

Міністерство освіти і науки України
Національний аерокосмічний університет
«Харківський авіаційний інститут»

Кафедра математичного моделювання та штучного інтелекту (№ 304)

ЗАТВЕРДЖУЮ

Гарант освітньої програми



Дмитро ЧУМАЧЕНКО

(підпис)

(ім'я та

ПРИЗВИЩЕ)

« 21 » 08 2025 р.

**СИЛАБУС ОБОВ'ЯЗКОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Системне програмування

(назва навчальної дисципліни)

Галузь знань: 12 «Інформаційні технології»

(шифр і найменування галузі знань)

Спеціальність: 122 «Комп'ютерні науки»

(код і найменування спеціальності)

Освітня програма: «Інтелектуальні системи та технології»

(найменування освітньої програми)

Рівень вищої освіти: *перший (бакалаврський)*

Силабус уведено в дію з 01.09.2025

Харків – 2025 р.

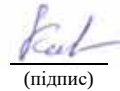
Розробник: Коробчинський К.П., доц. каф. 304, к. т. н., доц.
(прізвище та ініціали, посада, науковий ступінь і вчене звання)
(підпис)



Силабус навчальної дисципліни розглянуто на засіданні кафедри № 304
математичного моделювання та штучного інтелекту
(назва кафедри)

Протокол № 1 від « 21 » серпня 2025 р.

В. о. завідувача кафедри к.ф.-м.н., доцент
(науковий ступінь і вчене звання)



Олексій КАРТАШОВ
(ім'я та ПРІЗВИЩЕ)

Погоджено з представником здобувачів освіти:

Здобувач першого (бакалаврського)
рівня вищої освіти за ОП
«Інтелектуальні системи та технології»
(ім'я та ПРІЗВИЩЕ)



Катерина ПИЛИПЕНКО
(підпис)

1. Загальна інформація про викладача



ПІБ: *Коробчинський Кирил Петрович*

Посада: *доцент каф. 304*

Науковий ступінь: *канд. техн. наук*

Вчене звання: *доцент*

Перелік дисциплін, які викладає:

*Програмування та алгоритмічні мови;
Системне програмування; Базы даних та
інформаційні системи; Організація баз даних;
Нереляційні бази даних; Git для розподіленої
розробки ПЗ; Використання Git для розробки
програмного забезпечення*

Напрями наукових досліджень:

*Базы даних, адміністрування, web-
програмування*

Контактна інформація:

k.korobchinskiy@khai.edu

2. Опис навчальної дисципліни

Форма здобуття освіти	Денна
Семестр	5 (3 для СТ груп)
Мова викладання	Українська
Тип дисципліни	Обов'язкова
Обсяг дисципліни: кредити ЄКТС/ кількість годин	3 (3,5) кредита ЄКТС / 90 (105) годин (48 аудиторних, з яких: лекції – 16, практичні – 32; СРЗ – 42 (57));
Види навчальної діяльності	Лекції та практичні заняття, самостійна робота
Види контролю	Поточний контроль, модульний контроль, семестровий контроль – іспит
Пререквізити	“Дискретна математика”, “Програмування та алгоритмічні мови”, “Теорія алгоритмів та математична логіка”, “Операційні системи”

3. Мета та завдання навчальної дисципліни, переліки компетентностей та очікуваних результатів навчання

Мета: вивчення архітектури сучасних операційних систем, принципів взаємодії програмного забезпечення з апаратними складовими обчислювальних систем, а також набуття навичок розробки системного та прикладного ПЗ низького рівня.

Придбання студентами знань з основних механізмів керування ресурсами (процесами, пам'яттю, файловими системами) за допомогою системних викликів API, а також знайомство з основними прийоми, засобами та підходами системного програмування в середовищі сімейства Unix-подібних операційних систем (ОС) Linux, а також побудови багатопотокових алгоритмів та засобів міжпроцесної взаємодії мовами C/C++.

Мета досягається за рахунок сполучення таких форм навчання, як лекції, практичні роботи, а також керованої самостійної роботи студентів.

