

ЗАТВЕРДЖУЮ

Дека́н факультету _____ ФІТ



Тетяна ГОВОРУЩЕНКО
Ім'я, ПРІЗВИЩЕ

2025 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Теорія і технології проєктування спеціалізованих операційних систем
Назва дисципліни

Галузь знань – F Інформаційні технології
Спеціальність – F7 Комп'ютерна інженерія та програмування
Рівень вищої освіти – Другий (магістерський)
Освітньо-професійна програма – Комп'ютерна інженерія та програмування
Обсяг дисципліни – 5 кредитів ЄКТС, Шифр дисципліни – ОФП.07
Мова навчання – українська
Статус дисципліни: обов'язкова (науково-дослідна підготовка)
Факультет – Інформаційних технологій
Кафедра – Комп'ютерної інженерії та інформаційних систем

Форма здобуття освіти	Курс	Семестр	Загальний обсяг		Кількість годин						Курсовий проєкт*	Курсова робота	Форма семестрового контролю	
					Аудиторні заняття					Самостійна робота, у т.ч. ІРС			Залік	Іспит
			Кредити ЄКТС	Години	Разом	Лекції	Лабораторні роботи	Практичні заняття	Семінарські заняття					
Д	1	2	5	150	50	16	34			100				+
Разом ДФН			5	150	50	16	34			100				1

Примітка. *З навчальної дисципліни у 2 семестрі передбачений курсовий проєкт, зміст та вимоги до виконання якого регулюються відповідними методичними рекомендаціями.

Робоча програма складена на основі освітньо-професійної програми «Комп'ютерна інженерія та програмування» за спеціальністю F7 «Комп'ютерна інженерія та програмування»

Робоча програма складена _____
Підпис автора(ів) _____ ст.викладач Дмитро ДЕНИСЮК
Науковий ступінь, вчене звання, Ім'я, ПРІЗВИЩЕ автора(ів)

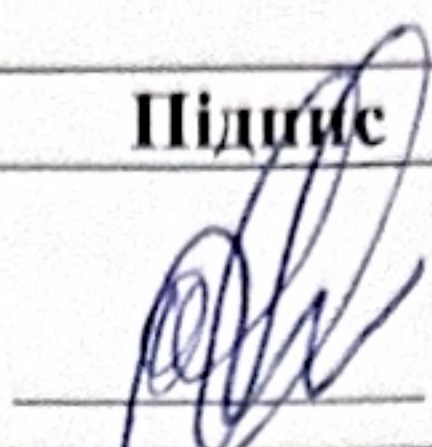
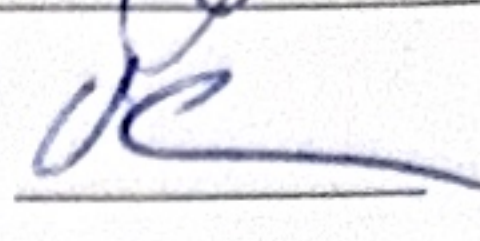
Схвалена на засіданні кафедри _____
Комп'ютерної інженерії та інформаційних систем

Протокол від _____18.08.2025_____ № _____1_____ Зав. кафедри _____
Підпис _____ Ольга ПАВЛОВА
Ім'я, ПРІЗВИЩЕ

Робоча програма розглянута та схвалена вченою радою факультету інформаційних технологій

Голова вченої ради факультету _____
Підпис _____ Тетяна ГОВОРУЩЕНКО
Ім'я, ПРІЗВИЩЕ

ЛИСТ ПОГОДЖЕННЯ

Посада	Назва кафедри	Підпис	Ініціали, прізвище
Завідувач кафедри, доктор філософії.	Комп'ютерної інженерії та інформаційних систем		Ольга ПАВЛОВА
Гарант освітньо-професійної програми, д.т.н., професор.	Комп'ютерної інженерії та інформаційних систем		Олег САВЕНКО

3. Пояснювальна записка

Дисципліна «Теорія і технології проектування спеціалізованих операційних систем» є однією із дисциплін циклу науково-дослідної підготовки і займає провідне місце у підготовці здобувачів другого (магістерського) рівня вищої освіти, очної (денної) форми здобуття вищої освіти, які навчаються за освітньою програмою «Комп'ютерна інженерія та програмування» в межах спеціальності 123 «Комп'ютерна інженерія».

Пререквізити – Іноземна мова за професійним спрямуванням (ОЗП.01), Філософські проблеми наукового пізнання (ОЗП.02), Методологія та організація наукових досліджень (ОЗП.04), Комп'ютерні системи штучного інтелекту (ОФП.01), Технології проектування програмних систем (ОФП.02), Технології проектування програмних систем (курсний проєкт) (ОФП.03), Теорія і проектування комп'ютерних та кіберфізичних систем і мереж (ОФП.04).

Постреквізити – Методологія забезпечення безпеки та захисту комп'ютерних систем (ОФП.06), Теорія, проектування та моделювання спеціалізованих комп'ютерних систем (ОФП.08), Науково-дослідна практика (ОФП.09), Кваліфікаційна робота (ОФП.10).

Відповідно до освітньої програми дисципліна сприяє забезпеченню:

компетентностей:

Інтегральна компетентність (ІК) – 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 їх 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 ІТ-інфраструктур.

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 проблеми в галузі комп'ютерних та інформаційних технологій, визначати їх обмеження.

програмних результатів навчання: застосовувати загальні підходи пізнання, методи математики, природничих та інженерних наук до розв'язання складних задач комп'ютерної інженерії, пов'язаних із проектуванням спеціалізованих операційних систем (ПРН 1);

застосовувати спеціалізовані концептуальні знання, що включають сучасні наукові здобутки у сфері комп'ютерної інженерії, необхідні для професійної діяльності, оригінального мислення та проведення досліджень у галузі теорії і технологій проектування операційних систем (ПРН 4);

аналізувати проблематику, ідентифікувати та формулювати конкретні проблеми у сфері створення спеціалізованих операційних систем, обирати ефективні методи їх вирішення (ПРН 6);

вирішувати задачі аналізу та синтезу комп'ютерних систем і мереж при проектуванні спеціалізованих операційних систем та їх компонентів (ПРН 7);

застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів для розроблення операційних систем реального часу, кіберфізичних, мобільних, робототехнічних і розподілених систем (ПРН 8);

розробляти програмне забезпечення для вбудованих, розподілених, мобільних і гібридних систем із використанням методів проектування спеціалізованих операційних систем і драйверів пристроїв (ПРН 9);

планувати і виконувати наукові дослідження в галузі теорії та технологій проектування спеціалізованих операційних систем, формулювати і перевіряти гіпотези, аналізувати результати, обґрунтовувати висновки (ПРН 14);

виконувати експериментальні дослідження, моделювання та тестування спеціалізованих операційних систем і проміжного програмного забезпечення для розподілених обчислень (ПРН 15);

проектувати та розробляти системне і прикладне програмне забезпечення, включаючи інтерфейси користувача спеціалізованих операційних систем, враховуючи їх призначення, якість і зручність використання (ПРН 17).

