

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра програмних систем і технологій

«ЗАТВЕРДЖУЮ»

Заступник декана

з навчально-виховної роботи

Наталія ТМЄНОВА

«29» серпня 2026 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
СУЧАСНІ ТЕХНОЛОГІЇ СИСТЕМНОГО ПРОГРАМУВАННЯ**

для студентів

галузь знань	12 Інформаційні технології
спеціальність	121 Інженерія програмного забезпечення
освітній рівень	бакалавр
освітня програма	Інженерія програмного забезпечення

вид дисципліни **вибіркова**

Форма навчання	денна
Навчальний рік	2026/2027
Семестр	5
Кількість кредитів ECTS	3
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладачі: **Васильєв О.М., професор**

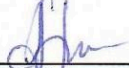
Пролонговано: на 20__/20__ н.р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)

на 20__/20__ н.р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)

КИЇВ – 2026

Розробник(и): **Васильєв Олексій Миколайович**, доктор фізико-математичних наук, професор кафедри програмних систем і технологій

Завідувач кафедри
програмних систем і технологій

 проф. Олексій БИЧКОВ
(підпис)

Протокол № 18 від 17 червня 2026 року

Схвалено науково-методичною комісією факультету інформаційних технологій

Протокол № 11 від 26 червня 2026 року

Голова науково-методичної комісії  Ганна КРАСОВСЬКА
(підпис)

1. Мета дисципліни "Сучасні технології системного програмування" полягає у формуванні у студентів фундаментальних та практичних знань щодо розробки системного програмного забезпечення, вивченні низькорівневих механізмів взаємодії з операційними системами, ефективного управління ресурсами (пам'ять, процеси, потоки, I/O), засобів міжпроцесної взаємодії (IPC), а також сучасних системних мов програмування.

2. Попередні вимоги до опанування або вибору навчальної дисципліни:

Володіти базовими знаннями з архітектури обчислювальних систем, принципів функціонування операційних систем, базовими навичками програмування мовами C/C++, розуміти концепції адресації пам'яті та структури даних.

3. Анотація навчальної дисципліни:

Навчальна дисципліна "Сучасні технології системного програмування" є складовою циклу професійної підготовки фахівців освітнього рівня "бакалавр". Вона надає студентам систематизовані знання та практичні навички створення високопродуктивного, безпечного та надійного системного програмного забезпечення. Розглядаються системні виклики OS, низькорівневе управління пам'яттю, багатопотоковість, синхронізація, системні інтерфейси, а також безпечні підходи до системного програмування.

Результати навчання полягають в знанні базових прийомів для створення ефективного програмного забезпечення системного рівня. Методи викладання: лекції, лабораторні заняття та самостійна робота студентів. Методи оцінювання: оцінювання лабораторних робіт, контрольні роботи після основних розділів курсу, іспит. Підсумкова оцінка виставляється на основі проміжних оцінок (60%) та іспиту (40%).

4. Завдання (навчальні цілі): засвоєння методологій побудови системних компонентів, оволодіння інструментами системного аналізу, налагодження та профілювання системного програмного забезпечення, розвиток здатності проектувати продуктивні та безпечні системні сервіси.

Загальні компетентності

ЗК-1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК-2. Здатність застосовувати знання у практичних ситуаціях.

Спеціальні (фахові, предметні) компетентності

ФК-1. Здатність ідентифікувати, класифікувати та формувати вимоги до програмного забезпечення.

ФК-2. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

ФК-4. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.

ФК-6. Здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки.

ФК-8. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.

ФК-12. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності загальної функціональності і надійності програмного забезпечення.

ФК-13. Здатність обґрунтовано обирати та обстоювати інструментарій з розробки та супроводження програмного забезпечення.

ФК-15. Здатність розуміти процеси побудови архітектури програмної системи та архітектурні стилі та застосовувати інструменти проектування архітектури системи, прототипувати та оцінювати архітектурні рішення.

ФК-17. Здатність створювати та супроводжувати захищене програмне забезпечення відповідно до вимог сучасних стандартів кібербезпеки та критеріїв безпеки захищених програм.

5. Результати навчання за дисципліною:

Результат навчання (1. знати; 2. вміти; 3. комунікація)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
1.1	Знати, розуміти і застосовувати ефективні підходи щодо проєктування програмного забезпечення	Лекції, лабораторні роботи, самостійна робота	Тести, звіти з лабораторних, усні опитування, контрольні роботи, залік, іспит	10
1.2	Вміння аналізувати проблеми щодо створення програмного забезпечення Мотивовано обирати мови програмування та технології для розв'язання завдань створення і програмного забезпечення			10
1.3	Мотивовано обирати мови програмування для розв'язання завдань створення і супроводження програмного забезпечення			20
2.1	Мати навички проєктування програмних продуктів			20
2.2	Мати навички реалізації основних парадигм реалізації програмних продуктів			20
2.3	Мати навички тестування програмного забезпечення			20

