

ЗАТВЕРДЖУЮ



Декан факультету інформаційних технологій

Тетяна ГОВОРУЩЕНКО
Ім'я, ПРІЗВИЩЕ

09 2025 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Системне та мережеве програмне забезпечення

Назва дисципліни

Галузь знань – 12 Інформаційні технології
Спеціальність – 122 Комп'ютерні науки
Рівень вищої освіти – Перший (бакалаврський)
Освітньо-професійна програма – Комп'ютерні науки
Обсяг дисципліни – 5 кредитів ЄКТС
Шифр дисципліни – ОПІ.06
Мова навчання – Українська
Статус дисципліни: Обов'язкова (професійної підготовки)
Факультет – Інформаційних технологій
Кафедра – Комп'ютерних наук

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	3	5	5	150	66	32	34			84				+
Разом ДФН			5	150	66	32	34			84				5

Робоча програма складена на основі освітньо-професійної програми «Комп'ютерні науки» за спеціальністю 122 «Комп'ютерні науки»

Робоча програма складена

Підпис автора

канд. техн. наук, доц. Руслан БАГРІЙ

Науковий ступінь, вчене звання, Ім'я, ПРІЗВИЩЕ автора

канд. техн. наук, доц. Олександр ПАСІЧНИК

Науковий ступінь, вчене звання, Ім'я, ПРІЗВИЩЕ автора

Схвалена на засіданні кафедри Комп'ютерних наук

Протокол від 29.08.2025 № 1.

Зав. кафедри

Підпис

Олександр БАРМАК

Ім'я, ПРІЗВИЩЕ

Робоча програма розглянута та схвалена вченою радою факультету інформаційних технологій

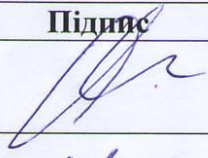
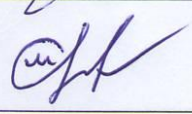
Голова вченої ради факультету

Підпис

Тетяна ГОВОРУЩЕНКО

Ім'я, ПРІЗВИЩЕ

ЛИСТ ПОГОДЖЕННЯ

Посада	Назва кафедри	Підпис	Ініціали, ПРИЗВИЩЕ
Завідувач кафедри, д-р. техн. наук, проф.	Комп'ютерних наук		Олександр БАРМАК
Гарант освітньо-професійної програми, канд. техн. наук, доц.	Комп'ютерних наук		Олександр МАЗУРЕЦЬ

3. Пояснювальна записка

Дисципліна «Системне та мережеве програмне забезпечення» є однією із дисциплін професійної підготовки і займає провідне місце у підготовці здобувачів першого (бакалаврського) рівня вищої освіти, очної (денної) (далі – денної) форми здобуття вищої освіти, які навчаються за освітньо-професійною програмою «Комп'ютерні науки» в межах спеціальності 122 «Комп'ютерні науки».

Пререквізити – Алгоритмізація та програмування (ОЗП.05), Дослідження операцій та основи теорії прийняття рішень (ОЗП.09), Основи програмної інженерії та тестування програмного забезпечення (ОПП.10), Технології захисту інформації та кібербезпека (ОПП.05).

Кореквізити – Технології розподілених систем та паралельних обчислень (ОПП.11), Проєктно-технологічна практика (ОПП.14), Професійна практика (ОПП.15), Кваліфікаційна робота (ОПП.16).

Відповідно до освітньої програми дисципліна сприяє забезпеченню:

Компетентності: здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення (ФК 12); здатність до розробки мережевого програмного забезпечення, що функціонує на основі різних топологій структурованих кабельних систем, використовує комп'ютерні системи і мережі передачі даних та аналізує якість роботи комп'ютерних мереж (ФК 13).

Програмні результати навчання: застосовувати мови системного програмування та методи розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення (ПРН 13); розуміти концепцію інформаційної безпеки, принципи безпечного проєктування програмного забезпечення, забезпечувати безпеку комп'ютерних мереж в умовах неповноти та невизначеності вихідних даних (ПРН 15).

Мета дисципліни. Формування у здобувачів вищої освіти компетентностей, необхідних для розробки та підтримки системного і мережевого програмного забезпечення, що включає знання принципів функціонування операційних систем, методів організації обчислювальних процесів, технологій мережевої взаємодії, віртуалізації та контейнеризації.

Предмет дисципліни. Методи, моделі та технології створення системного і мережевого програмного забезпечення; архітектурні підходи та інструментальні засоби, що застосовуються для проєктування, розробки, тестування та розгортання програмних систем у сучасних обчислювальних середовищах.

Завдання дисципліни. Формування у студентів практичних навичок роботи з процесами, потоками, пам'яттю, файловими системами та механізмами введення-виведення; розробки клієнт-серверних мережевих застосунків і сервісів з використанням відповідних протоколів; застосування технологій віртуалізації та контейнеризації для пакування, розгортання та безпечної ізоляції програмних рішень у сучасних обчислювальних середовищах.