Мета дисципліни. Формування у здобувачів вищої освіти компетентностей, необхідних для розв'язання складних спеціалізованих задач у сфері комп'ютерної інженерії, пов'язаних із проектуванням, розробленням та дослідженням спеціалізованих операційних систем для обчислювальних комплексів різного призначення. Дисципліна спрямована на розвиток у студентів умінь аналізувати архітектури сучасних операційних систем, застосовувати теорію і технології їх проектування для систем реального часу, кіберфізичних, мобільних, робототехнічних, розподілених та вбудованих систем, розробляти драйвери пристроїв і проміжне програмне забезпечення, а також здійснювати наукові дослідження у галузі створення надійних, безпечних і ефективних операційних систем нового покоління.

Предмет дисципліни. Архітектурні особливості, принципи побудови, методи та технології проектування спеціалізованих операційних систем для обчислювальних комплексів різного призначення, зокрема систем реального часу, Інтернету речей, кіберфізичних, мобільних, робототехнічних та розподілених систем. Предметом також є теорія та практика створення драйверів пристроїв, проміжного програмного забезпечення для розподілених і паралельних обчислень, а також дослідження проблем надійності, відмовостійкості й ефективності сучасних операційних систем.

Завдання дисципліни. Формування у здобувачів вищої освіти знань, умінь і практичних навичок з аналізу, проектування, розроблення та дослідження спеціалізованих операційних систем. Дисципліна спрямована на опанування архітектурних особливостей сучасних спеціалізованих обчислювальних систем, зокрема систем реального часу, Інтернету речей, кіберфізичних, мобільних, робототехнічних і розподілених систем, а також на розвиток умінь створювати компоненти операційних систем і драйвери пристроїв, проектувати проміжне програмне забезпечення для розподілених і паралельних обчислень, забезпечувати їхню надійність, безпеку та відмовостійкість. Особлива увага приділяється формуванню дослідницьких компетентностей у сфері розроблення та впровадження сучасних технологій проектування спеціалізованих операційних систем.

Результати навчання. Після вивчення дисципліни студент повинен володіти професійною термінологією та базовими концепціями теорії й технологій проектування спеціалізованих операційних систем; розуміти архітектурні принципи побудови, функціонування та взаємодії компонентів операційних систем різного призначення, зокрема систем реального часу, кіберфізичних, мобільних, робототехнічних і розподілених систем; уміти аналізувати, проектувати та розробляти компоненти операційних систем і драйвери пристроїв, застосовувати методи балансування навантаження, забезпечення відмовостійкості та безпеки; здійснювати моделювання, тестування й оцінювання ефективності операційних систем і проміжного програмного забезпечення; обґрунтовувати вибір технологій і методів розроблення з урахуванням сучасних тенденцій комп'ютерної інженерії; виконувати дослідницькі та експериментальні завдання, інтерпретувати результати і формулювати висновки щодо оптимізації архітектури спеціалізованих операційних систем.

4. Структура залікових кредитів дисципліни

Назва розділу (теми)	Кількість годин, відведених на:		
	Лекції	Лабораторні заняття	СРС
Тема 1. Вступ. Спеціалізовані ОС для систем	4	8	25
Тема 2. Спеціалізовані ОС для мікроконтролерних пристроїв і робототехнічних систем	6	10	25
Тема 3. Драйвери пристроїв	4	8	25
Тема 4. Проміжне програмне забезпечення для розподілених та паралельних обчислювальних систем	2	8	25
Разом:	16	34	100

5. Програма навчальної дисципліни

5.1 Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
	<i>Тема 1. Вступ. Спеціалізовані ОС для систем</i>	4
1	Лекція 1. Вступ. Концептуальні моделі сучасних спеціалізованих операційних систем. Літ.: [1-5]	2
2	Лекція 2. Теорія і технології проектування спеціалізованих операційних систем для керування і обробки даних в режимі реального часу. Літ.: [1-5]	2
	<i>Тема 2. Спеціалізовані ОС для мікроконтролерних пристроїв і робототехнічних систем</i>	6
3	Лекція 3. Концепції та технології проектування ОС Інтернету речей, ОС кіберфізичних систем, мобільних ОС Літ.: [1-5]	2
4	Лекція 4. Керування виконавчими пристроями в режимі реального часу по інтерфейсу USB. Літ.: [1-5]	2
5	Лекція 5. Робототехнічні системи та спеціалізовані ОС для них Літ.: [1-5]	2
	<i>Тема 3. Драйвери пристроїв</i>	4
6	Лекція 6. Драйверна концепція. Теорія і технології проектування драйверів для ОС Windows. Літ.: [1-5]	2
7	Лекція 7. Теорія і технології проектування драйверів для ОС Linux. Літ.: [1-5]	2
	<i>Тема 4. Проміжне програмне забезпечення для розподілених та паралельних обчислювальних систем</i>	4
8	Лекція 7. Проміжне програмне забезпечення для розподілених систем та його місце в загальній системі системного програмного забезпечення. Літ.: [1-5]	2
9	Лекція 8. Концепція блокчейн технології. Літ.: [1-5]	2
	Разом:	16

5.2 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.	4
2	Лабораторна робота 2. Тема: Проектування спеціалізованої RTOS для Інтернету речей на платформі Raspberry Pi. Реалізація базових функцій ядра.	4
3	Лабораторна робота 3. Тема: Проектування спеціалізованої RTOS для Інтернету речей на платформі Raspberry Pi. Реалізація рівнів привілеїв.	4
4	Лабораторна робота 4. Тема: Проектування спеціалізованої RTOS для Інтернету речей на платформі Raspberry Pi. Реалізація функцій обробки виняткових ситуацій.	4
5	Лабораторна робота 5. Тема: Проектування вбудованої RTOS для платформи Lego EV3 на базі ОС Linux.	4
6*	Лабораторна робота 6. Тема: Розроблення драйверів пристроїв в ОС Windows. Kernel-Mode Driver Framework. User-Mode Driver Framework.	4
7*	Лабораторна робота 7. Тема: Розроблення драйверів пристроїв як модулів ядра для ОС Linux.	4
8	Лабораторна робота 8. Тема: Реалізація блокчейн технологій.	4
Разом:		34

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.)