Основними завданнями вивчення дисципліни «Системне програмування» є:

- Формування системи теоретичних знань щодо фундаментальних стандартів системного програмування в середовищі Linux, зокрема специфікацій POSIX (Portable Operating System Interface), SUS (Single UNIX Specification) та LSB (Linux Standard Base).
- Опанування методології та інструментарію мови програмування C для розроблення ефективного системного програмного забезпечення, що функціонує на рівні взаємодії з ядром операційної системи.
- Засвоєння практичних навичок використання багаторівневих інтерфейсів програмування (API): від низькорівневих системних викликів введення-виведення до високорівневих функцій стандартних бібліотек для маніпуляції об'єктами файлової системи.
- Вивчення архітектурних особливостей та механізмів керування життєвим циклом процесів в ОС Linux, включаючи їх створення, синхронізацію та міжпроцесну взаємодію.
- Розвиток компетенцій у сфері паралельних обчислень, що передбачає проєктування та реалізацію багатопотокових програмних систем із застосуванням сучасних механізмів диспетчеризації та засобів запобігання конфліктам доступу до ресурсів.

Компетентності, які набуваються:

Загальні компетентності (ЗК)

Після закінчення цієї програми здобувач освіти буде здатен:

- ЗК1. Здатність до абстрактного мислення, аналізу та синтезу;
- ЗК2. Здатність застосовувати знання у практичних ситуаціях;
- ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності;
- ЗК5. Здатність спілкуватися іноземною мовою;
- ЗК8. Здатність генерувати нові ідеї (креативність);
- ЗК9. Здатність працювати в команді.

Фахові компетентності (СК):

Після закінчення цієї програми здобувач освіти буде здатен:

– СК3.Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проєктування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

– СК7.Здатність застосовувати теоретичні та практичні основи методології та технології моделювання для дослідження характеристик і поведінки складних об'єктів і систем, проводити обчислювальні експерименти з обробкою й аналізом результатів.

– СК8.Здатність проєктувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

– СК12.Здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення.

Програмні результати навчання (ПРН):

– ПРН5. Проєктувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислювальних функцій.

– ПРН13. Володіти мовами системного програмування та методами 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.

4. Зміст навчальної дисципліни

МОДУЛЬ 1

Змістовий модуль 1. Теоретичні засади та інструментарій системного програмування

Тема 1. Вступ до системного програмування. Визначення та предметна область системного програмування. Генезис обчислювальної техніки та еволюція архітектурних рішень. Класифікація програмного забезпечення: диференціація прикладного та системного рівнів. Огляд мов програмування, адаптованих для розробки системних компонентів та драйверів.

Тема 2. С-орієнтована парадигма системного розроблення. Історичний аспект та стандартизація мови С (ANSI, ISO). Типологія даних, механізми адміністрування пам'яті за допомогою покажчиків. Архітектурна побудова С-

програм. Функціональна декомпозиція та специфікація передачі параметрів. Реалізація низькорівневих операцій введення-виведення. Роль препроцесора у кросплатформному розробленні. Стандартна бібліотека як фундамент системних викликів.

Тема 3. Концептуальні засади функціонування ОС Linux. Архітектура інтерфейсів прикладного програмування (API). Аналіз галузевих стандартів: POSIX (Portable Operating System Interface), SUS (Single UNIX Specification) та LSB (Linux Standard Base). Системна ієрархія об'єктів: файлова система, процеси, користувачі та групи. Моделі розмежування прав доступу. Теоретичні основи сигнальної обробки та механізми міжпроцесної взаємодії (IPC). Методологія обробки системних помилок.

Змістовий модуль 2. Інтерфейси введення-виведення та керування даними

Тема 1. Низькорівневий файловий ввід-вивід у середовищі Linux. Фундаментальні системні виклики для маніпуляції файловими дескрипторами. Механізми синхронізації операцій введення-виведення для забезпечення цілісності даних. Алгоритми прямого позиціонування (lseek) та динамічного усічення файлових об'єктів.

Тема 2. Високорівневі та оптимізовані методи обміну даними. Архітектура буферизації потокового введення-виведення. Стандартні інтерфейси бібліотеки stdio. Взаємозв'язок потоків даних із файловими дескрипторами. Стратегії керування буферами та примусове скидання (flushing) даних. Спеціалізовані методи оптимізації: векторний (scatter/gather) ввід-вивід. Технологія відображення файлів у віртуальну пам'ять (memory mapping) як засіб підвищення продуктивності дискових операцій.

Модульний контроль.

МОДУЛЬ 2

Змістовий модуль 3. Багатозадачність та моделі паралельних обчислень

Тема 1. Архітектурні засади багатозадачності в ОС Linux. Фундаментальні концепції багатозадачності. Диференціація понять «програма», «процес» та «потік виконання». Ієрархічна структура та дескриптори атрибутів процесів. Механізми ініціалізації нових процесів. Життєвий цикл процесу: специфіка станів «зомбі» та «сирота». Функціонування системних служб (демонів). Принципи диспетчеризації та планування задач. Стратегії керування пріоритетами. Технології прив'язки процесів до обчислювальних ядер (CPU affinity). Огляд операційних систем реального часу (RTOS).