Результати навчання. Після вивчення дисципліни студент повинен: володіти понятійним апаратом у сфері системного та мережевого програмного забезпечення, зокрема фундаментальними принципами функціонування операційних систем; уміти розробляти програмні рішення, що взаємодіють з ключовими компонентами ОС, керувати життєвим циклом процесів і потоків та синхронізовувати їх доступ до спільних ресурсів для забезпечення цілісності даних; проєктувати та реалізовувати ефективні багатопотокові клієнт-серверні застосунки, використовуючи API сокетів та протоколи прикладного рівня; застосовувати технології віртуалізації та контейнеризації для пакування, розгортання та безпечної ізоляції програмного забезпечення, виконуючи розробку системних утиліт і мережевих сервісів для сучасних обчислювальних середовищ.

4. Структура залікових кредитів дисципліни

Назва розділу (теми)	Кількість годин, відведених на:		
	лекції	лабораторні заняття	СРС
Тема 1. Основи операційних систем та системного програмування	16	16	34
Тема 2. Розробка мережевого програмного забезпечення	8	8	20
Тема 3. Технології віртуалізації та контейнеризації	8	10	30
Разом:	32	34	84

5. Програма навчальної дисципліни

5.1 Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їх анотації	Кількість годин
	<i>Тема 1. Основи операційних систем та системного програмування</i>	16
1	Вступ до системного та мережевого ПЗ. Поняття системного ПЗ, його критерії та приклади. Огляд системних мов програмування з акцентом на Go. Визначення та ключові функції операційної системи як розширеної машини та менеджера ресурсів. Літ.: [1, 2, 3]	4
2	Процеси та потоки. Модель процесу, його складові та стани. Планування процесів та перемикання контексту. Концепція потоків як полегшених процесів, їх переваги, моделі та варіанти реалізації (в просторі користувача та в ядрі). Літ.: [1, 2, 3]	2
3	Синхронізація процесів та потоків. Міжпроцесна комунікація. Проблема ситуації змагання та поняття критичної області. Механізми синхронізації: від активного очікування до семафорів, м'ютексів, умовних змінних та високорівневих моніторів. Літ.: [1, 2, 3]	2
4	Керування пам'яттю. Роль диспетчера пам'яті. Абстракція адресного простору. Концепція віртуальної пам'яті: сторінкова та сегментна організація. Механізми трансляції адрес (таблиці сторінок, TLB) та алгоритми заміщення сторінок. Літ.: [1, 2]	2
5	Файлові системи. Концепція файлів, атрибутів та каталогів. Організація та структура файлових систем. Методи розміщення файлів на диску (неперервне, зв'язане, індексне) та керування вільним простором. Літ.: [1, 2]	2
6	Підсистема вводу-виводу. Принципи роботи апаратного та програмного забезпечення вводу-виводу. Роль драйверів пристроїв, обробники переривань. Загальна структура підсистеми I/O. Літ.: [1, 2]	2
7	Проблема тупиків та практичні методи їх уникнення. Аналіз проблеми взаємних блокувань (тупиків) та умов їх виникнення. Практичні підходи до запобігання тупикам у коді: дотримання порядку блокувань, використання тайм-аутів.	2

	Літ.: [1, 2]	
	<i>Тема 2. Розробка мережевого програмного забезпечення</i>	8
8	Основи комп'ютерних мереж. Моделі взаємодії відкритих систем OSI та TCP/IP, їх рівні та протоколи. Поняття інкапсуляції даних. IP-адресація (IPv4, IPv6) та порти. Характеристики та відмінності протоколів транспортного рівня TCP та UDP. Літ.: [1, 2, 4]	2
9	Програмування сокетів. Концепція та архітектура API сокетів Берклі. Розробка клієнт-серверних застосунків. Робота з потоковими сокетами (TCP) для надійного з'єднання та датаграмними сокетами (UDP) для швидкої передачі даних. Приклади на Go. Літ.: [1, 3, 4]	2
10	Розробка мережевих застосунків та протоколи прикладного рівня. Огляд поширених протоколів прикладного рівня з фокусом на HTTP. Структура HTTP-запитів та відповідей. Розробка простого веб-сервера та клієнта на Go. Формати передачі даних, зокрема JSON. Літ.: [1, 3, 4]	2
11	Паралелізм у мережевому програмуванні. Проблема обробки кількох клієнтів одночасно. Класичні моделі паралелізму (один процес/потік на з'єднання). Ефективне використання горутин та каналів в Go для побудови асинхронних, високопродуктивних мережевих сервісів. Літ.: [1, 3, 4]	2
	<i>Тема 3. Технології віртуалізації та контейнеризації</i>	8
12	Віртуалізація: від гіпервізорів до хмарних інфраструктур. Концепція віртуальної машини. Гіпервізори 1-го (bare-metal) та 2-го (hosted) типів. Технічні аспекти віртуалізації: апаратна підтримка (Intel VT-x, AMD-V), віртуалізація пам'яті (Shadow Page Tables, EPT/NPT) та вводу-виводу (емуляція, пара-віртуалізація). Роль віртуалізації як основи для хмарних сервісів IaaS. Літ.: [2, 6]	4
13	Контейнеризація та оркестрація. Порівняння контейнерів та віртуальних машин. Технології ядра ОС для контейнеризації: простори імен (namespaces) для ізоляції та контрольні групи (cgroups) для обмеження ресурсів. Екосистема Docker: образи, Dockerfile, контейнери. Вступ до оркестрації: проблеми керування контейнерами та роль Kubernetes. Літ.: [5, 6]	4
	Разом:	32

