

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ
Декан факультету комп'ютерних
інформаційних технологій
Ігор ЯКИМЕНКО

“ 28 ” серпня 2025 р.

ЗАТВЕРДЖУЮ
Директор навчально-наукового
інституту новітніх освітніх технологій
Святослав ПІГЕЛЬ

“ 29 ” серпня 2025 р.

ЗАТВЕРДЖУЮ
Проректор з науково-
педагогічної роботи
Віктор ОСТРОВЕРХОВ

2025 р.

РОБОЧА ПРОГРАМА

з дисципліни «Системне програмування»

ступінь вищої освіти – бакалавр

галузь знань – 12 “Інформаційні технології”

спеціальність – 123 “Комп'ютерна інженерія”

освітньо-професійна програма – „Комп'ютерна інженерія”

Кафедра комп'ютерної інженерії

Форма навчання	Курс	Семест р	Лекції (год.)	Лабораторії (год.)	ІРС (год.)	Тренінг (год.)	Самост. робота студ. (год.)	Разом (год.)	Залік сем.)	Екз. (сем.)
Денна	2	3	30	30	4	8	78	150	-	3
Заочна	2	3	8	4	0	0	138	150	-	4

29.08.2025
[Signature]

Тернопіль – ЗУНУ
2025

Робоча програма складена на основі освітньо – професійної програми підготовки бакалавра галузі знань 12 “Інформаційні технології” спеціальності 123 “Комп’ютерна інженерія”, затвердженої Вченою радою ЗУНУ (протокол № 11 від 26 червня 2024 р.).

Робочу програму склав к.т.н., доцент кафедри КІ

Юрій БАТЬКО

Робоча програма затверджена на засіданні кафедри комп’ютерної інженерії, протокол №1 від 26 серпня 2025 р.

Завідувач кафедри

Леся ДУБЧАК

Гарант ОП

Леся ДУБЧАК

СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ "СИСТЕМНЕ ПРОГРАМУВАННЯ "

1. Опис дисципліни "Системне програмування"

Дисципліна- «Системне програмування»	Галузь знань, спеціальність, ступінь вищої освіти	Характеристика навчальної дисципліни
Кількість кредитів – <i>Денна – 5,</i> <i>Заочна – 5</i>	галузь знань – 12 „Інформаційні технології”	Статус дисципліни – обов’язкова Мова навчання – українська
Кількість залікових модулів: 5	Спеціальність – 123 „Комп’ютерна інженерія”	Рік підготовки: <i>Денна - 2,</i> <i>Заочна – 2</i> Семестр: <i>Денна – 3, Заочна – 3,4</i>
Кількість змістових модулів – 4	Ступінь вищої освіти - бакалавр	Лекції: <i>Денна - 30 год.,</i> <i>Заочна – 8 год.</i> Лабораторні заняття: <i>Денна - 30 год.</i> <i>Заочна – 4 год.</i>
Загальна кількість годин – <i>Денна – 150 год.,</i> <i>Заочна - 150 год.</i>		Самостійна робота: <i>Денна - 78 год.</i> <i>Заочна – 138 год.</i> Тренінг–8 год. Індивідуальна робота: <i>Денна -4 год.</i>
Тижневих годин: <i>Денна:</i> 3 семестр – 10 год., з них аудиторних - 4 год.		Вид підсумкового контролю <i>Денна:</i> 3 семестр – екзамен <i>Заочна:</i> 4 семестр – екзамен

2. Мета й завдання дисципліни "Системне програмування"

2.1. Мета вивчення дисципліни

Програма та тематичний план дисципліни „Системне програмування” орієнтовані на глибоке та ґрунтовне засвоєння студентами систематичних знань та практичних навичок для створення системних програмного забезпечення з ефективним використанням сучасних технологій.

Мета дисципліни "Системне програмування" - вивчення теорії, методів і засобів розробки та експлуатації системного програмного забезпечення обчислювальних систем.

