

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

ЗАТВЕРДЖУЮ

В. о. декана факультету комп'ютерних
інформаційних технологій

Ігор ЯКИМЕНКО

“ ” 2023 р.

ЗАТВЕРДЖУЮ

В. о. проректора з науково-
педагогічної роботи

Віктор ОСТРОВЕРХОВ

2023 р.

ЗАТВЕРДЖУЮ

Директор навчально-наукового
інституту новітніх освітніх технологій

Святослав ПИТЕЛЬ

“ ” 2023 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	Семестр	Лекції (год.)	Лабораторні (год.)	ІРС (год.)	Тренінг, КПЗ (год.)	Самост. робота студ. (год.)	Разом (год.)	Залік сем.)	Екз. (сем.)
Денна	2	3	28	28	3	8	83	150	-	3
Заочна	2	3	8	4	0	0	138	150	-	4

31.08.2023/

Тернопіль – ЗУНУ
2023

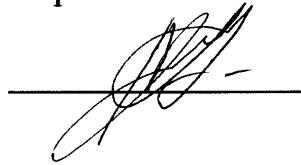
Робоча програма складена на основі освітньо – професійної програми підготовки бакалавра галузі знань 12 “Інформаційні технології” спеціальності 123 “Комп’ютерна інженерія”, затвердженої Вченою радою ЗУНУ (протокол № 9 від 15 червня 2022 р.).

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

Юрій БАТЬКО

Робоча програма затверджена на засіданні кафедри комп’ютерної інженерії, протокол №1 від 28 серпня 2023р.

Завідувач кафедри



Леся ДУБЧАК

Розглянуто та схвалено групою забезпечення спеціальності «Комп’ютерна інженерія», протокол №1 від 28 серпня 2023 р.

Голова ГЗС



Олег БЕРЕЗЬКИЙ

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 – 5	галузь знань – 12 „Інформаційні технології”	Статус дисципліни – нормативна Мова навчання – українська
Кількість залікових модулів: 4	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 – 3	Ступінь вищої освіти - бакалавр	Лекції: <i>Денна - 28 год.,</i> <i>Заочна – 8 год.</i> Лабораторні заняття: <i>Денна - 28 год.</i> <i>Заочна – 4 год.</i>
Загальна кількість годин – <i>Денна – 150 год.,</i> <i>Заочна - 150 год.</i>		Самостійна робота: <i>Денна - 83 год.</i> <i>Заочна – 138 год.</i> Тренінг,КПЗ – 10 год. Індивідуальна робота: <i>Денна -3 год.</i>
Тижневих годин: <i>Денна:</i> 3 семестр – 10,7 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. 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. 5. Обробка переривань при виконанні однієї задачі. 6. Обробка переривань при перемиканні на нову задачу.

Література: 4, 2.

Тема 5. 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. Керування ресурсами та пам'яттю операційних систем.

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.

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 3. 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 програмне забезпечення»

(денна форма навчання)

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

Тема 5. Концептуальні основи створення системного програмного забезпечення.	2	4	5			Опитування
Змістовий модуль 2. Керування ресурсами та пам'яттю операційних систем.						
Тема 6. Планування та диспетчеризація процесів	2	-	6	-	4	Опитування
Тема 7. Синхронізація процесів	2	4	6	-		Опитування
Тема 8. Керування реальною пам'яттю	2	-	6	-		Опитування
Тема 9. Керування віртуальною пам'яттю	2		6	-		Опитування
Тема 10. Інтерфейси операційних систем та API функції.	2	4	6	-		Опитування
Змістовий модуль 3. Архітектури операційних систем.						
Тема 11. Мова програмування Асемблер	2		6		2	Опитування
Тема 12. Математичні операції на мові Асемблер.	2	4	6	1		Опитування
Тема 13. Умовні операції на мові Асемблер	2	-	6			Опитування
Тема 14. Циклічні операції на мові Асемблер та робота з масивами	2	4	6	1		Опитування
Разом	28	28	83	3	8	

(заочна форма навчання)

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

Змістовий модуль 3. Архітектури операційних систем.					
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		-	-	-	10
Разом	8	4	0	0	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. Комплексне практичне індивідуальне завдання

Комплексні практичні індивідуальні завдання (КПІЗ) виконуються самостійно кожним студентом згідно з варіантом, отриманим у викладача. Варіанти КПІЗ з дисципліни «Системне програмування»:

1. Породження процесів та потоків у ОС Windows.
2. Породження обчислювальних процесів та потоків в ОС Linux.
3. Взаємодія між процесами і потоками та синхронізація процесів і

потоків у середовищі ОС Windows.