5.3 Зміст лабораторних занять

№ п/п	Тема лабораторного заняття	Кількість годин
	<i>Тема 1. Основи операційних систем та системного програмування</i>	16
1	Лабораторна робота 1. Огляд мови Go (Golang) Літ.: [1, 3]	4
2	Потоки даних. Обробка потоків даних. Вбудовані утиліти Літ.: [1, 3]	4
3	Обробка процесів та демонів (служб) Літ.: [1, 3]	4
4	Операції з файлами Літ.: [1, 3]	4
	<i>Тема 2. Розробка мережевого програмного забезпечення</i>	8

5	Основи роботи з сокетми Літ.: [1, 4]	4
6	Розробка багатопотокового HTTP-сервера Літ.: [1, 4]	4
	<i>Тема 3. Технології віртуалізації та контейнеризації</i>	10
7	Робота з Docker. Контейнеризація Go-застосунку Літ.: [3, 5]	4
8	Розгортання мікросервісу з Docker Compose Літ.: [5, 6]	6
	Разом:	34

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

Самостійна робота студентів усіх форм здобуття освіти полягає у систематичному опрацюванні програмного матеріалу з відповідних джерел інформації, підготовці до виконання і захисту лабораторних робіт. Крім цього, до послуг студентів сторінка навчальної дисципліни у Модульному середовищі для навчання, де розміщені Робоча програма дисципліни та необхідні документи з її навчально-методичного забезпечення.

Номер тижня	Вид самостійної роботи <i>П'ятий семестр</i>	Кількість годин
1-2	Опрацювання теоретичного матеріалу, підготовка до виконання лабораторної роботи №1.	8
3-4	Опрацювання теоретичного матеріалу, підготовка до захисту лабораторної роботи №1 та виконання лабораторної роботи №2.	8
5-6	Опрацювання теоретичного матеріалу, підготовка до захисту лабораторної роботи №2 та виконання лабораторної роботи №3.	8
7-8	Опрацювання теоретичного матеріалу, підготовка до захисту лабораторної роботи №3 та виконання лабораторної роботи №4. Початок виконання індивідуального завдання №1 самостійної роботи.	10
9-10	Опрацювання теоретичного матеріалу, підготовка до захисту лабораторної роботи №4 та виконання лабораторної роботи №5. Завершення виконання індивідуального завдання №1 самостійної роботи.	10
11-12	Опрацювання теоретичного матеріалу, підготовка до захисту лабораторної роботи №5 та виконання лабораторної роботи №6. Початок виконання індивідуального завдання №2 самостійної роботи.	10
13-14	Опрацювання теоретичного матеріалу, підготовка до захисту лабораторної роботи №6 та виконання лабораторної роботи №7.	10
15-16	Опрацювання теоретичного матеріалу, підготовка до захисту лабораторної роботи №7 та виконання лабораторної роботи №8. Завершення виконання індивідуального завдання №2 самостійної роботи.	10
17	Опрацювання теоретичного матеріалу, підготовка до захисту лабораторних робіт. Оформлення звіту з самостійної роботи. Презентація результатів виконання індивідуального завдання.	10
	Разом:	84

На самостійне опрацювання студентів виносяться визначені у методичних рекомендаціях до лабораторних занять та самостійної роботи індивідуальні домашні завдання (ІДЗ) з відповідних тем. Керівництво самостійною роботою та контроль за виконанням індивідуального завдання здійснюється викладачем згідно з розкладом консультацій у позаурочний час.

Вимоги до виконання індивідуального домашнього завдання викладені в Модульному середовищі для навчання на сторінці навчальної дисципліни.

6. Технології та методи навчання

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних технологій та методів навчання, зокрема: лекції (з використанням мультимедійних презентацій, методів візуалізації, пояснення, проблемного й інтерактивного навчання, методів стимулювання і мотивації, інформаційно-комунікаційних технологій, інтенсифікації та індивідуалізації навчання); лабораторні заняття (з використанням методів комп'ютерного моделювання, методів проєктної діяльності, аналізу проблемних ситуацій, пояснення, дискусії); самостійна робота (робота над засвоєнням теоретичного матеріалу, виконання індивідуальних завдань, підготовка до виконання та захисту лабораторних робіт, опрацювання навчально-методичних матеріалів, розміщених у модульному середовищі навчання), з використанням інформаційно-комп'ютерних технологій та технологій дистанційного навчання.