2.2 Завдання вивчення дисципліни

Завданням вивчення дисципліни є вивчення науково-практичного інструментарію проектування системних програмних додатків та їх складових за допомогою сучасних засобів автоматизованого проектування.

В результаті вивчення дисципліни студенти мають вміти визначати алгоритми вирішення поставлених задач, проектувати загальну структуру програмних додатків, реалізовувати поставлені задачі за допомогою основних бібліотек та команд мови MASM, а також вивчення системних функції Win API.

2.3 Найменування та опис компетентностей, формування котрих забезпечує вивчення дисципліни "Системне програмування":

K12. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

K13. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

K19. Здатність системно адмініструвати, використовувати, адаптувати та експлуатувати наявні інформаційні технології та системи.

2.4 Передумови для вивчення дисципліни

Зазначена дисципліна включена до циклу дисциплін професійної підготовки за переліком програми. У структурно-логічній схемі навчання зазначена дисципліна розміщена на II-му курсі. Для засвоєння дисципліни студентам необхідні знання та навички, отримані при вивченні наступних дисциплін: "Основи програмування", "Прикладне програмне забезпечення для комп'ютерних систем і мереж", а також цілеспрямованої роботи на лекційних та лабораторних заняттях, самостійної роботи студентів.

2.5 Результати навчання.

В результаті вивчення курсу „Системне програмування ” студенти повинні:

ПРН1. Знати і розуміти наукові положення, що лежать в основі функціонування комп'ютерних засобів, систем та мереж.

ПРН2. Мати навички проведення експериментів, збирання даних та моделювання в комп'ютерних системах.

ПРН7. Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності.

ПРН10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання

ПРН16. Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.

3. Зміст дисципліни «Системне програмування»

Змістовий модуль 1. Концептуальні основи системного програмного забезпечення.

Тема 1. Основні поняття і визначення.

1. Складові обчислювальної системи. 2. Класифікація програмного забезпечення обчислювальних систем. 3. Предмет і задачі курсу "Системне програмне забезпечення". 4. Еволюція системного програмного забезпечення.

Література: 9, 11.

Тема 2. Операційні системи.

1. Призначення та склад операційних систем. 2. Класифікація операційних систем. 3. Режими функціонування операційних систем. 4. Режими обслуговування запитів користувачів. 5. Вимоги до операційних систем реального часу. 6. Принципи побудови операційних систем.

Література: 11,8.

Тема 3. Архітектурні особливості мікропроцесорів i80x86.

1. Реальний та захищений режими роботи мікропроцесора. 2. Нові системні регістри. 3. Підтримка сегментної схеми організації віртуальної пам'яті. 4. Підтримка сторінкової схеми. 5. Режим віртуальної машини. 6. Рівні привілеїв процесора. 7. Механізм шлюзів для передачі управління.

Література: 4, 2,1.

Тема 4. Концептуальні основи створення системного програмного забезпечення.

1. Визначення та структура процесу. 2. Стани процесу. 3. Властивості процесу. 4. Класифікація процесів. 5. Визначення та властивості ресурсів. 6. Класифікація ресурсів.

Література: 11,6.

Змістовий модуль 2. Створення та управління процесами в операційних системах.

Тема 5. Організація системи переривань процесора i80x86.

1. Система переривань в реальному режимі роботи процесора. 2. Типи переривань. 3. Каскадні переривання. 4. Процедура обробки переривання. 5. Особливості системи переривань в захищеному режимі роботи процесора. 6. Обробка переривань при виконанні однієї задачі. 7. Обробка переривань при перемиканні на нову задачу.

Література: 4, 2.

Тема 6. Планування та диспетчеризація процесів.