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	1.1	1.2	1.3	2.1	2.2	2.3
Програмні результати навчання						
ПР-25. Знати архітектуру програмного забезпечення та принципи проєктування, володіти технологіями та методологіями розробки та реалізації програмного забезпечення та мережними технологіями, мати практичні навички з адміністрування комп'ютерних мереж в процесі розробки, розгортання та супроводження програмного забезпечення, розуміти і застосовувати програмне забезпечення мережевого обміну, обміну з сенсорами, віддаленої та розподіленої обробки інформації в інтересах інтернету речей	+	+	+	+	+	+

7. Схема формування оцінки

7.1. Для контролю знань студентів використовується модульно-рейтингова система. Результати навчання студентів оцінюються за 100-бальною шкалою. Семестр складається одного змістового модуля. Підсумковий контроль (іспит) проводиться у письмовому вигляді. Максимальна оцінка за іспит – 40 балів.

Обов'язковим є виконання та захист студентом лабораторних робіт, що передбачені робочою програмою навчальної дисципліни, здача модульної контрольної роботи. При цьому сумарна мінімальна кількість балів за семестр – 36.

Тип роботи	Мін. – 36 балів	Макс. – 60 балів
Виконання та захист лабораторних робіт	30	50
Модульна контрольна робота	6	10

Для студентів, які набрали сумарно меншу кількість балів ніж критично-розрахунковий мінімум (36 балів), для одержання допуску до іспиту є перездача модульної контрольної роботи, виконання та захист лабораторних робіт.

Замість виконання та захисту лабораторних робіт студенти можуть запропонувати для виконання індивідуальні та колективні проекти. Умови виконання проекту (чи проектів) такі:

- Тематика та виконавці проекту попередньо узгоджуються з викладачем.
- Результати проекту мають перекривати тематику декількох лабораторних робіт (не обов'язково всіх).
- У випадку реалізації проекту він зараховується замість тих лабораторних робіт, які перекриваються результатами проекту.
- Максимальна оцінка за проект дорівнює сумарній максимальній оцінці за лабораторні роботи, які перериваються результатами проекту.
- Якщо виконавців проекту декілька, оцінка кожному з них виставляється індивідуально з урахуванням внеску в проект та підготовкою під час захисту проекту.
- Лабораторні, які не перекриваються результатами проекту, здаються в індивідуальному порядку або в рамках іншого проекту.

- семестрове оцінювання:

Модульна контрольна робота: 10 балів.

- підсумкове оцінювання (у формі іспиту):

- максимальна кількість балів які можуть бути отримані студентом 40 балів на іспиті за 100-бальною шкалою;
- студент не допускається до іспиту, якщо під час семестру набрав менше ніж 36 балів;
- для отримання загальної позитивної оцінки з дисципліни оцінка за іспит не може бути меншою за 24 бали.

7.2. Організація оцінювання:

Упродовж семестру, після завершення лекційного матеріалу, проводиться письмова модульна контрольна робота з відкритими питаннями або у вигляді тесту.

Іспит проводиться письмово у вигляді відкритих питань або у вигляді тесту.

7.3. Шкала відповідності оцінок

За умови іспиту:

Відмінно	90-100
Добре	75-89
Задовільно	60-74
Незадовільно	0-59

8. Структура навчальної дисципліни. Тематичний план лекцій, практичних занять та самостійних робіт

5 СЕМЕСТР

№ теми	Назва лекції	Кількість годин		
		лекції	лабораторні	Самост. робота
1	Вступ до курсу. Призначення курсу. Особливості системного програмування на сучасному етапі	2	2	4
2	Методи реалізації системного програмування. Мови та технології для впровадження програмних рішень на системному рівні	2	2	8
3	Операційні системи, їхня архітектура, можливості щодо застосування програмних рішень	2	2	10
4	Ресурси, система введення/виведення, системні виклики та процеси	2	2	10
5	Керування пам'яттю та суміжні програмні задачі, комунікація процесів	2	2	10
6	Основні проблеми та задачі у системному програмуванні та шляхи їх вирішення	2	2	10
7	Можливості і перспективи використання сучасних технологій для системного програмування	2	2	8
	Модульна контрольна робота		2	
Всього		14	16	60

Загальний обсяг 90 год, в тому числі:

Лекцій – **14 год**.

Практичні заняття - **16 год**.

Самостійна робота – **60 год**.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА:

Основна

[1] Олексій Васильєв. Структурне програмування мовою C. "Ліра-К", Київ, 2025, 390 с.

[2] Бочкарьов О.Ю. Системне програмування в Linux. "Магнолія 2006", Львів, 229 с.

[3] Stewart Weiss, System Programming in Linux: A Hands-On Introduction. "No Starch Press", New York, 2025, 1048 p.

Додаткова

[1] Бичков О.С. Основи сучасного програмування. ВПЦ "Київський університет", 2007, 260 с.

[2] Олексій Васильєв. Програмування на C++ в прикладах і задачах. "Ліра-К", Київ, 2021, 382 с.

[3] Robert Love. Linux System Programming: Talking Directly to the Kernel and C Library. "O'Reilly Media", New York, 2013, 454 с.