювати отримані результати та аргументувати прийняті рішення;
- використовувати методи аналізу, ідентифікації й синтезу спеціалізованих операційних систем для спеціалізованих обчислювальних систем, зокрема реального часу, Інтернету речей, кіберфізичних, мобільних, робототехнічних, розподілених;
- проєктувати спеціалізовані операційні системи для спеціалізованих обчислювальних систем, зокрема реального часу, Інтернету речей, кіберфізичних, мобільних, робототехнічних, розподілених з урахуванням усіх аспектів їх життєвого циклу та поставленої задачі, включаючи створення, налаштування, експлуатацію та технічне обслуговування;
- проєктувати та розробляти інтерфейси спеціалізованих операційних систем, враховуючи їх призначення та зручність використання;
- використовувати методи проєктування спеціалізованих операційних систем при проєктуванні та розробленні спеціалізованих операційних систем для

спеціалізованих обчислювальних систем, зокрема реального часу, Інтернету речей, кіберфізичних, мобільних, робототехнічних, розподілених систем;

- досліджувати проблеми в галузі проектування спеціалізованих операційних систем, розподілених обчислень та блокчейн технологій та визначати їх обмеження;
- будувати архітектуру та створювати проміжне програмне забезпечення для розподілених обчислювальних систем;
- досліджувати, розробляти та обирати технології створення спеціалізованих операційних систем для спеціалізованих обчислювальних систем.

Компетентності, на формування яких спрямовано ОК:

Інтегральна – Здатність розв'язувати складні задачі і проблеми в галузі комп'ютерної інженерії або у процесі навчання, що передбачає проведення досліджень та/або здійснення інновацій та характеризується невизначеністю умов і вимог.

ЗК3. Здатність проводити дослідження на відповідному рівні.

ЗК6. Здатність виявляти, ставити та вирішувати проблеми.

ЗК10. Базові дослідницькі навички і вміння.

СК1. Здатність до визначення технічних характеристик, конструктивних особливостей, застосування і експлуатації програмних, програмно-технічних засобів, комп'ютерних систем та мереж різного призначення.

СК2. Здатність розробляти алгоритмічне та програмне забезпечення, компоненти комп'ютерних систем та мереж, Інтернет додатків, кіберфізичних систем з використанням сучасних методів і мов програмування, а також засобів і систем автоматизації проектування.

СК5. Здатність будувати архітектуру та створювати системне і прикладне програмне забезпечення комп'ютерних систем та мереж.

СК7. Здатність досліджувати, розробляти та обирати технології створення великих і надвеликих систем.

СК10. Здатність ідентифікувати, класифікувати та описувати роботу програмно-технічних засобів, комп'ютерних систем, мереж та їхніх компонентів.

СК11. Здатність обирати ефективні методи розв'язування складних задач комп'ютерної інженерії, критично оцінювати отримані результати та аргументувати прийняті рішення.

СК12. Здатність використовувати методи аналізу, ідентифікації й синтезу комп'ютерних систем та мереж, кіберфізичних систем, засобів Інтернету речей та IT-інфраструктур.

Спеціальні компетентності, визначені за освітньою програмою:

СК13. Здатність проектувати системи та їхні компоненти з урахуванням усіх аспектів їх життєвого циклу та поставленої задачі, включаючи створення, налаштування, експлуатацію, технічне обслуговування та утилізацію.

СК14. Здатність проектувати та розробляти інтерфейси користувача програмних систем, враховуючи їх призначення та зручність використання.

СК15. Здатність використовувати методи фундаментальних і прикладних дисциплін комп'ютерної інженерії при проектуванні та розробленні телекомунікаційних систем.

СК17. Здатність досліджувати проблему в галузі комп'ютерних та інформаційних технологій, визначати їх обмеження.

Програмні результати навчання, на забезпечення яких спрямовано ОК:

ПРН1. Застосовувати загальні підходи пізнання, методи математики, природничих та інженерних наук до розв'язання складних задач комп'ютерної інженерії.

ПРН4. Застосовувати спеціалізовані концептуальні знання, що включають сучасні наукові здобутки у сфері комп'ютерної інженерії, необхідні для професійної діяльності, оригінального мислення та проведення досліджень, критичного осмислення проблем інформаційних технологій та на межі галузей знань.

ПРН6. Аналізувати проблематику, ідентифікувати та формулювати конкретні проблеми, що потребують вирішення, обирати ефективні методи їх вирішення.

ПРН7. Вирішувати задачі аналізу та синтезу комп'ютерних систем та мереж.