1. Поняття та стратегії планування. 2. Функції та способи реалізації планувальників. 3. Структура планувальника. 4. Класифікація дисциплін диспетчеризації. 5. Дисципліни обслуговування FCFS, SJN, SRT, RR. 6. Дисципліни диспетчеризації з витисненням та без витиснення. 7. Способи забезпечення гарантованого обслуговування процесів. 8. Критерії порівняння алгоритмів диспетчеризації. 9. Вплив планування на продуктивність обчислювальної системи. 10. Використання динамічних пріоритетів.

Література: 2.

Тема 7. Синхронізація процесів.

1. Незалежні та взаємодіючі обчислювальні процеси. 2. Види задач синхронізації паралельних процесів. 3. Синхронізація за допомогою блокування пам'яті. Алгоритм Деккера. Команда "Перевірка та встановлення". 4. Використання семафорів для синхронізації та впорядкування паралельних процесів. 5. Монітороподібні засоби синхронізації паралельних процесів. 6. Поштові ящики. 7. Конвеєри. 8. Черги повідомлень. 9. Взаємоблокування.

Література: 2, 7.

Змістовий модуль 3. Керування ресурсами та пам'яттю операційних систем.

Тема 8. Керування реальною пам'яттю.

1. Підходи до керування реальною пам'яттю. 2. Неперервний розподіл оперативної пам'яті. 3. Розподіл з перекриттям. 4. Статичний розподіл пам'яті. 5. Динамічний розподіл пам'яті. 6. Розділи пам'яті з фіксованими розмірами. 7. Розділи пам'яті із змінними розмірами.

Література: 7, 8.

Тема 9. Керування віртуальною пам'яттю.

1. Структура, основні поняття та принципи віртуалізації пам'яті. 2. Сторінкова схема організації віртуальної пам'яті. 3. Сегментна схема організації віртуальної пам'яті. 4. Сегментно-сторінкова схема організації віртуальної пам'яті.

Література: 11, 2, 4.

Тема 10. Інтерфейси операційних систем та API функції.

1. Основні поняття і визначення. 2. Інтерфейс прикладного програмування API. 3. Реалізація функцій API на рівні операційної системи. 4. Реалізація функцій

API на рівні системи програмування. 5. Реалізація функцій API за допомогою зовнішніх бібліотек. POSIX-інтерфейс.

Література: 11, 7, 3.

Змістовий модуль 4. Мова програмування Асемблер.

Тема 11. Мова програмування Асемблер.

1. Основні поняття. 2. Структура програми на мові Асемблер та коментарі. 3. Поняття типів даних. 4. Типи адресації. 5. Команди пересилки даних. 6. Команда пересилки даних з умовою. 7. Команди отримання вмісту прапорів. 8. Команди обміну значеннями. 9. Асемблерні вставки на у код на мові C++.

Література: 8, 3.

Тема 12. Математичні операції на мові Асемблер.

1. Види математичних операцій. 2. Оператор додавання. 3. Оператор віднімання. 4. Множення та ділення цілочисельних чисел. 5. Модифікація прапорів під час математичних операцій. 6. Команди роботи зі стеком.

Література: 2, 3.

Тема 13. Умовні операції на мові Асемблер.

1. Логічні операції на мові Асемблер. 2. Команда AND. 3. Команда OR. 4. Команда NOT. 5. Команда TEST. 6. Організація безумовного переходу. 7. Організація умовного переходу.

Література: 1, 3.

Тема 14. Циклічні операції на мові Асемблер та робота з масивами.

1. Загальні принципи організації циклів на мові Асемблер. 2. Цикли з передумовою. 3. Цикли з післяумовою. 4. Цикли з лічильником. 5. Організація масивів на мові Асемблер. 6. Способи доступу до елементів масиву. 7. Заповнення масиву. 8. Пошук значення в масиві. 9. Сортування масиву

Література: 9, 3.