7. Методи контролю

Поточний контроль здійснюється під час аудиторних лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочою програмою і графіком освітнього процесу, в т.ч. з використанням Модульного середовища для навчання. При цьому використовуються такі методи поточного контролю:

- усне опитування перед допуском до лабораторного заняття;
- оцінювання результатів захисту лабораторних робіт;
- презентація і захист індивідуальних завдань.

При виведенні підсумкової семестрової оцінки враховуються результати як поточного контролю, так і підсумкового контролю, який проводиться з усього матеріалу дисципліни за білетами, попередньо розробленими і затвердженими на засіданні кафедри. Здобувач вищої освіти, який набрав з будь-якого виду навчальної роботи, суму балів нижчу за 60 відсотків від максимального балу, **не допускається** до семестрового контролю, поки не виконає обсяг роботи, передбачений Робочою програмою. Здобувач вищої освіти, який набрав позитивний середньозважений бал (60 відсотків і більше від максимального балу) з усіх видів поточного контролю і не склав іспит, вважається таким, який **має** академічну заборгованість. Ліквідація академічної заборгованості із семестрового контролю здійснюється у період екзаменаційної сесії або за графіком, встановленим деканатом відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ».

8. Політика дисципліни

Політика навчальної дисципліни загалом визначається системою вимог до здобувача вищої освіти, що передбачені чинними положеннями Університету про організацію і навчально-методичне забезпечення освітнього процесу. Зокрема, проходження інструктажу з техніки безпеки; відвідування занять з дисципліни є обов'язковим. За об'єктивних причин (підтверджених документально) теоретичне навчання за погодженням із лектором може відбуватись в он-лайн режимі. Успішне опанування дисципліни і формування фахових компетентностей і програмних результатів навчання передбачає необхідність підготовки до лабораторного заняття (вивчення теоретичного матеріалу з теми роботи, підготовку до усного опитування для допуску до заняття (наведені у Методичних рекомендаціях до лабораторних занять)), активно працювати на занятті, якісно підготувати звіт, захистити результати виконаної роботи, брати участь у дискусіях щодо прийнятих рішень при виконанні здобувачами лабораторних робіт тощо.

Здобувачі вищої освіти мають дотримуватися встановлених термінів виконання всіх видів навчальної роботи відповідно до робочої програми навчальної дисципліни. Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на наступному після виконання роботи занятті. Пропущене лабораторне заняття студент зобов'язаний відпрацювати у встановлений викладачем термін, але не пізніше, ніж за два тижні до кінця теоретичних занять у семестрі.

Засвоєння студентом теоретичного матеріалу з дисципліни перевіряється в процесі поточного контролю під час лабораторних занять. Виконання індивідуального завдання завершується його презентацією у терміни, встановлені графіком самостійної роботи.

Здобувач вищої освіти, виконуючи індивідуальну роботу з дисципліни, має дотримуватися політики доброчесності (заборонені списування, плагіат (в т.ч. із використанням мобільних девайсів)). У разі виявлення порушення політики академічної доброчесності в будь-яких видах навчальної роботи здобувач вищої освіти отримує незадовільну оцінку і має повторно виконати

завдання з відповідної теми (виду роботи), що передбачені робочою програмою. Будь-які форми порушення академічної доброчесності **не допускаються**.

У межах вивчення навчальної дисципліни здобувачам вищої освіти передбачено визнання і зарахування результатів навчання, набутих шляхом неформальної освіти, що розміщені на доступних платформах, які сприяють формуванню компетентностей і поглибленню результатів навчання, визначених робочою програмою дисципліни, або забезпечують вивчення відповідної теми та/або виду робіт з програми навчальної дисципліни (детальніше у Положенні про порядок визнання та зарахування результатів навчання здобувачів вищої освіти у ХНУ).

Як результат виконання лабораторних робіт №1-4 зараховуються онлайн-курси:

– Getting Started with Go

<https://www.coursera.org/learn/golang-getting-started>

– Learn Go (Codecademy)

<https://www.codecademy.com/learn/learn-go>

Як результат виконання лабораторних робіт №5-8 зараховуються онлайн-курси:

– DevOps Beginner Self-Paced Course (EPAM)

<https://campus.epam.ua/ua/training/3473>

– Introduction to Containers w/ Docker, Kubernetes & OpenShift (IBM)

<https://www.coursera.org/learn/ibm-containers-docker-kubernetes-openshift>

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>помилки</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								Самостійна робота		Семестровий контроль	Разом
Лабораторні роботи №:								ІДЗ		Іспит	Сума балів
1	2	3	4	5	6	7	8	1	2		
Кількість балів за вид навчальної роботи (мінімум-максимум)											
3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5	6-10	6-10	24-40	60-100*
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 й оформлена відповідно до встановлених методичними рекомендаціями вимог лабораторна робота комплексно оцінюється викладачем при її захисті з урахуванням таких критеріїв: самостійність та правильність виконання; знання теоретичного матеріалу; здатність аргументувати прийняті рішення; повнота та якість отриманих результатів; дотримання вимог до оформлення та структурування роботи; уміння виправляти виявлені помилки у ході захисту.