5.3 Зміст самостійної (у т.ч. індивідуальної) роботи здобувача вищої освіти

На самостійне опрацювання студентів виносяться визначені у методичних рекомендаціях до лабораторних занять та самостійної роботи питання з кожної теми. Обсяг самостійної роботи (100 годин) включає опрацювання лекційного матеріалу, підготовку до виконання лабораторних робіт та їх захисту, а також підготовку до поточного контролю.

Номер тижня	Вид самостійної роботи	Кількість годин
1	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	10
2	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	10
3	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	10
4	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	10
5	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	10
6	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	10
7	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	10
8	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	10
9	Підготовка до іспиту	20
Разом 1-й семестр:		100

Керівництво самостійною роботою та контроль за її виконанням здійснюється викладачем згідно з розкладом консультацій у позаурочний час. Вимоги до виконання самостійної роботи, включаючи опрацювання теоретичного матеріалу та розробку програми до лабораторних робіт, викладені у Модульному середовищі для навчання (MOODLE) на сторінці навчальної дисципліни.

6. Технології та методи навчання

Процес навчання з дисципліни базується на поєднанні традиційних і сучасних технологій та методів викладання, спрямованих на формування у здобувачів освіти теоретичних знань і практичних умінь у сфері проектування спеціалізованих операційних систем.

Лекції проводяться з використанням мультимедійних презентацій, методів візуалізації, пояснення, проблемного й інтерактивного навчання, інформаційно-комунікаційних технологій, елементів мотивації та стимулювання пізнавальної активності студентів. Використовуються методи інтенсифікації та індивідуалізації навчального процесу.

Лабораторні заняття спрямовані на формування практичних навичок розроблення спеціалізованих операційних систем і драйверів пристроїв. Заняття проводяться із застосуванням методів інструктування, демонстрування, спостереження, аналізу кейсів, розв'язування ситуаційних і практичних завдань, ділових ігор, мозкових штурмів, тренінгових вправ і елементів дискусії. Акцент робиться на застосуванні комп'ютерного моделювання, проектної діяльності та використанні сучасних середовищ розроблення.

Самостійна робота здобувачів освіти передбачає опрацювання теоретичного матеріалу, підготовку до лабораторних робіт, виконання індивідуальних і домашніх завдань, підготовку до поточного та підсумкового контролю. Вона здійснюється з активним використанням інформаційно-комп'ютерних технологій і ресурсів дистанційного навчання (зокрема, навчального середовища Moodle).

7. Методи контролю

Поточний контроль здійснюється під час лекційних і лабораторних занять, а також у процесі проведення тестування відповідно до графіка освітнього процесу та робочої програми дисципліни, у тому числі з використанням модульного середовища для навчання Moodle.

Методи поточного контролю:

- усне опитування перед допуском до виконання лабораторної роботи;
- оцінювання результатів захисту лабораторних робіт;
- тестовий контроль засвоєння теоретичного матеріалу з окремих тем або розділів;
- оцінювання результатів виконання індивідуальних завдань;
- аналіз та оцінювання участі здобувача в обговоренні проблемних питань під час занять.

Підсумковий контроль проводиться у формі іспиту, який охоплює зміст усього навчального матеріалу дисципліни. Іспит проводиться за білетами, розробленими викладачем і затвердженими на засіданні кафедри комп'ютерної інженерії та інформаційних систем.

При виставленні підсумкової семестрової оцінки враховуються результати поточного та підсумкового контролю. Здобувач вищої освіти, який за результатами поточного контролю набрав менше 60% від максимального балу, не допускається до семестрового контролю до моменту виконання усіх передбачених робочою програмою видів навчальної діяльності.

Здобувач, який отримав позитивний середньозважений бал (60% і більше від максимально можливого) з усіх видів поточного контролю, але не склав іспит, вважається таким, що має академічну заборгованість. Її ліквідація здійснюється під час екзаменаційної сесії або у строки, встановлені деканатом відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у Хмельницькому національному університеті».

8. Політика дисципліни

Політика навчальної дисципліни визначається вимогами чинних нормативних документів Хмельницького національного університету щодо організації освітнього процесу, навчально-методичного забезпечення та дотримання принципів академічної доброчесності.

Здобувачі вищої освіти зобов'язані проходити інструктаж з техніки безпеки перед початком виконання лабораторних робіт. Відвідування лекційних і лабораторних занять є обов'язковим. У разі наявності об'єктивних причин, підтверджених документально, навчання може здійснюватися в онлайн-режимі за погодженням з викладачем.

Успішне опанування дисципліни передбачає попередню підготовку до лабораторних занять (опрацювання теоретичного матеріалу з теми, підготовку протоколу роботи, виконання попередніх розрахунків, підготовку до усного опитування для допуску до заняття відповідно до методичних рекомендацій). На лабораторному занятті студент повинен активно працювати, виконувати експериментальні та проєктні завдання, підготувати звіт і захистити результати роботи, а також брати участь у дискусіях щодо технічних і проєктних рішень.

Термін захисту лабораторної роботи вважається своєчасним, якщо здобувач освіти захистив її на наступному занятті після виконання. Пропущене лабораторне заняття має бути відпрацьоване у встановлений викладачем термін, але не пізніше ніж за два тижні до завершення теоретичних занять у семестрі.

Засвоєння теоретичного матеріалу оцінюється за результатами поточних тестувань (у системі Moodle або в аудиторії), усного опитування перед лабораторними роботами та захисту індивідуальних завдань. Поточний контроль дозволяє відстежувати рівень розуміння матеріалу, системність мислення та здатність застосовувати знання на практиці.