ПРН8. Застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів комп'ютерних систем та мереж для вирішення складних задач комп'ютерної інженерії та дотичних проблем.

ПРН9. Розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем.

ПРН14. Планувати і виконувати наукові дослідження в сфері комп'ютерної інженерії, формулювати і перевіряти гіпотези, обирати методики та інструменти, аналізувати результати, обґрунтовувати висновки.

ПРН15. Вміти виконувати експериментальні дослідження за професійною тематикою.

ПРН17. Вміти проектувати та розробляти системне та прикладне програмне забезпечення, включаючи інтерфейси користувача програмних систем, враховуючи його призначення, необхідний рівень якості та зручність використання (usability).

ТЕОРІЯ І ТЕХНОЛОГІЇ ПРОЄКТУВАННЯ СПЕЦІАЛІЗОВАНИХ ОПЕРАЦІЙНИХ СИСТЕМ

| | |
|--|------------------------|
| Тип дисципліни | Обов'язкова |
| Рівень вищої освіти | Другий (магістерський) |
| Мова викладання | Українська |
| Семестр | 2 |
| Кількість встановлених кредитів ЄКТС | 5 |
| Форми навчання, для яких викладається дисципліна | Денна |

Результати навчання. Студент, який успішно завершив вивчення дисципліни, повинен: застосовувати загальні підходи пізнання, методи математики, природничих та інженерних наук до розв'язання складних задач комп'ютерної інженерії; застосовувати спеціалізовані концептуальні знання, що включають сучасні наукові здобутки у сфері комп'ютерної інженерії, необхідні для професійної діяльності, оригінального мислення та проведення досліджень, критичного осмислення проблем інформаційних технологій та на межі галузей знань; аналізувати проблематику, ідентифікувати та формулювати конкретні проблеми, що потребують вирішення, обирати ефективні методи їх вирішення; вирішувати задачі аналізу та синтезу комп'ютерних систем та мереж; застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів комп'ютерних систем та мереж для вирішення складних задач комп'ютерної інженерії та дотичних проблем; розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем; планувати і виконувати наукові дослідження в сфері комп'ютерної інженерії, формулювати і перевіряти гіпотези, обирати методики та інструменти, аналізувати результати, обґрунтовувати висновки; вміти виконувати експериментальні дослідження за професійною тематикою; вміти проєктувати та розробляти системне та прикладне програмне забезпечення, включаючи інтерфейси користувача програмних систем, враховуючи його призначення, необхідний рівень якості та зручність використання (usability).

Зміст навчальної дисципліни. Архітектурні особливості сучасних спеціалізованих обчислювальних систем, зокрема реального часу, Інтернету речей, кіберфізичних, мобільних, робототехнічних, розподілених, а також теорію проєктування спеціалізованих ОС для цих систем, приклади та шляхи розв'язку наукових задач з теорії проєктування спеціалізованих операційних систем; засади драйверної концепції як основи взаємодії операційної системи (користувача) з пристроями (системними/периферійними, реальними/віртуальними тощо), принципи роботи драйверної концепції та теорію і технології проєктування драйверів з використанням цієї концепції; проміжне програмне забезпечення для розподілених систем та його місце в загальній системі системного програмного забезпечення; концепція блокчейн технології; проблемні наукові задачі в системі реалізації паралельних обчислень та шляхи їх розвитку; балансування навантаження та забезпечення відмовостійкості в розподілених мережах.

Запланована навчальна діяльність: лекції – 18 год., лабораторні заняття – 36 год., самостійна робота – 96 год.; разом – 150 год.

Методи навчання: словесні, наочні, проблемно-пошукові (лекції); пояснювально-ілюстративні, практичні, проблемно-пошукові, частково-пошукові (лабораторні заняття), практичні, дослідницькі, частково-пошукові (самостійна робота: індивідуальні завдання).

Форми і методи оцінювання результатів навчання: захисти лабораторних робіт, тестовий контроль, підсумковий контрольний захід.

Вид семестрового контролю: іспит.