4. Структура залікового кредиту дисципліни «Системне програмування»

(денна форма навчання)

	Кількість годин					Контрольні заходи
	Лекції	Лабораторні заняття	Самостій на робота	Індивідуальна робота	Трейнінг	
Змістовий модуль 1. Концептуальні основи системного програмного забезпечення.						
Тема 1. Основні поняття і визначення	2	-	4	1	2	Опитування
Тема 2. Операційні системи	2	4	4			Опитування
Тема 3. Архітектурні особливості мікропроцесорів i80x86	2	-	6			Опитування
Тема 4. Концептуальні основи створення	2		6			Опитування

системного програмного забезпечення.						
Змістовий модуль 2. Створення та управління процесами в операційних системах						
Тема 5. Організація системи переривань процесора i80x86.	2	4	4	1	2	Опитування
Тема 6. Планування та диспетчеризація процесів	2	-	6			Опитування
Тема 7. Синхронізація процесів	2	4	6			Опитування
Змістовий модуль 3. Керування ресурсами та пам'яттю операційних систем.						
Тема 8. Керування реальною пам'яттю	2	-	6	1	2	Опитування
Тема 9. Керування віртуальною пам'яттю	2		6			Опитування
Тема 10. Інтерфейси операційних систем та API функції.	4	6	6			Опитування
Змістовий модуль 4. Мова програмування Асемблер						
Тема 11. Мова програмування Асемблер	2		6	1	2	Опитування
Тема 12. Математичні операції на мові Асемблер.	2	6	6			Опитування
Тема 13. Умовні операції на мові Асемблер	2	-	6			Опитування
Тема 14. Циклічні операції на мові Асемблер та робота з масивами	2	6	6			Опитування
Разом	30	30	78	4	8	

(заочна форма навчання)

	Кількість годин		
	Лекції	Лабораторні заняття	Самостійна робота
Змістовий модуль 1. Концептуальні основи системного програмного забезпечення.			
Тема 1. Основні поняття і визначення	2	-	8
Тема 2. Операційні системи	-	-	10
Тема 3. Архітектурні особливості мікропроцесорів i80x86	-	-	10
Тема 4. Концептуальні основи створення системного програмного забезпечення.	-	2	10
Змістовий модуль 2. Створення та управління процесами в операційних системах.			
Тема 5. Організація системи переривань процесора i80x86.	-	-	10
Тема 6. Планування та диспетчеризація процесів	-	-	10
Тема 7. Синхронізація процесів	-	-	10
Змістовий модуль 3. Керування ресурсами та пам'яттю операційних систем.			
Тема 8. Керування реальною пам'яттю	2	-	10
Тема 9. Керування віртуальною пам'яттю	-	-	10
Тема 10. Інтерфейси операційних систем та API функції.			10

Змістовий модуль 4. Мережеве програмування на основі Java			
Тема 11. Мова програмування Асемблер	2	-	10
Тема 12. Математичні операції на мові Асемблер.	-	2	10
Тема 13. Умовні операції на мові Асемблер	-	-	10
Тема 14. Циклічні операції на мові Асемблер та робота з масивами	2	-	10
Разом	8	4	138

5. Тематика лабораторних занять

Лабораторна робота № 1.

Тема: Створення віконного додатку засобами функцій Win API.

Мета: Ознайомитися з структурою та принципами функціонування віконних програмних додатків в ОС Windows створених за допомогою функцій Win API.

Питання для обговорення:

1. Структура програмного додатку в ОС Windows.
2. Поняття точки входу та передачі параметрів.
3. Створення графічного вікна програмного додатку.

Література: 2

Лабораторна робота № 2.

Тема: Створення та обробка повідомлень в ОС Windows.

Мета: Ознайомитися з структурою та принципами створення, перехоплення та обробки повідомлень в ОС Windows.

Питання для обговорення:

1. Поняття «повідомлення».
2. Створення та опрацювання повідомлень.
3. Організація взаємодії програми та користувача.

Література: 1.

Лабораторна робота № 3.

Тема: Процеси та потоки в ОС Windows.