Результат виконання і захисту здобувачем вищої освіти кожної лабораторної роботи оцінюється відповідно до таблиці "**Критеріїв оцінювання навчальних досягнень здобувача вищої освіти**" (мінімальний позитивний бал – 3 бали, максимальний – 5 балів).

У випадку виявлення здобувачем рівня знань, нижчого ніж 60 відсотків від максимального балу, встановленого Робочою програмою для кожної структурної одиниці, лабораторна робота йому *не зараховується* і для її захисту він має детальніше опрацювати матеріал з теми роботи, методику її виконання, виправити грубі помилки та повторно вийти на її захист у призначений для цього викладачем час.

Оцінювання результатів виконання індивідуального домашнього завдання

Виконане та оформлене відповідно до вимог, визначених Методичними рекомендаціями, індивідуальне домашнє завдання (ІДЗ) комплексно оцінюється викладачем з урахуванням таких критеріїв: самостійність виконання; правильність розв'язання поставлених задач; обґрунтованість вибору методів розв'язання; повнота пояснень та аргументованість відповідей; якість оформлення та дотримання вимог до структури і змісту роботи.

Результат виконання здобувачем вищої освіти кожного ІДЗ оцінюється відповідно до таблиці **Критеріїв оцінювання навчальних досягнень здобувача вищої освіти** з урахуванням рівня досягнення запланованих програмних результатів навчання та сформованих компетентностей. За підсумками захисту присвоюється відповідна сума балів (мінімальний позитивний бал – 6 балів, максимальний – 10 балів).

У разі, якщо здобувач вищої освіти виявив рівень знань і виконання ІДЗ, що нижчий ніж 60 відсотків від максимальної кількості балів, встановленої Робочою програмою для цієї структурної одиниці, завдання не зараховується. У такому випадку студент має повторно опрацювати зміст завдання, усунути помилки та здати на перевірку доопрацьоване ІДЗ у терміни, погоджені з викладачем.

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	6	8	10
Теоретичне питання № 2	6	8	10
Практичне завдання	12	16	20
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	Відмінно/Excellent – 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		Добре/Good – середній (максимально достатній) рівень досягнення запланованих результатів навчання з навчальної дисципліни та готовності до подальшого навчання та/або професійної діяльності за фахом Задовільно/Satisfactory – Наявні мінімально достатні для подальшого навчання та/або професійної діяльності за фахом результати навчання з навчальної дисципліни
C	73-82		
D	66-72		
E	60-65		
FX	40-59	Незараховано	Незадовільно/Fail – 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		Незадовільно/Fail – Результати навчання відсутні

10. Питання для самоконтролю результатів навчання

1. Що таке системне програмне забезпечення та які його основні критерії?
2. Наведіть 3-4 приклади системного ПЗ.
3. Що таке компіляція в нативний код і чому це важливо для системних мов?
4. Поясніть поняття "незалежність збірки від іншого ПЗ".
5. Що таке операційна система?
6. Поясніть дві ключові функції ОС: як менеджера ресурсів та як розширеної машини.
7. У чому полягає різниця між привілейованим режимом (режимом ядра) та режимом користувача?
8. Назвіть основні підсистеми сучасної операційної системи.
9. Охарактеризуйте поняття "процес" з точки зору ОС.
10. Яка інформація зберігається в блоці керування процесом (PCB)?
11. Опишіть п'ять основних станів процесу.
12. Що таке перемикання контексту і які накладні витрати з ним пов'язані?
13. Чим потік (thread) відрізняється від процесу?
14. Які ресурси є спільними для потоків одного процесу, а які – унікальними для кожного потоку?
15. Поясніть різницю між потоками рівня користувача та потоками рівня ядра.
16. Яку роль виконує планувальник процесів?
17. Що таке "ситуація змагання" (race condition)? Наведіть приклад.
18. Поясніть поняття критичної області та вимоги до її реалізації (взаємне виключення, прогрес, обмежене очікування).
19. Що таке м'ютекс і як він забезпечує взаємне виключення?
20. Порівняйте м'ютекс та семафор. У чому їхня ключова відмінність?
21. Опишіть класичну проблему "виробника та споживача".
22. Для чого використовуються умовні змінні і з яким примітивом синхронізації вони завжди працюють у парі?
23. Поясніть модель конкурентності в Go: що таке горутини та канали?
24. У чому різниця між буферизованим та небуферизованим каналом в Go?
25. Що таке адресний простір процесу?
26. Поясніть різницю між віртуальною та фізичною адресою. Який апаратний компонент виконує їх перетворення?
27. Що таке сторінкова організація пам'яті (paging)? Що таке сторінка та сторінковий кадр?
28. Яке призначення має таблиця сторінок?
29. Як працює буфер швидкого перетворення адреси (TLB) і яку проблему він вирішує?
30. Що таке "відмова сторінки" (page fault) і як ОС її обробляє?
31. Опишіть концепцію алгоритму заміщення сторінок LRU (Least Recently Used).