Навчальні ресурси:

1. Tanenbaum, Andrew S.; Woodhull, Albert S. Operating systems: design and implementation. Englewood Cliffs: Prentice Hall, 2017, 1120 p.
2. Simmonds, Chris. Mastering Embedded Linux Programming. Packt Publishing Ltd, 2017, 360 p.
3. Cooling, Jim. Real-time Operating Systems Book 1: The Foundations (The engineering of real-time embedded systems). Independently published, 2018, 296 p.
4. Silberschatz, A., Galvin, P.B., Gagne, G. Operating system concepts essentials / A. Silberschatz, P.B. Galvin, G. Gagne. – John Wiley & Sons, Inc., 2019, 784 p.
5. Wang, K. C. Embedded real-time operating systems. In: Embedded and Real-Time Operating Systems. Springer, Cham, 2017. p. 401-475.
6. Rivnak, M., Deligiannidis, L. Real-Time Operating Systems: Course Development. In Advances in Security, Networks, and Internet of Things – Springer, Cham, 2021.
7. Модульне середовище для навчання MOODLE. Доступ до ресурсу: <https://msn.khnu.km.ua/course/view.php?id=3663>.
8. Електронна бібліотека університету. Доступ до ресурсу: http://lib.khnu.km.ua/asp/php_f/page_lib.php.

Викладач: старший викладач Денисюк Дмитро Олександрович

1. СТРУКТУРА ЗАЛКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ

| Назва теми | Кількість годин, відведених на: | | |
|---|---------------------------------|--------------------|-------------------|
| | лекції | лабораторні роботи | самостійну роботу |
| Тема 1. Вступ. Спеціалізовані ОС для систем реального часу (real-time operating system, RTOS) | 6 | 16 | 44 |
| Тема 2. Спеціалізовані ОС для мікроконтролерних пристроїв і робототехнічних систем | 4 | 4 | 11 |
| Тема 3. Драйвери пристроїв | 4 | 8 | 22 |
| Тема 4. Проміжне програмне забезпечення для розподілених та паралельних обчислювальних систем | 4 | 8 | 19 |
| Разом за семестр: | 18 | 36 | 96 |

2. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

2.1. Зміст лекційного курсу

| Номер лекції | Перелік змістових модулів, тем лекцій, їх анотації | Кількість годин |
|--------------|--|-----------------|
| 1. | <p>Тема 1, Лекція 1. Тема: Вступ. Концептуальні моделі сучасних спеціалізованих операційних систем. Класифікація сучасних спеціалізованих ОС. Місце спеціалізованих операційних систем в загальній класифікації ОС. Дослідження архітектурних особливостей сучасних спеціалізованих обчислювальних систем. Типові складові спеціалізованих ОС. Теоретичні основи операційних систем, зокрема спеціалізованих. Застосування спеціалізованих операційних систем. Апаратно-програмні пристрої в системі задач проектування спеціалізованих ОС. Дослідження проблематики та актуальності проектування спеціалізованих операційних систем. Задачі та проблеми підвищення надійності, безпеки та керування пам'яттю в сучасних спеціалізованих операційних системах. Приклади та шляхи розв'язку наукових задач з теорії проектування спеціалізованих операційних систем. Дослідження методик та інструментів для проектування спеціалізованих операційних систем. Літ.: [1-5]</p> | 2 |
| 2. | <p>Тема 1, Лекція 2. Тема: Теорія і технології проектування спеціалізованих операційних систем для керування і обробки даних в режимі реального часу. Моделі систем реального часу. Задачі і технології систем реального часу та методи їх реалізації. Апаратно-програмні пристрої. Інтерфейси апаратно-програмних пристроїв. Моделі і стандарти інтерфейсів. Літ.: [1, 6, 7]</p> | 2 |
| 3. | <p>Тема 1, Лекція 3. Тема: Концепції та технології проектування ОС Інтернету речей, ОС кіберфізичних систем, мобільних ОС. Моделі систем Інтернету речей, кіберфізичних систем, мобільних систем та</p> | 2 |