Мета: Ознайомитися з структурою та принципами роботи багатопоточних програмних додатків в ОС Windows.

Питання для обговорення:

1. Поняття процесу.
2. Поняття потоку.
3. Організація розпаралелювання роботи програми.

Література: 1.

Лабораторна робота № 4

Тема: Функції WinAPI отримання та встановлення системних характеристик ОС

Windows.

Мета: Отримати практичні навички по використанню WinAPI функцій для роботи з системними параметрами в ОС Windows.

Питання для обговорення:

1. Поняття системних параметрів.
2. API функції для перегляду системних параметрів.
3. API функції для встановлення системних параметрів.

Література: 2.

Лабораторна робота № 5.

Тема: Лінійні алгоритми та алгоритми розгалуження з використанням вставок на мові Assembler.

Мета: Отримати практичні навички по реалізації лінійних алгоритмів та алгоритмів розгалуження з використанням вставок на мові Assembler.

Питання для обговорення:

1. Структура програми на мові асемблер.
2. Команди обміном інформацією.
3. Команди умовних та безумовних переходів.

Література: 2.

Лабораторна робота № 6.

Тема: Циклічні алгоритми алгоритми та операції над масивами з використанням вставок на мові Assembler.

Мета: Отримати практичні навички по реалізації циклічних алгоритмів та операції над масивами з використанням вставок на мові Assembler.

Питання для обговорення:

1. Команди організації циклів.
2. Основні принципи адресації.
3. Основні алгоритми роботи з масивами.

Література: 1,2,6,3

6. Організація і проведення тренінгу з дисципліни «Системне програмування»

№п/п	Вид роботи	Порядок проведення тренінгу
1	Синхронізація між процесами «читач-письменник»	1. Способи забезпечення гарантованого обслуговування процесів. 2. Апаратна підтримка синхронізації. 3. Написати алгоритмічний код вирішення задачі синхронізації між процесами «читач-письменник» з допомогою семафорів (процес «письменник» при поступленні у чергу зупиняє всі «читачів»). 4. Визначити безпечний чи небезпечний стан системи щодо

		використання одного ресурсу та блокування виконання процесів
2	Задача синхронізації між процесами «взаємного виключення»	1.Критерії порівняння алгоритмів диспетчеризації. 2.Монітороподібні засоби синхронізації паралельних процесів. 3. Написати алгоритмічний код вирішення задачі синхронізації між процесами «взаємного виключення» з допомогою семафорів. 4. Визначити мінімальне значення ресурсу X для забезпечення «безпечного стану» системи
3	Задача синхронізації «виробник-споживач»	1.Вплив планування на ефективність обчислювальних систем. 2.Поштові ящики 3. Написати алгоритмічний код вирішення задачі синхронізації «виробник-споживач» між процесами з допомогою семафорів. 4. Визначити безпечний чи небезпечний стан системи щодо використання декількох ресурсів та блокування виконання процесів
4	Задача синхронізації між процесами «читач-письменник»	1.Використання динамічних пріоритетів. 2.Черги повідомлень, конвеєри. 3. Написати алгоритмічний код вирішення задачі синхронізації між процесами «читач-письменник» з допомогою семафорів (процес «письменник» очікує зчитування всіма «читачами») 4. Визначити безпечний чи небезпечний стан системи щодо використання одного ресурсу та блокування виконання процесів

7. Самостійна робота студентів (підготовка наскрізного проєкту)

(денна форма навчання)

№ п/п	Тематика	Завдання
1	Дослідження методів та технологій створення системних програм	Аналіз предметної області
2	Програмні засоби проектування та реалізації системних програм	
3	Особливості мов програмування низького рівня	
4	Проектування структури системного програмного додатку	Розробка програмного коду
5	Мови програмування для реалізації системних програм	
6	Реалізація програмного коду системного програмного додатку	
7	Аналіз апаратних ресурсів системи обробки даних	Оптимізація програмного коду
8	Оптимізація програмного коду	
9	Тестування програмного додатку	Тестування та порівняння розробленого програмного додатку
10	Порівняльний аналіз з програмами аналогами	