Під час виконання індивідуальних завдань здобувач повинен дотримуватися політики академічної доброчесності. Забороняються будь-які форми списування, плагіату (в тому числі з використанням електронних засобів), несанкціоноване копіювання програмного коду чи проєктних рішень. У разі виявлення порушень здобувач отримує незадовільну оцінку та зобов'язаний повторно виконати роботу відповідно до вимог програми.

У межах вивчення дисципліни допускається визнання результатів неформальної освіти — онлайн-курсів, що тематично відповідають програмі дисципліни (наприклад, курси з розроблення драйверів, проєктування RTOS, Linux Kernel Development, IoT Systems Design тощо). Визнання результатів здійснюється згідно з «Положенням про порядок визнання та зарахування результатів навчання здобувачів вищої освіти у ХНУ».

9. Оцінювання результатів навчання студентів у семестрі

Оцінювання академічних досягнень здобувача вищої освіти здійснюється відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ». При поточному оцінюванні виконаної здобувачем роботи з кожної структурної одиниці і отриманих ним результатів викладач виставляє йому певну кількість балів із встановлених Робочою програмою для цього виду роботи. При цьому кожна структурна одиниця навчальної роботи може бути зарахована, якщо здобувач набрав не менше 60 відсотків (мінімальний рівень для позитивної оцінки) від максимально можливої суми балів, призначеної структурній одиниці.

При оцінюванні результатів навчання здобувачів вищої освіти з будь-якого виду навчальної роботи (структурної одиниці) рекомендується використовувати наведені нижче узагальнені критерії:

Таблиця – Критерії оцінювання навчальних досягнень здобувача вищої освіти

Оцінка та рівень досягнення здобувачем запланованих ПРН та сформованих компетентностей	Узагальнений зміст критерія оцінювання
--	--

Відмінно (високий)	Здобувач вищої освіти глибоко і у повному обсязі опанував зміст навчального матеріалу, легко в ньому орієнтується і вміло використовує понятійний апарат; уміє пов'язувати теорію з практикою, вирішувати практичні завдання, впевнено висловлювати і обґрунтовувати свої судження. Відмінна оцінка передбачає логічний виклад відповіді мовою викладання (в усній або у письмовій формі), демонструє якісне оформлення роботи і володіння спеціальними приладами та інструментами, прикладними програмами. Здобувач не вагається при видозміні запитання, вміє робити детальні та узагальнюючі висновки, демонструє практичні навички з вирішення фахових завдань. При відповіді допустив дві–три несуттєві <i>похибки</i> .
Добре (середній)	Здобувач вищої освіти виявив повне засвоєння навчального матеріалу, володіє понятійним апаратом, орієнтується у вивченому матеріалі; свідомо використовує теоретичні знання для вирішення практичних задач; виклад відповіді грамотний, але у змісті і формі відповіді можуть мати місце окремі неточності, нечіткі формулювання правил, закономірностей тощо. Відповідь здобувача вищої освіти будується на основі самостійного мислення. Здобувач вищої освіти у відповіді допустив дві–три <i>несуттєві помилки</i> .
Задовільно (достатній)	Здобувач вищої освіти виявив знання основного програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь здобувача вищої освіти будується на рівні репродуктивного мислення, здобувач вищої освіти має слабкі знання структури навчальної дисципліни, допускає неточності і <i>суттєві помилки</i> у відповіді, вагається при відповіді на видозмінене запитання. Разом з тим, набув навичок, необхідних для виконання нескладних практичних завдань, які відповідають мінімальним критеріям оцінювання і володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.
Незадовільно (недостатній)	Здобувач вищої освіти виявив 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 здобувачеві вищої освіти, який не може продовжити навчання без додаткової роботи з вивчення навчальної дисципліни.

Структурування дисципліни за видами навчальної роботи і оцінювання результатів навчання студентів денної форми здобуття освіти у семестрі

Аудиторна робота								Контрольні заходи		Семестровий контроль	
Лабораторні роботи №:								Тестовий контроль:		Іспит	Разом балів
1	2	3	4	5	6	7	8	Т 1-3	Т 4-6		
Кількість балів за вид навчальної роботи (мінімум-максимум)											60-100*
3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5	6-10	6-10	24-40	
24-40								12-20		24-40	

Примітки: *За набрану з будь-якого виду навчальної роботи з дисципліни кількість балів, нижче встановленого мінімуму, здобувач отримує незадовільну оцінку і має її перездати у встановлений викладачем (деканом) термін. Інституційна оцінка встановлюється відповідно до таблиці «Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС».

Оцінювання результатів захисту лабораторної роботи

Виконана та оформлена відповідно до встановлених Методичних рекомендацій лабораторна робота з дисципліни «Теорія і технології проектування спеціалізованих операційних систем» комплексно оцінюється викладачем під час її захисту з урахуванням таких критеріїв: самостійність та правильність виконання завдання; повнота і точність відповідей на запитання викладача; знання теоретичних основ і методики проектування компонентів спеціалізованих операційних систем; якість реалізованого програмного або технічного рішення; дотримання вимог до оформлення звіту, структурної та графічної частини роботи.

Результат виконання і захисту здобувачем вищої освіти кожної лабораторної роботи оцінюється відповідно до таблиці «Критерії оцінювання навчальних досягнень здобувача вищої освіти».

Мінімальний позитивний бал за лабораторну роботу становить 3 бали, максимальний — 5 балів.

У випадку виявлення рівня знань здобувача нижчого ніж 60% від максимального балу, встановленого Робочою програмою для кожної лабораторної роботи, вона не зараховується. Для повторного захисту здобувач має детальніше опрацювати теоретичний матеріал з теми, відновити чи скоригувати результати роботи, усунути недоліки в оформленні звіту та повторно вийти на захист у термін, визначений викладачем.

Оцінювання результатів тестового контролю

Кожний тестовий контроль, передбачений Робочою програмою навчальної дисципліни «Теорія і технології проектування спеціалізованих операційних систем», містить 25 тестових завдань, кожне з яких має однакову вагу.

Оцінювання результатів тестування здійснюється відповідно до кількості правильно наданих відповідей.

Залежно від відсотка правильних відповідей здобувач вищої освіти може отримати від 6 до 10 балів згідно з наведеною таблицею розподілу балів.