| | | |
|----|--|----|
| | спеціалізованих ОС як їх складових в процесі синтезу систем. Методи проєктування спеціалізованих операційних систем для Інтернету речей. Дослідження проблематики гетерогенності та керування ресурсами в операційних системах Інтернету речей. Методи проєктування спеціалізованих операційних систем для мобільних систем. Методи проєктування спеціалізованих операційних систем для кіберфізичних систем. Літ.: [8-13] | |
| 4 | Тема 2, Лекція 4. Тема: Керування виконавчими пристроями в режимі реального часу по інтерфейсу USB. Моделі передачі даних. Літ.: [1] | 2 |
| 5. | Тема 2, Лекція 5. Тема: Робототехнічні системи та спеціалізовані ОС для них. Моделі робототехнічних систем. Технології штучного інтелекту в робототехнічних системах. Дослідження стратегій розвитку робототехнічних операційних систем на прикладі сучасних наукових задач. Літ.: [1, 14, 15] | 2 |
| 6. | Тема 3, Лекція 6. Тема: Драйверна концепція. Теорія і технології проєктування драйверів для ОС Windows. Типи драйверів. Драйвери режиму ядра та режиму користувача. Віртуальна модель пристрою та його інтерфейсу. Human Interface Devices, HID. Network Driver Interface Specification, NDIS. Драйвери протоколів. Літ.: [16] | 2 |
| | Тема 3, Лекція 7. Тема: Теорія і технології проєктування драйверів для ОС Linux. Технології проєктування драйверів для ОС Linux. Компіляція драйверів з ядром. Реалізація драйверів як модулів ядра. Літ.: [17-20] | 2 |
| 8. | Тема 4, Лекція 8. Тема: Проміжне програмне забезпечення для розподілених систем та його місце в загальній системі системного програмного забезпечення. Принципи створення розподілених систем. Теорія розподілених обчислень. Проблемні наукові задачі в системі здійснення розподілених обчислень. Проміжне програмне забезпечення для розподілених систем: принципи, засоби створення, зв'язок з операційними системами. Літ.: [21-23] | 2 |
| 9. | Тема 4, Лекція 9. Тема: Концепція блокчейн технології. Принципи побудови блокчейн-мереж. Життєвий цикл транзакції. Надійність та моделі розгортання блокчейну. Паралельні обчислення. Проблемні наукові задачі в системі реалізації паралельних обчислень. Балансування навантаження серверів Linux. HAProxy, Keepalived: ПЗ для забезпечення відмовостійкості мережних сервісів та балансування навантаження. Nginx. Літ.: [24-27] | 2 |
| | Разом за семестр: | 18 |

2.2. Зміст лабораторних занять

| № п/п | Тема лабораторного заняття | Кількість годин |
|-------|--|-----------------|
| 1. | Лабораторна робота 1. Тема: Проектування спеціалізованої RTOS для Інтернету речей на платформі Raspberry Pi. Створення макету проекту. Додавання макросів для виведення налагоджувальних повідомлень. Обгортання глобальних змінних у примітиві синхронізації. Тестування проекту в емуляторі QEMU. | 4 |
| 2. | Лабораторна робота 2. Тема: Проектування спеціалізованої RTOS для Інтернету речей на платформі Raspberry Pi. Реалізація базових функцій ядра. Додавання до проекту драйверів GPIO та UART та перенесення його на цільову платформу. Реалізація базових функцій ядра RTOS. Додавання функцій апаратного таймера для усунення циклічних затримок в драйвері GPIO. Тестування на реальній платформі. | 4 |
| 3. | Лабораторна робота 3. Тема: Проектування спеціалізованої RTOS для Інтернету речей на платформі Raspberry Pi. Реалізація рівнів привілеїв. Реалізація функції завантаження ядра по UART. Апаратне налагодження по JTAG. Реалізація рівнів привілеїв. Тестування на реальній платформі. | 4 |
| 4. | Лабораторна робота 4. Тема: Проектування спеціалізованої RTOS для Інтернету речей на платформі Raspberry Pi. Реалізація функцій обробки виняткових ситуацій. Розмежування адресного простору ядра і користувача. Реалізація функцій обробки виняткових ситуацій. Комплексне тестування на реальній платформі. | 4 |
| 5. | Лабораторна робота 5. Тема: Проектування вбудованої RTOS для платформи Lego EV3 на базі ОС Linux. Модифікація базового ядра Linux для забезпечення функцій RTOS. Дослідження роботи RTOS на платформі Lego EV3. | 4 |
| 6.* | Лабораторна робота 6. Тема: Розроблення драйверів пристроїв в ОС Windows. Kernel-Mode Driver Framework. User-Mode Driver Framework. Створення, розгортання, тестування та налагодження драйверів периферійних та віртуальних пристроїв. | 4 |
| 7.* | Лабораторна робота 7. Тема: Розроблення драйверів пристроїв як модулів ядра для ОС Linux. Створення драйвера як модуля ядра. Збирання ядра разом з модулем драйвера. | 4 |
| 8. | Лабораторна робота 8. Тема: Реалізація блокчейн технологій. Побудова блокчейну для нефінансових операцій. | 4 |
| 9. | Підсумкове заняття | 4 |
| | Разом за семестр: | 36 |

Примітка: * Лабораторна робота може бути зарахована за наявності сертифікатів з проходження курсів (Udemy, Linux Kernel Driver Development, <https://www.udemy.com/course/linux-kernel-driver-development/> та ін.)