(заочна форма навчання)

№ п/п	Тематика
1	Операційна система і які її основні компоненти
2	Основні функції та обов'язки ядра операційної системи.
3	Менеджер пам'яті в операційній системі та основні методи управління пам'яттю
4	Багатозадачність та основні принципи її реалізації
5	Основні типи процесів і потоків (threads), їх основні відмінності
6	Синхронізація потоків та механізми синхронізації
7	Принципи роботи драйверів пристроїв та їх взаємодія з операційною системою.
8	Інтерфейси прикладного програмування (API) та роль в системному програмуванні.
9	Основні техніки оптимізації виконання програм на рівні операційної системи та апаратного забезпечення
10	Роль і функції планувальників задач (schedulers) в операційних системах
11	Особливості системного програмування на рівні ядра ОС
12	Основні підходи до абстракції апаратного забезпечення в сучасних операційних системах
13	Концепція віртуалізації пам'яті та її вплив на ефективність виконання багатозадачних програм
14	Використання механізмів міжпроцесової взаємодії для реалізації системи процесів з <i>"поштовими скриньками"</i> .
15	Концепція і принципи реалізації модульності в системному програмуванні.
16	Методи керування доступом до файлових систем в сучасних операційних системах.

8. Засоби оцінювання та методи демонстрування результатів навчання

У навчальному процесі застосовуються: лекції, в тому числі з використанням мультимедіапроектора та інших ТЗН; практичні заняття; індивідуальні заняття, самостійна робота студента, робота в Інтернет.

У процесі вивчення дисципліни "Системне програмування" використовуються наступні засоби оцінювання та методи демонстрування результатів навчання:

- поточні опитування;
- модульне тестування;
- презентації результатів виконання завдань та досліджень;
- оцінювання результатів виконання тренінгів;
- оцінювання результатів самостійної роботи;
- екзамен.

9. Критерії, форми поточного та підсумкового контролю

Підсумковий бал (за 100-бальною шкалою) з дисципліни «Системне програмування» визначається як середньозважена величина, в залежності від питомої ваги кожної складової залікового кредиту:

3 семестр

Модуль 1	Модуль 2	Модуль 3	Модуль 4	Модуль 5
----------	----------	----------	----------	----------

10%	10%	10%	10%	5%	15%	40%
Поточне оцінювання	Модульний контроль 1	Поточне оцінювання	Модульний контроль 2	Тренінги	Самостійна робота	Екзамен
Середнє арифметичне за 4 лабораторні роботи	Тестові завдання	Середнє арифметичне за 2 лабораторні роботи	Тестові завдання	Виконання 4 завдань	Виконання наскрізного проекту із 4 завдань	2 теоретичні питання по 25 балів = 50 балів, Задача = 50 балів

Оцінка за «Поточне оцінювання» визначається як середнє арифметичне оцінок, отриманих студентом при виконанні лабораторних завдань. Кожне завдання оцінюється за 100 бальною шкалою на основі якості виконання та відповідності поставленим завданням. Бали нараховуються відповідно до таких критеріїв:

- 15 балів – розуміння теоретичних відомостей (розкриття понятійного апарату відповідно до теми лабораторного заняття);
- 25 балів – правильність розв’язування (коректність реалізації, правильний вибір підходу);
- 10 балів – якість виконання (унікальність рішення, стиль програмування, якість програмного коду);
- 10 балів – результативність / тестування (програма видає коректний результат при тестуванні);
- 10 балів – аналіз і висновки (опис отриманих результатів, засоби їх досягнення);
- 30 балів – захист практичної роботи (відповідь на запитання викладача).