Розподіл балів в залежності від наданих правильних відповідей на тестові завдання

Кількість правильних відповідей	1-13	14-16	17-18	19-20	21-22	23-25
Відсоток правильних відповідей	0-59	60-65	66-72	73-82	83-89	90-100
Кількість балів	-	6	7	8		

На тестування відводиться 20-25 хвилин. Студент проходить тестування в он-лайн режимі у Модульному середовищі для навчання. Також, студент може проходити тестування письмово, записуючи правильні відповіді у талоні відповідей. При отриманні негативної оцінки тест слід перездати до терміну *наступного* контролю.

Оцінювання результатів підсумкового семестрового контролю (іспит)

Освітня програма передбачає підсумковий семестровий контроль з дисципліни у формі іспиту, завданням якого є системне й об'єктивне оцінювання як теоретичної, так і практичної підготовки здобувача з навчальної дисципліни. Складання іспиту відбувається за попередньо розробленими і затвердженими на засіданні кафедри білетами. Відповідно до цього в екзаменаційному білеті пропонується поєднання питань як теоретичного (в т.ч. у тестовій формі), так і практичного характеру.

Таблиця – Оцінювання результатів підсумкового семестрового контролю здобувачів денної форми навчання (40 балів для підсумкового контролю)

Види завдань	Для кожного окремого виду завдань		
	Мінімальний (достатній) бал (задовільно)	Потенційні позитивні бали* (середній бал) (добре)	Максимальний (високий) бал (відмінно)
Теоретичне питання № 1	3	4	5
Теоретичне питання № 2	3	4	5
Практичне завдання (6 задач по 3 бали)	18	24	30
Разом:	24		40

Примітка. *Позитивний бал за іспит, відмінний від мінімального (24 бали) та максимального (40 балів), знаходиться в межах 25-39 балів та розраховується як сума балів за усі структурні елементи (завдання) іспиту.

Для кожного окремого виду завдань підсумкового семестрового контролю застосовуються критерії оцінювання навчальних досягнень здобувача вищої освіти, наведені вище (Таблиця – Критерії оцінювання навчальних досягнень здобувача вищої освіти).

Підсумкова семестрова оцінка за інституційною шкалою і шкалою ЄКТС визначається в автоматизованому режимі після внесення викладачем результатів оцінювання у балах з усіх видів навчальної роботи до електронного журналу. Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС наведені нижче у таблиці «Співвідношення».

Семестровий іспит виставляється, якщо загальна сума балів, яку набрав студент з дисципліни за результатами поточного контролю, знаходиться у межах від 60 до 100 балів. При цьому за інституційною шкалою ставиться оцінка «відмінно/добре/задовільно», а за шкалою ЄКТС – буквене позначення оцінки, що відповідає набраній студентом сумі балів відповідно до таблиці Співвідношення.

Таблиця – Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ЄКТС	Рейтингова шкала балів	Інституційна оцінка (рівень досягнення здобувачем вищої освіти запланованих результатів навчання з навчальної дисципліни)	
		Залік	Іспит/диференційований залік
A	90-100	Зараховано	<i>Відмінно/Excellent</i> – високий рівень досягнення запланованих результатів навчання з навчальної дисципліни, що свідчить про безумовну готовність здобувача до подальшого навчання та/або професійної діяльності за фахом
B	83-89		<i>Добре/Good</i> – середній (максимально достатній) рівень досягнення запланованих результатів навчання з навчальної дисципліни та готовності до подальшого навчання та/або професійної діяльності за фахом
C	73-82		<i>Задовільно/Satisfactory</i> – Наявні мінімально достатні для подальшого навчання та/або професійної діяльності за фахом результати навчання з навчальної дисципліни
D	66-72		
E	60-65		
FX	40-59	Незараховано	<i>Незадовільно/Fail</i> – Низка запланованих результатів навчання з навчальної дисципліни відсутня. Рівень набутих результатів навчання є недостатнім для подальшого навчання та/або професійної діяльності за фахом
F	0-39		<i>Незадовільно/Fail</i> – Результати навчання відсутні

10. Питання для самоконтролю результатів навчання

1. Теоретичні основи та класифікація спеціалізованих операційних систем

- 1.1. Поняття та визначення спеціалізованої операційної системи.
- 1.2. Класифікація сучасних спеціалізованих ОС.
- 1.3. Місце спеціалізованих ОС у загальній класифікації операційних систем.
- 1.4. Основні принципи побудови спеціалізованих ОС.
- 1.5. Теоретичні основи функціонування операційних систем.
- 1.6. Архітектурні особливості спеціалізованих обчислювальних систем.
- 1.7. Типові складові спеціалізованої операційної системи.
- 1.8. Основні функції ядра спеціалізованої ОС.
- 1.9. Задачі диспетчера процесів у спеціалізованих ОС.
- 1.10. Засоби керування пам'яттю у спеціалізованих ОС.
- 1.11. Методи керування пристроями введення-виведення.
- 1.12. Основні підходи до забезпечення надійності та безпеки в спеціалізованих ОС.
- 1.13. Апаратно-програмні засоби в задачах проєктування спеціалізованих ОС.
- 1.14. Сфери застосування спеціалізованих операційних систем.
- 1.15. Проблематика проєктування спеціалізованих ОС у сучасних обчислювальних середовищах.
- 1.16. Підходи до підвищення надійності та стійкості спеціалізованих ОС.
- 1.17. Поняття ресурсного управління у спеціалізованих ОС.
- 1.18. Приклади сучасних спеціалізованих операційних систем.
- 1.19. Основні наукові напрями розвитку спеціалізованих ОС.
- 1.20. Проблеми масштабування та інтеграції спеціалізованих ОС у гетерогенних середовищах.

2. Моделі та технології систем реального часу

- 2.1. Поняття систем реального часу.
- 2.2. Класифікація систем реального часу.
- 2.3. Моделі систем реального часу.
- 2.4. Методи керування процесами у системах реального часу.
- 2.5. Апаратно-програмна підтримка систем реального часу.
- 2.6. Типи планувальників у системах реального часу.
- 2.7. Переваги й недоліки систем із фіксованими пріоритетами.
- 2.8. Моделі обробки подій у системах реального часу.
- 2.9. Забезпечення детермінованості виконання процесів.
- 2.10. Методи тестування систем реального часу.