2.3. Зміст самостійної (індивідуальної) роботи

Обсяг самостійної роботи з дисципліни становить 96 годин. Він включає опрацювання лекційного матеріалу, підготовку до виконання лабораторних робіт і їх захисту, підготовку до поточного контролю.

| Номер тижня | Вид самостійної роботи | К-ть годин |
|-------------|---|------------|
| 1-2 | Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №1. Самостійне опрацювання теоретичного матеріалу. Самостійна робота над розробкою програми до лабораторної роботи №1. Підготовка до захисту лабораторної роботи №1 | 11 |
| 3-4 | Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №2. Самостійне опрацювання теоретичного матеріалу. Самостійна робота над розробкою програми до лабораторної роботи №2. Підготовка до захисту лабораторної роботи №2 | 11 |
| 5-6 | Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №3. Самостійне опрацювання теоретичного матеріалу. Самостійна робота над розробкою програми до лабораторної роботи №3. Підготовка до захисту лабораторної роботи №3 | 11 |
| 7-8 | Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №4. Самостійне опрацювання теоретичного матеріалу. Самостійна робота над розробкою програми до лабораторної роботи №4. Підготовка до захисту лабораторної роботи №4 | 11 |
| 9-10 | Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №5. Самостійне опрацювання теоретичного матеріалу. Самостійна робота над розробкою програми до лабораторної роботи №5. Підготовка до захисту лабораторної роботи №5 | 11 |
| 11-12 | Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №6. Самостійне опрацювання теоретичного матеріалу. Самостійна робота над розробкою програми до лабораторної роботи №6. Підготовка до захисту лабораторної роботи №6 | 11 |
| 13-14 | Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №7. Самостійне опрацювання теоретичного матеріалу. Самостійна робота над розробкою програми до лабораторної роботи №7. Підготовка до захисту лабораторної роботи №7 | 11 |
| 15-16 | Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №8. Самостійне опрацювання теоретичного матеріалу. Самостійна робота над розробкою програми до лабораторної роботи №8. Підготовка до захисту лабораторної роботи №8 | 11 |
| 17 | Підготовка до іспиту | 8 |
| | Разом за семестр: | 96 |

3. МЕТОДИ НАВЧАННЯ

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних методів. Зокрема, лекції проводяться в основному, з використанням словесних, наочних, проблемно-пошукових методів. Лабораторні заняття проводяться пояснювально-ілюстративними методами, практичними, проблемно-пошуковими та частково-пошуковими методами; самостійна робота передбачає виконання індивідуальних завдань із залученням практичних, дослідницьких, частково-пошукових методів та з

використанням інформаційних технологій і мають за мету набуття студентами практичних навичок з вирішення задач з використанням програмування під різні типи операційних систем.

4. ФОРМИ І МЕТОДИ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Поточний контроль здійснюється під час лекційних та лабораторних занять. Семестровий контроль проводиться у формі іспиту. При цьому при виведенні остаточної оцінки враховуються результати поточного контролю.

Кожний вид роботи з дисципліни оцінюється за *чотирибальною* шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів навчальних робіт, виконаних і зданих *позитивно* з врахуванням коефіцієнта вагомості. Студент, який набрав позитивний середньозважений бал за поточну роботу і не здав підсумковий контрольний захід (іспит), вважається невстигаючим.

При оцінюванні знань студентів використовуються різні засоби контролю, зокрема: усне опитування перед допуском до виконання лабораторної роботи – здійснюється на її початку; якість виконання, набуття теоретичних знань і практичних навичок перевіряється шляхом захисту кожної лабораторної роботи згідно з робочою програмою дисципліни і робочим навчальним планом.

При оцінюванні знань студентів викладач керується наступними критеріями.

Оцінку "відмінно" отримує студент за глибоке і повне опанування змісту навчального матеріалу, в якому він легко орієнтується, понятійного апарату, за уміння зв'язувати теорію з практикою, вирішувати практичні завдання, висловлювати і обґрунтовувати свої судження. Відмінна оцінка передбачає грамотний, логічний виклад відповіді (як в усній, так і в письмовій формі), якісне зовнішнє оформлення. Студент повинен набути практичних навичок із матеріалу за програмою курсу.

Оцінка "відмінно" виставляється студенту, який глибоко засвоїв матеріал, вміє його раціонально застосувати. Студент не повинен вагатися при видозміні запитання, повинен робити детальні та узагальнюючі висновки.