32. Що таке сегментація пам'яті і які переваги вона надає порівняно зі сторінковою організацією?
33. Поясніть, що таке файл з точки зору операційної системи.
34. Назвіть 4-5 поширених атрибутів файлу.
35. Що таке директорія (каталог) і як вона допомагає організувати файлову систему?
36. Опишіть три основні методи розміщення файлів на диску: неперервний, зв'язаний та індексний.
37. Що таке файловий дескриптор (або handle у Windows)?
38. Як операційна система відстежує вільні блоки на диску?
39. Яка головна мета підсистеми вводу-виводу?
40. Поясніть роль драйвера пристрою.
41. Порівняйте блокуючий (синхронний) та неблокуючий (асинхронний) ввід-вивід.
42. Що таке "тупик" (deadlock)?
43. Назвіть чотири необхідні умови для виникнення тупика.
44. Опишіть практичний спосіб запобігання тупикам у програмному коді.
45. Назвіть основні рівні моделі TCP/IP.
46. Яка функція транспортного рівня?
47. Порівняйте протоколи TCP та UDP за надійністю, наявністю з'єднання та швидкістю.
48. Що таке IP-адреса та номер порту? Яка їхня спільна роль?
49. Поясніть процес інкапсуляції даних при передачі по мережі.
50. Яке призначення системи доменних імен (DNS)?
51. Що таке сокет у контексті мережевого програмування?
52. Опишіть послідовність системних викликів для встановлення TCP-з'єднання на боці сервера.
53. Опишіть послідовність системних викликів для встановлення TCP-з'єднання на боці клієнта.
54. Чим відрізняється надсилання та отримання даних через UDP-сокет від TCP-сокета?
55. Яка різниця між потоковим (stream) та датаграмним (datagram) сокетом?
56. Яка роль функції bind() в налаштуванні сервера?
57. Що повертає функція accept() і чому це важливо?
58. Опишіть базову структуру HTTP-запиту (start line, headers, body).
59. Опишіть базову структуру HTTP-відповіді. Що означають коди стану 200, 404 та 500?
60. Назвіть 4 основні методи HTTP та їхнє призначення.
61. Яку роль відіграє формат JSON у сучасних мережевих API?
62. Як реалізувати простий HTTP-сервер за допомогою стандартної бібліотеки Go?
63. Чому паралелізм є критично важливим для ефективної роботи мережевих серверів?
64. Опишіть модель "один потік на з'єднання" та її основні недоліки.
65. Як модель конкурентності Go (горутини та канали) допомагає уникнути цих недоліків?
66. Поясніть, як можна обробити декілька клієнтських з'єднань одночасно в Go-програмі.
67. Що таке віртуалізація і які основні проблеми вона вирішує?
68. Поясніть різницю між гіпервізором 1-го (bare-metal) та 2-го (hosted) типу.
69. У чому полягає роль апаратної підтримки (наприклад, Intel VT-x) у сучасній віртуалізації?
70. Як віртуалізація є основою для хмарної моделі IaaS (Infrastructure as a Service)?
71. Порівняйте віртуальні машини та контейнери за рівнем ізоляції та споживанням ресурсів.
72. Що таке віртуалізація на рівні операційної системи?
73. Поясніть роль просторів імен (namespaces) та контрольних груп (cgroups) у технології контейнерів.
74. Що таке образ контейнера (container image)?
75. Яке призначення має Dockerfile?
76. Що таке реєстр контейнерів (container registry)?
77. Що таке оркестрація контейнерів?
78. Яку основну проблему вирішує Kubernetes?
79. Що таке Pod в термінології Kubernetes?
80. Для чого потрібен Service у Kubernetes?

11. Навчально-методичне забезпечення

Освітній процес з дисципліни «Системне та мережеве програмне забезпечення» повністю і в достатній кількості забезпечений необхідною навчально-методичною літературою. Зокрема, викладачами кафедри підготовлені і видані такі роботи:

1. Багрій Р.О., Пасічник О.А., Скрипник Т.К., Вознюк Л.О. Системне та мережеве програмне забезпечення. Теоретичний курс та лабораторний практикум: навчальний посібник. – Хмельницький : ХНУ, 2024. – 125 с.