Тема 2. Багатопотокова організація програмних систем. Теоретичні основи потокової декомпозиції. Порівняльний аналіз однопотокових та багатопотокових архітектур: переваги, обмеження та сфери застосування. Моделі багатопотоковості (Many-to-One, One-to-One, Many-to-Many). Проблематика конкурентності: стан гонитви (race condition), взаємоблокування та методи їх нівелювання. Стандартизація інтерфейсу POSIX Threads (Pthreads) для керування паралельними потоками.

Змістовий модуль 4. Керування ресурсами та апаратна ієрархія

Тема 1. Архітектура та адміністрування пам'яті. Структура віртуального адресного простору процесу. Методологія виділення та звільнення динамічної пам'яті. Керування сегментами даних та програмним стеком. Стратегії розширеного керування пам'яттю. Механізми блокування сторінок пам'яті в ОЗП для запобігання свопінгу.

Тема 2. Асинхронна обробка подій за допомогою сигналів. Концептуальна модель сигналів як засобів міжпроцесного сповіщення. Класифікація та семантика системних сигналів. Методи встановлення обробників (handlers). Механізми блокування та маскувння сигналів. Розширені інтерфейси керування сигнальною чергою для забезпечення детермінізму програмних систем.

Тема 3. Рівнева організація обчислювальних систем. Ієрархія рівнів комп'ютерної організації: від цифрової логіки до архітектури системи команд (ISA). Еволюційний розвиток архітектурних рішень. Класифікація сучасних обчислювальних систем. Мікроархітектура центрального процесора. Організація підсистеми пам'яті та рівнів кешування. Принципи функціонування контролерів пристроїв введення-виведення та інтерфейсів взаємодії з периферією.

Модульний контроль.

5. Методи контролю

Визначення рівня засвоєння студентом навчального матеріалу дисципліни здійснюється шляхом проведення поточних (захист практичних робіт) і підсумкових контролів (захист змістовного модуля, іспит).

Поточний контроль має на меті систематично відстежувати, наскільки студент розуміє та засвоює матеріал, вміє виконувати лабораторні роботи, самостійно вирішувати завдання та писати відповідний код, готувати звіти та представляти результати дослідження в усній або письмовій формі.

Підсумковий контроль покликаний оцінити глибину знань студента з усього курсу, його розуміння зв'язків між різними темами, здатність застосовувати отримані знання на практиці та самостійно вирішувати задачі з аналізу даних, які виникають в рамках дисципліни.

Таким чином методами контролю є:

1. Опитування.
2. Практичні роботи.
3. Модульні контрольні роботи.

Форма підсумкового контролю успішності навчання: іспит у 5 (3) сем.

6. Критерії оцінювання та розподіл балів, які отримують здобувачі освіти

Таблиця 6.1 – Розподіл балів, які отримують здобувачі освіти

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Модуль 1			
Робота на лекціях	0...1	5	0...5
Виконання і захист лабораторних (практичних) робіт	0...4	5	0...20
Поточне тестування	0...2	5	0...10
Модульний контроль	0...15	1	0...15
Модуль 2			
Робота на лекціях	0...1	5	0...5
Виконання і захист лабораторних (практичних) робіт	0...4	5	0...20
Поточне тестування	0...15	1	0...15
Модульний контроль	0...2	5	0...10
Усього за семестр			0...100

Семестровий контроль – іспит проводиться у разі відмови студента від балів поточного тестування й за наявності допуску до іспиту/заліку. Під час складання семестрового іспиту/заліку студент має можливість отримати максимум 100 балів.

Білет для іспиту/заліку складається з 4 запитань (теоретичних та практичних). За кожне питання отримає 25 балів (сума – 100 балів).

6.2. Якісні критерії оцінювання

Необхідний обсяг знань для одержання позитивної оцінки:

– *Фундаментальні засади: основні положення та концепції системного програмування, відмінності між системним та прикладним програмним забезпеченням.*

– *Стандартизація: ключові галузеві стандарти розробки системних компонентів (POSIX, SUS, LSB) та їх роль у забезпеченні мобільності ПЗ.*

– *Архітектурні особливості: ієрархічна структура та життєвий цикл процесів і потоків в операційній системі Linux.*

– *Механізми взаємодії: принципи роботи системних інтерфейсів (API), системних викликів та обробки сигналів.*

– *Керування ресурсами: організація віртуального адресного простору процесу, стратегії виділення пам'яті та рівні ієрархії обчислювальної системи.*