Оцінку "добре" отримує студент за повне засвоєння навчального матеріалу, володіння понятійним апаратом, орієнтування в вивченому матеріалі, свідоме використання знань для вирішення практичних завдань, грамотний виклад відповіді, але у змісті і формі відповіді мали місце окремі неточності (похибки), нечіткі формулювання закономірностей тощо. Відповідь студента повинна будуватись на основі самостійного мислення.

Оцінку "добре" отримує студент за правильну відповідь з однією суттєвою помилкою.

Оцінки "задовільно" заслуговує студент, який виявив знання основного навчально-програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, що справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь студента будується на рівні репродуктивного мислення, студент слабо знає структуру курсу, допускає помилки у відповіді, засвоїв і набув практичних навичок у складанні програм, але допустив неточності. Вагається при відповіді на видозмінене запитання, разом з тим студент володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.

Оцінки "задовільно" заслуговує студент за неповне опанування матеріалу, але отримані знання і набуті практичні навички із розроблення блоків операційних систем та драйверів.

Оцінка "незадовільно" виставляється, коли студент має розрізнені, безсистемні знання, не вміє виділяти головне і другорядне, допускається помилок у визначенні понять, перекручує їх зміст, хаотично і невпевнено викладає матеріал, не може використовувати знання при вирішенні практичних завдань. Як правило, оцінка "незадовільно" виставляється студенту, який не може продовжити навчання без додаткових знань з курсу.

Оцінка "незадовільно" виставляється студенту за повне незнання і нерозуміння навчального матеріалу або відмову від відповіді і передбачає повторне навчання студента з дисципліни.

Кожний вид роботи оцінюється за чотирибальною шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів робіт.

Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів у семестрі за ваговими коефіцієнтами

| Аудиторна робота | | | | | | | | Тестовий контроль | Форма підсумкового контролю |
|-----------------------|---|---|---|-----|---|---|---|-------------------|-----------------------------|
| Лабораторні роботи №: | | | | | | | | Тест | Іспит |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | |
| ВК: | | | | 0,5 | | | | 0,1 | 0,4 |

Примітка: ВК – ваговий коефіцієнт.

Для переходу від вітчизняної оцінки до оцінки за шкалою ECTS необхідно знайти середньоарифметичну оцінку за вітчизняною шкалою, помножити її на відповідний ваговий коефіцієнт і, додавши всі складові, отримаємо суму балів, які визначають конкретну оцінку ECTS.

Перехід від вітчизняної шкали оцінювання до європейської (ECTS)

| Оцінка ECTS | Бали | Вітчизняна оцінка | |
|-------------|------------|-------------------|--|
| A | 4,75-5,00 | 5 | ВІДМІННО – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навиків |
| B | 4,25-4,74 | 4 | ДОБРЕ – повне знання навчального матеріалу з кількома незначними помилками |
| C | 3,75-4,24 | 4 | ДОБРЕ – в загальному правильна відповідь з однією суттєвою помилкою |
| D | 3,25-3,74 | 3 | ЗАДОВІЛЬНО – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією |
| E | 3,00-3,24 | 3 | ЗАДОВІЛЬНО – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання |
| FX | 2,00 -2,99 | 2 | НЕЗАДОВІЛЬНО – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни |
| F | 0,00-1,99 | 2 | НЕЗАДОВІЛЬНО – необхідна серйозна подальша робота і повторне вивчення дисципліни |

Оцінка за іспит виставляється при отриманні студентом з дисципліни від 3,00 до 5,00 балів. При цьому за вітчизняною шкалою ставиться відповідна кількість балів, а за шкалою ECTS – оцінка, що відповідає набраній студентом кількості балів.

5. ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Класифікація сучасних спеціалізованих ОС.
2. Місце спеціалізованих операційних систем в загальній класифікації ОС.
3. Дослідження архітектурних особливостей сучасних спеціалізованих обчислювальних систем.
4. Типові складові спеціалізованих ОС.
5. Теоретичні основи операційних систем, зокрема спеціалізованих.
6. Застосування спеціалізованих операційних систем.
7. Апаратно-програмні пристрої в системі задач проєктування спеціалізованих ОС.
8. Дослідження проблематики та актуальності проєктування спеціалізованих операційних систем.
9. Задачі та проблеми підвищення надійності, безпеки та керування пам'яттю в сучасних спеціалізованих операційних системах.
10. Приклади та шляхи розв'язку наукових задач з теорії проєктування спеціалізованих операційних систем.
11. Дослідження методик та інструментів для проєктування спеціалізованих операційних систем.
12. Моделі систем реального часу.
13. Задачі і технології систем реального часу та методи їх реалізації.
14. Апаратно-програмні пристрої.
15. Інтерфейси апаратно-програмних пристроїв.
16. Моделі і стандарти інтерфейсів апаратно-програмних пристроїв.
17. Моделі систем Інтернету речей.
18. Моделі кіберфізичних систем.
19. Моделі мобільних систем.
20. Моделі спеціалізованих ОС як складових в процесі синтезу систем.
21. Методи проєктування спеціалізованих операційних систем для Інтернету речей.
22. Дослідження проблематики гетерогенності та керування ресурсами в операційних системах Інтернету речей.
23. Методи проєктування спеціалізованих операційних систем для мобільних систем.
24. Методи проєктування спеціалізованих операційних систем для кіберфізичних систем.
25. Моделі передачі даних.
26. Моделі робототехнічних систем.
27. Технології штучного інтелекту в робототехнічних системах.
28. Стратегії розвитку робототехнічних операційних систем на прикладі сучасних наукових задач.
29. Типи драйверів.
30. Драйвери режиму ядра та режиму користувача.
31. Віртуальна модель пристрою та його інтерфейсу.
32. Human Interface Devices, HID.
33. Network Driver Interface Specification, NDIS.
34. Драйвери протоколів.
35. Потік даних в KMDF / UMDF.
36. Керування живленням і модель вводу / виводу в WDF.
37. Setup application programming interface (SetupAPI).
38. Особливості моделі введення / виведення UMDF (KMDF).

39. Об'єктна модель WDF.
40. Об'єктна модель KMDF.
41. Об'єкти KMDF. Життєвий цикл об'єкта KMDF.
42. Об'єкти UMDF. Життєвий цикл UMDF-драйвера.
43. Windows Driver Foundation (WDF).
44. Структура драйвера.
45. Основні функції для побудови драйвера.
46. Особливості вибору моделі драйвера для ОС Windows.
47. Технології проєктування драйверів для ОС Linux.
48. Компіляція драйверів з ядром.
49. Реалізація драйверів як модулів ядра.
50. Принципи створення розподілених систем.
51. Теорія розподілених обчислень.
52. Проблемні наукові задачі в системі здійснення розподілених обчислень.
53. Проміжне програмне забезпечення для розподілених систем: принципи, засоби створення, зв'язок з операційними системами.
54. Принципи побудови блокчейн-мереж.
55. Життєвий цикл транзакції.
56. Надійність та моделі розгортання блокчейну.
57. Паралельні обчислення.
58. Проблемні наукові задачі в системі реалізації паралельних обчислень.
59. Балансування навантаження серверів Linux.
60. HAProxy.
61. Keepalived.
62. Nginx.

6. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Бобровнікова К.Ю. Теорія і технології проєктування спеціалізованих операційних систем. Методичні вказівки для виконання лабораторних робіт для здобувачів вищої освіти за освітньо-науковою програмою «Комп'ютерна інженерія та програмування» спеціальності 123 «Комп'ютерна інженерія» другого (магістерського) рівня вищої освіти. Хмельницький: ХНУ, 2021. – 99 с.

7.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна література

1. Tanenbaum, Andrew S.; Woodhull, Albert S. Operating systems: design and implementation. Englewood Cliffs: Prentice Hall, 2017, 1120 p.
2. Marufuzzaman, M., Al Karim, S., Rahman, M. S., Zahid, N. M., & Sidek, L. M. A review on reliability, security and memory management of numerous operating systems. Indonesian Journal of Electrical Engineering and Informatics (IJEI), 7(3), 2019, pp. 577-585.
3. Szopa, A. Comparison Analysis of Accessible Features Built into Operating Systems. In International Conference on Applied Human Factors and Ergonomics – Springer, Cham, 2021.
4. Silberschatz, A., Galvin, P.B., Gagne, G. Operating system concepts essentials. John Wiley & Sons, Inc., 2019, 784 p.
5. Malallah, H., Zeebaree, S. R., Zebari, R. R., Sadeeq, M. A., Ageed, Z. S., Ibrahim, I. M., ... & Merceedi, K. J. A comprehensive study of kernel (issues and concepts) in different operating systems. Asian Journal of Research in Computer Science, 2021, pp. 16-31.