3. Інтернет речей, мобільні та кіберфізичні системи

- 3.1. Поняття операційної системи Інтернету речей (IoT OS).
- 3.2. Архітектура IoT-пристроїв.
- 3.3. Моделі систем Інтернету речей.
- 3.4. Методи проєктування спеціалізованих ОС для IoT.

- 3.5. Проблема гетерогенності в IoT-системах.
- 3.6. Керування ресурсами в операційних системах Інтернету речей.
- 3.7. Особливості енергоспоживання в IoT-ОС.
- 3.8. Моделі кіберфізичних систем.
- 3.9. Методи проектування ОС для кіберфізичних систем.
- 3.10. Моделі мобільних систем.
- 3.11. Методи проектування ОС для мобільних пристроїв.
- 3.12. Безпека в мобільних ОС.
- 3.13. Віртуалізація в мобільних і IoT-системах.
- 3.14. Проблеми сумісності IoT-ОС.
- 3.15. Приклади спеціалізованих ОС для IoT: Zephyr, RIOT, FreeRTOS.
- 4. Робототехнічні системи та штучний інтелект**
 - 4.1. Моделі робототехнічних систем.
 - 4.2. Особливості архітектури ROS (Robot Operating System).
 - 4.3. Технології штучного інтелекту в робототехнічних системах.
 - 4.4. Методи машинного навчання у робототехнічних ОС.
 - 4.5. Інтеграція сенсорів у робототехнічних ОС.
 - 4.6. Планування руху в робототехнічних системах.
 - 4.7. Стратегії розвитку робототехнічних ОС.
 - 4.8. Переваги ROS 2 порівняно з ROS 1.
 - 4.9. Засоби симуляції роботів у ROS / Gazebo.
 - 4.10. Використання ROS у кіберфізичних системах.
- 5. Проектування драйверів пристроїв**
 - 5.1. Поняття драйвера пристрою.
 - 5.2. Типи драйверів.
 - 5.3. Драйвери режиму ядра та режиму користувача.
 - 5.4. Віртуальна модель пристрою та його інтерфейсу.
 - 5.5. Human Interface Devices (HID).
 - 5.6. Network Driver Interface Specification (NDIS).
 - 5.7. Драйвери протоколів.
 - 5.8. Потік даних у KMDF / UMDF.
 - 5.9. Керування живленням і модель введення / виведення у WDF.
 - 5.10. Setup Application Programming Interface (SetupAPI).
 - 5.11. Особливості моделі введення / виведення UMDF / KMDF.
 - 5.12. Об'єктна модель WDF.
 - 5.13. Об'єктна модель KMDF.
 - 5.14. Життєвий цикл об'єкта KMDF.
 - 5.15. Об'єкти UMDF та життєвий цикл UMDF-драйвера.
 - 5.16. Windows Driver Foundation (WDF).

- 5.17. Структура драйвера.
- 5.18. Основні функції для побудови драйвера.
- 5.19. Вибір моделі драйвера для ОС Windows.
- 5.20. Технології проектування драйверів для Linux.
- 5.21. Компіляція драйверів разом із ядром.
- 5.22. Реалізація драйверів як модулів ядра.
- 5.23. Відмінності між статичним і динамічним підключенням драйверів.
- 5.24. Особливості налагодження драйверів у середовищі Windows і Linux.
- 5.25. Приклади відкритих драйверів для IoT-систем.

6. Розподілені та паралельні системи

- 6.1. Принципи створення розподілених систем.
- 6.2. Теорія розподілених обчислень.
- 6.3. Проблемні наукові задачі в розподілених системах.
- 6.4. Проміжне програмне забезпечення для розподілених систем.
- 6.5. Архітектура клієнт-сервер у розподілених ОС.
- 6.6. Принципи побудови блокчейн-мереж.
- 6.7. Життєвий цикл транзакції в блокчейні.
- 6.8. Моделі надійності розподілених реєстрів.
- 6.9. Паралельні обчислення та їх особливості.
- 6.10. Проблемні задачі реалізації паралельних обчислень.
- 6.11. Балансування навантаження серверів Linux.
- 6.12. Технологія HAProxy.
- 6.13. Технологія Keepalived.
- 6.14. Nginx як інструмент балансування навантаження.
- 6.15. Підходи до відмовостійкості в розподілених ОС.
- 6.16. Моніторинг продуктивності у паралельних системах.
- 6.17. Використання контейнеризації у розподілених ОС.
- 6.18. Засоби автоматизації розгортання розподілених систем (Ansible, Kubernetes).
- 6.19. Підходи до оптимізації обчислень у гібридних середовищах.
- 6.20. Перспективи розвитку спеціалізованих ОС для розподілених обчислень.

11. Навчально-методичне забезпечення

Освітній процес з дисципліни «Теорія і технології проектування спеціалізованих операційних систем» повністю і в достатній кількості забезпечений необхідними навчально-методичними матеріалами, що охоплюють лекційний курс, лабораторні заняття, самостійну роботу та підсумковий контроль.

12. Матеріально-технічне та програмне забезпечення дисципліни (за потреби)

Інформаційна та комп'ютерна підтримка: ПК, планшет, смартфон або інший мобільний пристрій, проектор. Програмне забезпечення: програми Microsoft Office або аналогічні, доступ до мережі Інтернет, робота з презентаціями. Вивчення навчальної дисципліни не потребує використання спеціального програмного прикладного забезпечення, крім загальноновживаних програм і операційних систем.

Бобровнікова К.Ю. Теорія і технології проектування 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.

13. Рекомендована література:

Основна

1. Galvin, P. B., et al. Operating System Concepts, 11th Edition (Wiley, 2024) — сучасне видання класичного підручника.
2. Tanenbaum, A. S., & Bos, H. Modern Operating Systems, 5th Edition (Pearson, 2021) — останнє видання.
3. Stallings, W. Operating Systems: Internals and Design Principles, 10th Edition (2022) — акцент на внутрішню архітектуру.
4. Liu, C., & Yuan, W. Real-Time Systems: Scheduling, Analysis, and Verification, 2nd Ed. (Springer, 2022) — для систем реального часу.
5. Zhang, Y., & Zhang, X. Embedded and Real-Time Operating Systems for ARM Cortex-M Microcontrollers, 2023 (Springer) — спеціалізований на вбудованих ОС.