– *Системний ввід-вивід: методологія маніпуляції файловими об'єктами через низькорівневі та високорівневі дескрипторні інтерфейси.*

Необхідний обсяг вмінь для одержання позитивної оцінки:

- *Розробка системного ПЗ: проектувати та реалізовувати програмні рішення на мові C з використанням системних бібліотек середовища Linux.*
- *Керування потоками виконання: створювати багатозадачні системи, застосовуючи механізми породження процесів та управління їхнім життєвим циклом (зокрема, обробку станів «зомбі» та роботу з демонами).*
- *Паралельне програмування: розробляти багатопотокові додатки (Pthreads), забезпечуючи коректну синхронізацію та уникаючи конфліктів доступу до спільних ресурсів.*
- *Адміністрування пам'яті: ефективно використовувати засоби динамічного розподілу пам'яті та механізми відображення файлів у пам'ять (memory mapping).*
- *Налагодження та обробка помилок: застосовувати спеціалізований інструментарій для ідентифікації та локалізації системних помилок через аналіз кодів errno та сигнальну обробку.*
- *Оптимізація обміну даними: реалізовувати ефективні алгоритми файлового вводу-виводу, враховуючи особливості буферизації та векторних операцій.*

6.3 Критерії оцінювання роботи студента протягом семестру

Задовільно (60–74 бали). Студент має продемонструвати базовий рівень теоретичної підготовки та володіння фундаментальним інструментарієм системного розробника.

Вимоги: успішний захист усіх практичних робіт та проходження модульного тестування.

Знання: основні положення та термінологія системного програмування, стандарти POSIX, SUS та LSB; базові конструкції мови C та принципи роботи препроцесора.

Уміння: застосовувати інтерфейси прикладного програмування (API) для реалізації базових операцій низькорівневого та високорівневого введення-виведення. Мати навички роботи у терміналі Linux та виконання компіляції програмного коду за допомогою GCC. Вміти використовувати системні виклики для маніпуляції файловими дескрипторами та обробки елементарних станів помилок (errno).

Добре (75–89 балів). Студент має виявити системні знання програмного матеріалу, впевнено володіти методами розробки багатозадачних систем та виконувати завдання без суттєвих методичних помилок.

Вимоги: тверде знання мінімуму, своєчасний захист лабораторних робіт, успішне виконання контрольних робіт, проходження тестування та опрацювання тем, винесених на самостійну роботу.

Знання: архітектурні принципи багатозадачності, життєвий цикл процесів та потоків, механізми керування віртуальною пам'яттю, концепція сигнальної обробки.

Уміння: проектувати багатопотокові додатки (Pthreads) із застосуванням примітивів синхронізації (м'ютекси, семафори) для нівелювання станів гонитви. Вміти здійснювати керування сегментами пам'яті та реалізовувати відображення файлів у пам'ять (memory mapping). Студент повинен вміти організовувати складну ієрархію процесів та володіти методами міжпроцесної взаємодії (IPC).

Відмінно (90–100 балів). Студент має продемонструвати глибоке засвоєння всіх тем дисципліни, здатність до системного аналізу та архітектурного проектування складного системного ПЗ.

Вимоги: здача всіх контрольних точок (лабораторних робіт, модульного контролю) з оцінкою «відмінно».

Знання та уміння: доскональне розуміння взаємодії апаратного забезпечення з ядром операційної системи. Здатність самостійно проектувати оптимізовані системні утиліти, проводити глибоке налагодження (debugging) та профілювання коду. Уміння творчо застосовувати теоретичні знання (KST, IRT або подібні моделі у контексті системних задач) для вирішення нестандартних проблем оптимізації та забезпечення детермінізму програмних систем.

Шкала оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційований залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

7. Політика навчального курсу

Відвідування занять. З огляду на інтерактивний формат курсу, відвідування практичних занять є обов'язковим. Студенти, які не можуть регулярно відвідувати заняття, повинні протягом тижня узгодити з викладачем індивідуальний графік відпрацювання пропущених занять. Пропущені заняття відпрацьовуються на найближчій консультації протягом тижня після пропуску у формі співбесіди за питаннями з плану заняття. В окремих випадках дозволяється відпрацювання у письмовій формі шляхом виконання індивідуального письмового завдання.