Модульний контроль проводиться у формі виконання тестових завдань з однією правильною відповіддю. Тестовий контроль складається з теоретичних питань і одного практичного завдання, що передбачає проведення практичних дій перед вибором варіанту відповіді. Частка виконання практичного завдання становить 40% (40 балів при правильній відповіді) у підсумку. Відповідно частка теоретичної частини складає 60% (60 балів). Максимальний бал за виконання теоретичних питань модульного контролю залежить від середнього арифметичного правильних відповідей (наприклад,

якщо тестів 20 і студент відповів на 15 правильно, то результат рівний $15/20 = 0.75$, тобто 75 % від максимально можливого балу). Підсумкове значення виконання тестового контролю визначається сумою балів отриманих при виконанні тестових і практичних завдань.

Тренінг оцінюється з повнотою реалізації, якістю програмного коду, тестуванням та аналізом ефективності виконання завдань відповідно до п.7 даної робочої програми. Оцінювання рівня виконання завдань тренінгу проводиться викладачем за 100-бальною шкалою згідно таблиці 1.

Таблиця 1 – Критерії оцінювання тренінгу

Завдання	Макс. бал	Високий рівень	Достатній рівень	Середній рівень	Низький рівень
1. Синхронізація між процесами «читач-письменник»	20	16-20 ґрунтовний огляд з наведеним порівняльним аналізом	11-15 достатньо аргументований, але без огляду аналогів	6-10 поверхневий огляд	0-5 відсутній або невідповідне
2. Задача синхронізації між процесами «взаємного виключення»	20	16-20 аргументоване, з порівняльним аналізом	11-15 достатньо аргументоване, але без альтернатив	6-10 поверхневий аналіз	0-5 відсутнє або невідповідне
3. Задача синхронізації «виробник-споживач»	30	21-30 реалізація у повній мірі	11-20 реалізація з незначними недоліками	6-10 передбачено лише часткові випадки	0-5 відсутнє або невідповідне
4. Задача синхронізації між процесами «читач-письменник»	30	21-30 комплексне тестування	11-20 тестування, часткове	6-10 часткова відповідність, наявність помилок	0-5 відсутнє або невідповідне
Разом	100	-	-	-	-

Модуль «Самостійна робота» оцінюється викладачем за якістю виконаного індивідуального завдання та звіту відповідно до критеріїв, визначених у розділі 6 «Самостійна робота» цієї робочої програми. Підсумковий бал розраховується як сума

оцінок за окремими критеріями у таблиці 2.

Таблиця 2 – Критерії оцінювання самостійної роботи

Завдання	Макс. бал	Високий рівень	Достатній рівень	Середній рівень	Низький рівень
Проектування структури та алгоритму синхронізації роботи процесів для ОС Windows.	30	22-30 алгоритм реалізовано у повній мірі	15-21 алгоритм реалізований в неповній мірі	10-14 поверховий аналіз	0-9 відсутнє або невідповідне
Розробка модулів для реалізації програмного додатку на мові програмування C++ з включеннями елементів на мові низького рівня	30	22-30 універсальний підхід	15-21 загальний, не універсальний підхід	10-14 типові рішення	0-9 відсутнє або невідповідне
Програмна реалізація системного програмного додатку та проведення його комплексного тестування	40	26-40 комплексне тестування	17-25 тестування, часткове	10-14 часткова відповідність, наявність помилок	0-9 відсутнє або невідповідне
Разом	100	-	-	-	-