Додаткова

6. Erciyes, K. Distributed Real-Time Systems: Theory and Practice. Springer, Cham, 2019. 287 p.
7. Wang, K. C. Embedded and Real-Time Operating Systems. Springer, Cham, 2017. 439 p.
8. Liu, C., & Yuan, W. Real-Time Systems: Scheduling, Analysis, and Verification. 2nd ed. Springer, 2022. 312 p.
9. Agarwal, V. K., Kumar, N., Singh, R., & 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, pp. 91–105.
10. 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.
11. Serino, A., & Cheng, L. Real-Time Operating Systems for Cyber-Physical Systems: Current Status and Future Research. IEEE Cybermatics Conference Proceedings, 2020, pp. 419–425.
12. Dawarka, V., & Bekaroo, G. Building and Evaluating Cloud Robotic Systems: A Systematic Review. Robotics and Computer-Integrated Manufacturing, 2022, 73, 102240.
13. 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.
14. Lu, Y. The Blockchain: State-of-the-Art and Research Challenges. Journal of Industrial Information Integration, 15, 2019, pp. 80–90.
15. Trobec, R., Slivnik, B., Bulic, P., & Robic, B. Introduction to Parallel Computing: From Algorithms to Programming on State-of-Art Platforms. Springer, Cham, 2018.
16. Shukur, H., Zeebaree, S. R., Ahmed, A. J., & Tahir, B. S. 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.
17. Balador, A., Ericsson, N., & Bakhshi, Z. Communication Middleware Technologies for Industrial Distributed Control Systems: A Literature Review. IEEE ETFA Conference Proceedings, 2017, pp. 1–6.
18. Marufuzzaman, M., Rahman, M. S., & Zahid, N. M. A Review on Reliability, Security and Memory Management of Numerous Operating Systems. Indonesian Journal of Electrical Engineering and Informatics, 7(3), 2019, pp. 577–585.
19. Javed, F., Afzal, M. K., & Kim, B. S. IoT Operating Systems Support, Networking Technologies, Applications, and Challenges: A Comparative Review. IEEE Communications Surveys & Tutorials, 20(3), 2018, pp. 2062–2100.
20. Bakhshi, Z., & Balador, A. Industrial IoT Middleware: Requirements, Challenges, and Future Directions. IEEE Access, 2020, 8, pp. 198895–198920.
- 21.
22. Rafiuzzaman, M., Gascon-Samson, J., & Pattabiraman, K. ThingsJS: Towards a Flexible and Self-Adaptable Middleware for Dynamic and Heterogeneous IoT Environments. Proceedings of the 4th Workshop on Middleware for IoT, ACM, 2019, pp. 11–16.
23. Love, R. Linux Kernel Development: A Comprehensive Guide to the Kernel Internals. Addison-Wesley, 2021. 480 p.
24. Simmonds, C. Mastering Embedded Linux Programming. 3rd ed. Packt Publishing, 2021. 546 p.
25. Cooling, J. Real-Time Operating Systems Book 1: The Foundations. Independently Published, 2020. 324 p.
26. Zeebaree, S. R., et al. Design and Implementation of a Parallel and Distributed System for IoT Data Processing. International Journal of Interactive Mobile Technologies (iJIM), 2021, 15(10), pp. 70–85.

14. Інформаційні ресурси

1. Модульне середовище для навчання. URL : <https://msn.khmnu.edu.ua/course/view.php?id=3663>
2. Електронна бібліотека ХНУ. URL: <http://library.khmnu.edu.ua/>
3. Інституційний репозитарій ХНУ. URL : <https://elar.khmnu.edu.ua/home>

ТЕОРІЯ І ТЕХНОЛОГІЇ ПРОЄКТУВАННЯ СПЕЦІАЛІЗОВАНИХ ОПЕРАЦІЙНИХ СИСТЕМ

Тип дисципліни	Обов'язкова
Рівень вищої освіти	Другий (магістерський)
Мова викладання	Українська
Кількість призначених кредитів ЄКТС	5,0
Форми здобуття освіти, для яких викладається дисципліна	Очна (денна)

Результати навчання. Після вивчення навчальної дисципліни «Теорія і технології проєктування спеціалізованих операційних систем» здобувач вищої освіти повинен володіти професійною термінологією та базовими концепціями теорії й технологій проєктування спеціалізованих операційних систем; знати архітектурні особливості, принципи побудови, функціонування та взаємодії компонентів операційних систем реального часу, кіберфізичних, мобільних, робототехнічних, розподілених та вбудованих систем; уміти аналізувати, проєктувати та розробляти компоненти операційних систем і драйвери пристроїв, застосовувати сучасні методи і технології створення спеціалізованих операційних систем, розробляти проміжне програмне забезпечення для розподілених і паралельних обчислень, забезпечуючи їх ефективність, відмовостійкість і масштабованість; виконувати наукові та експериментальні дослідження, моделювати процеси функціонування спеціалізованих операційних систем, оцінювати їхню продуктивність, надійність і безпеку; використовувати технології блокчейн, контейнеризації, віртуалізації та балансування навантаження у розподілених обчислювальних середовищах; дотримуватися принципів академічної доброчесності та стандартів безпеки програмного забезпечення під час реалізації індивідуальних і колективних проєктів

Зміст навчальної дисципліни. Архітектурні особливості та класифікація сучасних спеціалізованих операційних систем; принципи побудови, функціонування та взаємодії компонентів операційних систем для систем реального часу, кіберфізичних, мобільних, робототехнічних, розподілених і вбудованих систем; теорія та технології проєктування драйверів пристроїв у середовищах Windows і Linux; проміжне програмне забезпечення (middleware) для розподілених і паралельних обчислень; принципи побудови та функціонування блокчейн-технологій; методи забезпечення надійності, безпеки, балансування навантаження та відмовостійкості у спеціалізованих операційних системах; сучасні інструменти та середовища розроблення, моделювання і тестування спеціалізованих ОС.

Пререквізити: Іноземна мова за професійним спрямуванням (ОЗП.01); Філософські проблеми наукового пізнання (ОЗП.02); Методологія та організація наукових досліджень (ОЗП.04); Комп'ютерні системи штучного інтелекту (ОФП.01); Технології проєктування програмних систем (ОФП.02); Технології проєктування програмних систем (курсовий проєкт) (ОФП.03); Теорія і проєктування комп'ютерних та кіберфізичних систем і мереж (ОФП.04).