Дотримання вимог академічної доброчесності здобувачами освіти під час вивчення навчальної дисципліни. Під час вивчення навчальної дисципліни здобувачі освіти мають дотримуватися загальноприйнятих морально-етичних норм і правил поведінки, вимог академічної доброчесності, передбачених Положенням про академічну доброчесність Національного аерокосмічного університету «Харківський авіаційний інститут» (<https://khai.edu/assets/files/polozhennya/polozhennya-pro-akademichnu-dobrochesnist.pdf>). Очікується,

що роботи здобувачів освіти будуть їх оригінальними дослідженнями або міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших здобувачів освіти становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі здобувача освіти є підставою для її незарахування викладачем незалежно від масштабів плагіату чи обману.

Вирішення конфліктів. Порядок і процедури врегулювання конфліктів, пов'язаних із корупційними діями, зіткненням інтересів, різними формами дискримінації, сексуальними домаганнями, міжособистісними стосунками та іншими ситуаціями, що можуть виникнути під час навчання, а також правила етичної поведінки регламентуються Кодексом етичної поведінки в Національному аерокосмічному університеті «Харківський авіаційний інститут» (<https://khai.edu.ua/university/normativna-baza/ustanovchi-dokumenti/kodeks-etichnoi-povedinki/>).

8. Методичне забезпечення

Електронний ресурс, на якому розміщено навчально-методичний комплекс дисципліни – <http://library.khai.edu/library/fulltexts/doc/1001Bazi.pdf>, який включає в себе:

- Силабус дисципліни;
- конспект лекцій, підручники (навчальні посібники), в тому числі в електронному вигляді, які за змістом повністю відповідають робочій програмі дисципліни;
- методичні вказівки та рекомендації для виконання курсових робіт та проєктів, розрахункових та розрахунково-графічних робіт, лабораторних та практичних робіт, а також рекомендації для самостійної підготовки;
- приклади розв'язування типових задач чи виконання типових завдань;
- питання, тести для контрольних заходів;
- каталоги інформаційних ресурсів.

Увесь науково методичний комплект з дисципліни розміщено на освітньому порталі: <https://study.korobchinskiy.com/course/view.php?id=11>

9. Рекомендована література

Базова

1. Системне програмування в ОС Linux Бочкарьов О.Ю. Навчальний посібник. — Магнолія 2006, 2025. — 229 с.
2. Роберт Лав. Системне програмування в Linux: Навчальний посібник. / Пер. з англ. — Х.: Фабула, 2013. — 448 с.
3. Майкл Керріск. Linux Programming Interface. Вичерпне керівництво з системного програмування в Linux та UNIX. — К.: Діалектика, 2019. — 1552 с.

4. Браян Керніган, Денніс Рітчі. Мова програмування C: Навчальний посібник. — К.: ТОВ "Вільямс", 2015. — 304 с..

5. Вільям Стівенс, Стівен Раго. UNIX. Професійне програмування: Навчальне видання. — К.: Символ-Плюс, 2016. — 944 с.

6. Коноваленко І.В. Програмування системних задач у середовищі Linux: Конспект лекцій. — Тернопіль: ТНТУ, 2018. — 140 с.

Допоміжна

1. Стівенс В. Р., Феннер Б., Рудофф Е. М. UNIX: мережеве програмування. API сокетів: Навчальне видання. — К.: Діалектика, 2018. — 1040 с.

2. Корбет Дж., Рубіні А., Кроа-Хартман Г. Драйвери пристроїв Linux: Практичний посібник. — К.: Символ-Плюс, 2016. — 752 с.

3. Роберт Сікорд. Безпечне програмування мовами C та C++: Навчальний посібник. — К.: ТОВ "Вільямс", 2017. — 496 с.

4. Девід Р. Бутенхоф. Багатопотокове програмування з використанням POSIX Threads. — К.: Питер, 2015. — 384 с.

5. Деніел Бовет, Марко Чезаті. Дослідження ядра Linux: Теоретичні основи та програмна реалізація. — К.: Символ-Плюс, 2014. — 1104 с.

10. Інформаційні ресурси

1. Коробчинський К. П. Системне програмування: Навчальний курс [Електронний ресурс] / К. П. Коробчинський / Портал ХАІ. – Режим доступу: <https://study.korobchinskiy.com/>

2. Коробчинський К. П. Матеріали блогу. [Електронний ресурс] / К. П. Коробчинський / Портал ХАІ. – Режим доступу: <https://korobchinskiy.com/>

3. <https://mentor.khai.edu> – Електронний курс Національного аерокосмічного університету ім. М.Є. Жуковського «ХАІ» курс “Системне програмування”

4. Сайт науково-технічної бібліотеки університету [Електронний ресурс]. – Режим доступу: <https://library.khai.edu>.