Модуль «Екзамен» оцінюється викладачем за 100-бальною шкалою та включає два теоретичних запитання по 20 балів та одну задачу – 60 балів. Якщо студент надає вичерпну відповідь на теоретичне запитання, наводить приклади, надає аргументовані відповіді на додаткові запитання, то отримує 20 балів. Якщо відповідь містить несуттєві неточності, приклади малоінформативні або загального виду, відповіді на додаткові запитання є неповними, то отримує 15 балів. Якщо відповідь містить несуттєві неточності, приклади малоінформативні або загального виду, не було надано відповіді на додаткові запитання, то отримує 10 балів. Якщо відповідь містить неточності, не наведені приклади, на додаткові питання дано часткові відповіді, то студент отримує 5 балів. Якщо студент розв'язав задачу, обґрунтував вибір обраного підходу, здійснив тестування, провів детальний аналіз отриманих результатів, то отримує 60 балів. Коректно розв'язана задача без наведеного аргументації вибору алгоритму роботи та здійснених висновків – 40 балів. Коректно розв'язана задача без аргументації вибору засобів розробки та алгоритму, без проведеного комплексу

тестування – 30 балів. Частковий розв’язок поставленої задачі – 20 балів. Частковий розв’язок поставленої задачі із неточностями – 10 балів.

Отримана оцінка за національною шкалою ECTS виставляється відповідно до шкали оцінювання рівня знань.

Шкала оцінювання

За шкалою університету	За національною шкалою	За шкалою ECTS
90-100	Відмінно	A (відмінно)
85-89	Добре	B (дуже добре)
75-84		C (добре)
65-74		D (задовільно)
60-64	Задовільно	E (достатньо)
35-59		FX (незадовільно, з можливістю повторного складання)
1-34	Незадовільно	F (незадовільно, з обов’язковим повторним курсом)

11. Інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна

№	Найменування	Номер теми
1.	MS Visual Studio Community	1-10
2.	DEV C++	1-10
3.	ASM Visual	11-14

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Alexander Stepanov and Paul McJones Elements of Programming Semigroup Press, 2019, 218
2. Kenneth Leroy Busbee, Dave Braunschweig. Programming Fundamentals - A Modular Structured Approach, 2nd Edition Houston Community College; eBook (Creative Commons Licensed)
3. Changkun Ou Modern C++ Tutorial: C++11/14/17/20 On the Fly - The Fastest Guide towards Modern C++ GitHub; eBook (Creative Commons Licensed), 92.
4. Johan Sannemo Principles of Algorithmic Problem Solving KTH Royal Institute of Technology, 2018, 351.
5. Divakar Viswanath Scientific Programming and Computer Architecture. The MIT Press, 2017, 434.
6. Frank B. Brokken C++ Annotations: An Extensive Tutorial about the C++ Programming Language. University of Groningen, 2021.
7. Bjarne Stroustrup, Herb Sutter C++ Core Guidelines Standard C++ Foundation and GitHub. 2022.
8. C++ Notes for Professionals. Stack Overflow Contributors. Goalkicker. 2018, 708.
9. Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, Shriram Krishnamurthi How to Design Programs, second edition: An Introduction to Programming and Computing (The MIT Press) second edition 2018, 792.
10. Dr. Rian Quinn. Hands-On System Programming with C++: Build Performant and Concurrent Unix and Linux Systems with C++17. Packt Publishing 2018, 522.

11. Florian Jatón The Constitution of Algorithms: Ground-Truthing, Programming, Formulating. The MIT Press, 2021, 154.
12. Xinyu Liu. Elementary Algorithms. GitHub.com, 2021, 503.
13. Granville Barnett, Luca Del Tongo, John Bullinaria. Data Structures and Algorithms: Annotated Reference with Examples. University of Birmingham, 2019, 112.
14. Зайцев, В. Г. Операційні системи. / Навчальний посібник для студентів спеціальності 123 «Комп'ютерна інженерія» / В. Г. Зайцев, І. П. Дробязко. – Київ: КПІ ім. Ігоря Сікорського, 2019. – 240 с.
15. Федотова-Півень І.М. Операційні системи: навчальний посібник. [за ред. В.М. Рудницького] / І.М. Федотова-Півень, І.В. Миронець, О.Б. Півень, С.В. Сисоєнко, Т.В. Миронюк; Черкаський державний технологічний університет. – Харків: ТОВ «ДІСА ПЛЮС», 2019. – 216 с.