6. Erciyas, K. Distributed Real-Time Operating Systems and the Middleware. In *Distributed Real-Time Systems*. Springer, Cham, 2019, pp. 119-148.
7. Serino, A., & Cheng, L. Real-Time Operating Systems for Cyber-Physical Systems: Current Status and Future Research. In *2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, 2020, pp. 419-425. IEEE.
8. Agarwal, V. K., Kumar, N., Singh, R., Pathak, A., Bhushan, B. Operating Systems Support and Network Optimization View of Internet of Things. In *Emerging Technologies in Data Mining and Information Security – Springer, Singapore*, 2021.
9. Bansal, S., & Kumar, D. IoT ecosystem: A survey on devices, gateways, operating systems, middleware and communication. *International Journal of Wireless Information Networks*, Springer, Cham, 2020, pp. 1-25.
10. Javed, F., Afzal, M. K., Sharif, M., & Kim, B. S. Internet of Things (IoT) operating systems support, networking technologies, applications, and challenges: A comparative review. *IEEE Communications Surveys & Tutorials*, 20(3), 2018, pp. 2062-2100.
11. Musaddiq, A., Zikria, Y. B., Hahm, O., Yu, H., Bashir, A. K., & Kim, S. W. A survey on resource management in IoT operating systems. *IEEE Access*, 6, 2018, pp. 8459-8482.
12. Rivnak, M., Deligiannidis, L. Real-Time Operating Systems: Course Development. In *Advances in Security, Networks, and Internet of Things – Springer, Cham*, 2021.
13. Korchi, A., Khachouch, M. K., Lakhrissi, Y., & Moumen, A. Classification of existing mobile cross-platform approaches. In *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE) IEEE*, 2020, pp. 1-5.
14. Dawarka, V., & Bekaroo, G. Building and evaluating cloud robotic systems: A systematic review. *Robotics and Computer-Integrated Manufacturing*, 73, 2022, 102240.
15. Wang, K. C. Embedded real-time operating systems. In: *Embedded and Real-Time Operating Systems*. Springer, Cham, 2017. p. 401-475.
16. Windows Driver Kit documentation [Электронный ресурс] : [Интернет-портал]. – Электронні дані. – [© Microsoft, 2018]. – Режим доступу: <https://docs.microsoft.com/en-us/windows-hardware/drivers/> (дата звернення 22.09.2018). – Назва з екрана.
17. Love, R. *The Linux kernel: a description of the development process*. Robert Love, 2019, 496 p.
18. Granneman, Scott. *Linux phrasebook*. Sams Publishing, 2019, 416 p.
19. Simmonds, Chris. *Mastering Embedded Linux Programming*. Packt Publishing Ltd, 2017, 360 p.
20. Cooling, Jim. *Real-time Operating Systems Book 1: The Foundations (The engineering of real-time embedded systems)*. Independently published, 2018, 296 p.
21. Balador, A., Ericsson, N., Bakhshi, Z. Communication middleware technologies for industrial distributed control systems: A literature review. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2017, IEEE, pp. 1-6.
22. Gascon-Samson, J., Rafiuzzaman, M., Pattabiraman, K. Thingsjs: Towards a flexible and self-adaptable middleware for dynamic and heterogeneous iot environments. In *Proceedings of the 4th Workshop on Middleware and Applications for the Internet of Things*, 2017, pp. 11-16.
23. Mohamed, N., Al-Jaroodi, J., Lazarova-Molnar, S., Jawhar, I. Middleware challenges for cyber-physical systems. *Scalable Computing: Practice and Experience*, 18(4), 2017, pp. 331-346.
24. Shukur, H., Zeebaree, S. R., Ahmed, A. J., Zebari, R. R., Ahmed, O., Tahir, B. S. A., Sadeeq, M. A. A state of art survey for concurrent computation and clustering of parallel computing for distributed systems. *Journal of Applied Science and Technology Trends*, 1(4), 2020, pp. 148-154.

25. LU, Yang. The blockchain: State-of-the-art and research challenges. *Journal of Industrial Information Integration*, 2019, 15, pp. 80-90.
26. Trobec, R., Slivnik, B., Bulic, P., Robic, B. *Introduction to Parallel Computing: From Algorithms to Programming on State-of-Art Platforms*, 2018, Springer.
27. Erciyes, K. (2019). Distributed real-time systems. In *Distributed Real-Time Systems* (pp. 41-62). Springer, Cham.

8. ІНФОРМАЦІЙНІ РЕСУРСИ

Модульне середовище для навчання MOODLE. Доступ до ресурсу:
<https://msn.khnu.km.ua/course/view.php?id=3663>