12. Матеріально-технічне та програмне забезпечення дисципліни

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 потребує використання програмних засобів для системного та мережевого програмування. Основний інструментарій включає: мову програмування Go, середовище розробки Visual Studio Code та систему контейнеризації Docker Desktop. Усі зазначені програмні продукти є безкоштовними та вільно розповсюджуваними для навчальних цілей.

13. Рекомендована література:

Основна

1. Багрій Р.О., Пасічник О.А., Скрипник Т.К., Вознюк Л.О. Системне та мережеве програмне забезпечення. Теоретичний курс та лабораторний практикум: навчальний посібник. – Хмельницький : ХНУ, 2024. – 125 с.

2. Tanenbaum A. Modern operating systems. – Global ed. – Harlow: Pearson Education Limited, 2023. – 1184 p. – ISBN 978-1292459662.

3. Bodner J. Learning Go: an idiomatic approach to real-world Go programming. – 2nd ed. – Sebastopol, CA: O'Reilly Media, 2024. – 564 p. – ISBN 9781098139285.

4. Tsoukalos M. Mastering Go: leverage Go's expertise for advanced utilities, empowering you to develop professional software. – 4th ed. – Birmingham: Packt Publishing, 2024. – 740 p. – ISBN 978-1805127147.

5. Poulton N. Docker deep dive. – London: Nielsen Book, 2024. – 307 p. – ISBN 978-1916585379.

6. Domingus J., Arundel J. Cloud Native DevOps with Kubernetes: Building, Deploying, and Scaling Modern Applications in the Cloud. – 2nd ed. – Sebastopol: O'Reilly Media, 2022. – 353 p. – ISBN 978-1098115470.

Додаткова

1. Stallings W. Operating Systems: Internals and Design Principles. 9th ed. – Pearson, 2018. – 912 p. – ISBN 978-0134670959.

2. Guerrieri A. Hands-On System Programming with Go. – Packt Publishing, 2019. – 660 p. – ISBN 978-1789804072.

3. Gregg B. Systems Performance: Enterprise and the Cloud. 2nd ed. – Addison-Wesley, 2020. – 840 p. – ISBN 978-0136821694.

4. Woodbeck A. Network Programming with Go. – No Starch Press, 2021. – 392 p. – ISBN 978-1718500884.

5. Donovan A. A., Kernighan B. W. The Go Programming Language. – Addison-Wesley, 2015. – 400 p. – ISBN 978-0134190440.

6. Офіційна документація мови програмування Go [Електронний ресурс]. – Режим доступу: <https://go.dev/doc/>

7. Go by Example: Інтерактивні приклади коду на Go [Електронний ресурс]. – Режим доступу: <https://gobyexample.com/>

8. Beej's Guide to Network Programming [Електронний ресурс]. – Режим доступу: <https://beej.us/guide/bgnet/>

9. Docker Documentation [Електронний ресурс]. – Режим доступу: <https://docs.docker.com/>

10. Kubernetes Documentation [Електронний ресурс]. – Режим доступу: <https://kubernetes.io/docs/>
11. MDN Web Docs: Довідник з протоколу HTTP [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/HTTP>
12. How To Code in Go (DigitalOcean Community Tutorials) [Електронний ресурс]. – Режим доступу: https://www.digitalocean.com/community/tutorial_series/how-to-code-in-go
13. Julia Evans' blog: Wizard zines about systems & networking [Електронний ресурс]. – Режим доступу: <https://wizardzines.com/>
14. The Illustrated Children's Guide to Kubernetes [Електронний ресурс]. – Режим доступу: <https://www.cncf.io/the-illustrated-childrens-guide-to-kubernetes/>
15. Concurrency in Go (Ardan Labs) [Електронний ресурс]. – Режим доступу: <https://www.ardanlabs.com/blog/2014/01/concurrency-in-go-tools-and-techniques.html>
16. Brendan Gregg's Blog: Systems Performance [Електронний ресурс]. – Режим доступу: <https://www.brendangregg.com/blog/index.html>
17. Мазурець О., Молчанова М., Собко О., Дідур В., Багрій Р. Архітектура системного програмного забезпечення мережевого сервісу для агрегації спортивних новин. *Наука і техніка сьогодні. Серія «Техніка»*. 2025. № 8 (49). С. 1542–1555. DOI: [https://doi.org/10.52058/2786-6025-2025-8\(49\)-1542-1555](https://doi.org/10.52058/2786-6025-2025-8(49)-1542-1555)
18. Багрій Р., Бармак О., Манзюк Е. Підвищення стійкості паролів у веб-системах за допомогою вдосконалених схем хешування. *Herald of Khmelnytskyi National University. Technical Sciences*. 2024. № 1 (331). С. 48–51. DOI: <https://doi.org/10.31891/2307-5732-2024-331-6>