Кореквізити: Методологія забезпечення безпеки та захисту комп'ютерних систем (ОФП.06); Теорія, проєктування та моделювання спеціалізованих комп'ютерних систем (ОФП.08); Науково-дослідна практика (ОФП.09); Кваліфікаційна робота (ОФП.10).

Запланована навчальна діяльність: Мінімальний обсяг навчальних занять в одному кредиті ЄКТС навчальної дисципліни для другого (магістерського) рівня вищої освіти за денною формою здобуття освіти становить 8 годин на 1 кредит ЄКТС. Загальний обсяг дисципліни — 5 кредитів ЄКТС (150 годин), із них 50 годин аудиторних занять (16 годин лекцій, 34 години лабораторних робіт) та 100 годин самостійної роботи здобувача освіти.

Форми (методи) навчання: лекції (з використанням мультимедійних презентацій, методів візуалізації, проблемного й інтерактивного навчання, інформаційно-комунікаційних технологій, елементів мотивації та стимулювання пізнавальної активності здобувачів освіти);

лабораторні заняття (з використанням інструктування, демонстрування, аналізу кейсів, розв'язування ситуаційних і практичних завдань, елементів дискусії, застосування комп'ютерного моделювання, проєктної діяльності та сучасних середовищ розроблення операційних систем);

самостійна робота (опрацювання теоретичного матеріалу, підготовка до виконання лабораторних робіт, поточного та підсумкового контролю, виконання індивідуальних і дослідницьких завдань із використанням ресурсів дистанційного навчання, зокрема середовища Moodle).

Форми оцінювання результатів навчання: оцінювання результатів захисту лабораторних робіт; тестовий контроль засвоєння теоретичного матеріалу з окремих тем або розділів; оцінювання результатів виконання індивідуальних завдань; аналіз та оцінювання участі здобувача в обговоренні проблемних питань під час занять; підсумковий контроль у формі іспиту.

Вид семестрового контролю: іспит

Навчальні ресурси:

1. Galvin, P. B., et al. Operating System Concepts, 11th Edition (Wiley, 2024) — сучасне видання класичного підручника.
2. Tanenbaum, A. S., & Bos, H. Modern Operating Systems, 5th Edition (Pearson, 2021) — останні видання.
3. Stallings, W. Operating Systems: Internals and Design Principles, 10th Edition (2022) — акцент на внутрішню архітектуру.
4. Liu, C., & Yuan, W. Real-Time Systems: Scheduling, Analysis, and Verification, 2nd Ed. (Springer, 2022) — для систем реального часу.
5. Zhang, Y., & Zhang, X. Embedded and Real-Time Operating Systems for ARM Cortex-M Microcontrollers, 2023 (Springer) — спеціалізований на вбудованих ОС.
6. Erciyes, K. Distributed Real-Time Systems: Theory and Practice. Springer, Cham, 2019. 287 p.
7. Wang, K. C. Embedded and Real-Time Operating Systems. Springer, Cham, 2017. 439 p.
8. Liu, C., & Yuan, W. Real-Time Systems: Scheduling, Analysis, and Verification. 2nd ed. Springer, 2022. 312 p.
9. Agarwal, V. K., Kumar, N., Singh, R., & 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, pp. 91–105.
10. 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.
11. Serino, A., & Cheng, L. Real-Time Operating Systems for Cyber-Physical Systems: Current Status and Future Research. IEEE Cybermatics Conference Proceedings, 2020, pp. 419–425.
12. Dawarka, V., & Bekaroo, G. Building and Evaluating Cloud Robotic Systems: A Systematic Review. Robotics and Computer-Integrated Manufacturing, 2022, 73, 102240.
13. 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.
14. Lu, Y. The Blockchain: State-of-the-Art and Research Challenges. Journal of Industrial Information Integration, 15, 2019, pp. 80–90.
15. Trobec, R., Slivnik, B., Bulic, P., & Robic, B. Introduction to Parallel Computing: From Algorithms to Programming on State-of-Art Platforms. Springer, Cham, 2018.
16. Shukur, H., Zeebaree, S. R., Ahmed, A. J., & Tahir, B. S. 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.
17. Balador, A., Ericsson, N., & Bakhshi, Z. Communication Middleware Technologies for Industrial Distributed Control Systems: A Literature Review. IEEE ETFA Conference Proceedings, 2017, pp. 1–6.
18. Marufuzzaman, M., Rahman, M. S., & Zahid, N. M. A Review on Reliability, Security and Memory Management of Numerous Operating Systems. Indonesian Journal of Electrical Engineering and Informatics, 7(3), 2019, pp. 577–585.
19. Javed, F., Afzal, M. K., & Kim, B. S. IoT Operating Systems Support, Networking Technologies, Applications, and Challenges: A Comparative Review. IEEE Communications Surveys & Tutorials, 20(3), 2018, pp. 2062–2100.
20. Bakhshi, Z., & Balador, A. Industrial IoT Middleware: Requirements, Challenges, and Future Directions. IEEE Access, 2020, 8, pp. 198895–198920.
- 21.
22. Rafiuzzaman, M., Gascon-Samson, J., & Pattabiraman, K. ThingsJS: Towards a Flexible and Self-Adaptable Middleware for Dynamic and Heterogeneous IoT Environments. Proceedings of the 4th Workshop on Middleware for IoT, ACM, 2019, pp. 11–16.
23. Love, R. Linux Kernel Development: A Comprehensive Guide to the Kernel Internals. Addison-Wesley, 2021. 480 p.
24. Simmonds, C. Mastering Embedded Linux Programming. 3rd ed. Packt Publishing, 2021. 546 p.
25. Cooling, J. Real-Time Operating Systems Book 1: The Foundations. Independently Published, 2020. 324 p.
26. Zeebaree, S. R., et al. Design and Implementation of a Parallel and Distributed System for IoT Data Processing. International Journal of Interactive Mobile Technologies (iJIM), 2021, 15(10), pp. 70–85.
27. Модульне середовище для навчання. URL : <https://msn.khmnmu.edu.ua/course/view.php?id=3663>
28. Електронна бібліотека ХНУ. URL: <http://library.khmnmu.edu.ua/>
29. Інституційний репозитарій ХНУ. URL : <https://elar.khmnmu.edu.ua/home>

Викладачі: старший викладач Денисюк Д.О.