14. Інформаційні ресурси

1. Модульне середовище для навчання. URL : <https://msn.khmnu.edu.ua/course/view.php?id=2110>
2. Електронна бібліотека ХНУ. URL: <http://library.khmnu.edu.ua/>
3. Інституційний репозитарій ХНУ. URL : <https://elar.khmnu.edu.ua/home>

СИСТЕМНЕ ТА МЕРЕЖЕВЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

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

Результати навчання. Після вивчення дисципліни студент повинен: володіти понятійним апаратом у сфері системного та мережевого програмного забезпечення, зокрема фундаментальними принципами функціонування операційних систем; уміти розробляти програмні рішення, що взаємодіють з ключовими компонентами ОС, керувати життєвим циклом процесів і потоків та синхронізувати їх доступ до спільних ресурсів для забезпечення цілісності даних; проєктувати та реалізовувати ефективні багатопотокові клієнт-серверні застосунки, використовуючи API сокетів та протоколи прикладного рівня; застосовувати технології віртуалізації та контейнеризації для пакування, розгортання та безпечної ізоляції програмного забезпечення, виконуючи розробку системних утиліт і мережевих сервісів для сучасних обчислювальних середовищ.

Зміст навчальної дисципліни. Вступ до системного та мережевого програмного забезпечення; принципи функціонування операційних систем, включаючи керування процесами, потоками, їх синхронізацію, віртуальну пам'ять та файлові системи; основи мережевого програмування на базі TCP/IP, розробка паралельних клієнт-серверних застосунків з використанням сокетів, протоколу HTTP та горутин; сучасні технології розгортання ПЗ, зокрема віртуалізація, контейнеризація застосунків за допомогою Docker та основи їх оркестрації.

Пререквізити: алгоритмізація та програмування, дослідження операцій та основи теорії прийняття рішень, основи програмної інженерії та тестування програмного забезпечення, технології захисту інформації та кібербезпека.

Кореквізити: технології розподілених систем та паралельних обчислень, проєктно-технологічна практика, професійна практика, кваліфікаційна робота.

Запланована навчальна діяльність: Мінімальний обсяг навчальних занять в одному кредиті ЄКТС навчальної дисципліни для *першого* (бакалаврського) рівня вищої освіти за денною формою здобуття освіти становить 10 годин на 1 кредит ЄКТС.

Форми (методи) навчання: лекції (з використанням мультимедійних презентацій, методів візуалізації, пояснення, проблемного й інтерактивного навчання, методів стимулювання і мотивації, інформаційно-комунікаційних технологій, інтенсифікації та індивідуалізації навчання); лабораторні заняття (з використанням методів комп'ютерного моделювання, методів проєктної діяльності, аналізу проблемних ситуацій, пояснення, дискусії); самостійна робота (робота над засвоєнням теоретичного матеріалу, виконання індивідуальних завдань, підготовка до виконання та захисту лабораторних робіт, опрацювання навчально-методичних матеріалів, розміщених у модульному середовищі навчання).

Форми оцінювання результатів навчання: оцінювання лабораторних робіт; виконання та захист індивідуального завдання.

Вид семестрового контролю: іспит – 5 семестр.

Навчальні ресурси:

1. Багрій Р.О., Пасічник О.А., Скрипник Т.К., Вознюк Л.О. Системне та мережеве програмне забезпечення. Теоретичний курс та лабораторний практикум: навчальний посібник. – Хмельницький : ХНУ, 2024. – 125 с.
2. Tanenbaum A. Modern operating systems. – Global ed. – Harlow: Pearson Education Limited, 2023. – 1184 p. – ISBN 978-1292459662.
3. Bodner J. Learning Go: an idiomatic approach to real-world Go programming. – 2nd ed. – Sebastopol, CA: O'Reilly Media, 2024. – 564 p. – ISBN 9781098139285.
4. Tsoukalos M. Mastering Go: leverage Go's expertise for advanced utilities, empowering you to develop professional software. – 4th ed. – Birmingham: Packt Publishing, 2024. – 740 p. – ISBN 978-1805127147.
5. Poulton N. Docker deep dive. – London: Nielsen Book, 2024. – 307 p. – ISBN 978-1916585379.
6. Domingus J., Arundel J. Cloud Native DevOps with Kubernetes: Building, Deploying, and Scaling Modern Applications in the Cloud. – 2nd ed. – Sebastopol: O'Reilly Media, 2022. – 353 p. – ISBN 978-1098115470.
7. Модульне середовище для навчання. URL: <https://msn.khmnmu.edu.ua/course/view.php?id=1904>
8. Електронна бібліотека університету. URL: <http://library.khmnmu.edu.ua/>

Викладачі: канд. техн.наук, доцент Руслан БАГРІЙ, канд. техн.наук, доцент Олександр ПАСІЧНИК