4. Організація взаємодії процесів за допомогою каналів в ОС Linux.
5. Створення динамічних бібліотек DLL.
6. Керування процесами і потоками у середовищі ОС Windows.
7. Робота з реєстром ОС Windows засобами Win API.
8. Організація вводу / виводу. Переривання. Сигнали надійні і ненадійні в ОС Linux.
9. Кросплатформенне СПЗ.
10. Обробка помилок в ОС Windows.
11. Кодування Unicode та ANSI.
12. Об'єкти ядра. Облік користувачів об'єктів ядра. Захист об'єктів ядра.
13. Визначення обмежень, що накладаються на процеси в завданнях.
14. Потоки в ОС Windows.
15. Бібліотечні функції мови C/C++ для роботи з потоками.
16. Припинення і відновлення потоків і процесів.
17. Асинхронний виклик функцій.
18. Потоки і робота з ними.
19. Особливості проектування файлів на різних платформах.
20. Спільний доступ процесів до даних через механізм проектування.
21. Стандартна купа (heap) процесу.
22. Основи Dll, Dll і адресний простір процесу.
23. Функція входу/виходу.
24. Відкладене завантаження Dll.
25. Функції Dll Main і бібліотека мови C/C++.
26. Модифікація базових адрес EXE і DLL модулів.
27. Динамічна локальна пам'ять.
28. Статична локальна пам'ять.

7. Самостійна робота студентів

(денна форма навчання)

№ п/п	Тематика
1	Використання механізму системних викликів ОС Linux для програмного розмонтовування підключених файлових систем.
2	Розробка програми-скрипта для виводу інформації про зареєстрованих в системі користувачів, а також груп користувачів.
3	Розробка системної утиліти для створення каталогів в середовищі ОС Linux із гнучким інтерфейсом через командний рядок.
4	Розробка системної утиліти для копіювання файлів в середовищі ОС Linux із гнучким інтерфейсом через командний рядок.
5	Розробка системної утиліти для переміщення файлів та каталогів в середовищі ОС Linux із гнучким інтерфейсом через командний рядок.
6	Розробка системної утиліти для видалення файлів та каталогів в середовищі ОС Linux із гнучким інтерфейсом через командний рядок.
7	Розробка системної утиліти для зміни власників файлів та каталогів в середовищі ОС Linux із гнучким інтерфейсом через командний рядок.

8	Розробка системної утиліти для зміни прав доступу до файлів та каталогів в середовищі ОС Linux із гнучким інтерфейсом через командний рядок.
9	Розробка системної утиліти для створення зсилок на файли та каталоги в середовищі ОС Linux із гнучким інтерфейсом через командний рядок.
10	Розробка системної утиліти для виводу вмісту каталогів, а також інформації про файли (права доступу, власник, група, розмір, дата створення) в середовищі ОС Linux із гнучким інтерфейсом через командний рядок.

(заочна форма навчання)

№ п/п	Тематика
1	Використання механізму системних викликів ОС Linux для програмного розмонтування підключених файлових систем.
2	Розробка програми-скрипта для виводу інформації про зареєстрованих в системі користувачів, а також груп користувачів.
3	Розробка системної утиліти для створення каталогів в середовищі ОС Linux із гнучким інтерфейсом через командний рядок.
4	Розробка системної утиліти для копіювання файлів в середовищі ОС Linux із гнучким інтерфейсом через командний рядок.
5	Розробка системної утиліти для переміщення файлів та каталогів в середовищі ОС Linux із гнучким інтерфейсом через командний рядок.
6	Розробка системної утиліти для видалення файлів та каталогів в середовищі ОС Linux із гнучким інтерфейсом через командний рядок.
7	Розробка системної утиліти для зміни власників файлів та каталогів в середовищі ОС Linux із гнучким інтерфейсом через командний рядок.
8	Розробка системної утиліти для зміни прав доступу до файлів та каталогів в середовищі ОС Linux із гнучким інтерфейсом через командний рядок.
9	Розробка системної утиліти для створення зсилок на файли та каталоги в середовищі ОС Linux із гнучким інтерфейсом через командний рядок.
10	Розробка системної утиліти для виводу вмісту каталогів, а також інформації про файли (права доступу, власник, група, розмір, дата створення) в середовищі ОС Linux із гнучким інтерфейсом через командний рядок.
11	Використання механізму системних викликів ОС Linux, для відслідковування процесу виконання системних програм.
12	Використання механізму системних викликів ОС Linux для розробки інтерпретатора команд типу <i>shell</i> .
13	Використання механізму системних викликів ОС Linux для планування виконання задач.
14	Використання механізмів міжпроцесової взаємодії в ОС Linux для реалізації системи процесів з " <i>поштовими скриньками</i> ".
15	Розробка програми-скрипта для аналізу виводу команди " <i>dmesg</i> " і пошуку відповідного місця у вихідних кодах ядра ОС Linux.

8 Організація і проведення тренінгу з дисципліни «Системне програмування»

№п/п	Вид роботи	Порядок проведення тренінгу
1	Синхронізація між процесами «читаць-письменник»	1. Способи забезпечення гарантованого обслуговування процесів. 2. Апаратна підтримка синхронізації.

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

9. 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;
- залікове модульне тестування та опитування;
- презентації результатів виконання завдань та досліджень;
- оцінювання результатів КППЗ;
- завдання на лабораторному обладнанні;
- ректорська контрольна робота;
- екзамен.

10. Критерії, форми поточного та підсумкового контролю

В процесі вивчення дисципліни «Системне програмування» використовуються наступні методи оцінювання навчальної роботи студента:

- поточне тестування та опитування;
- підсумкове тестування по кожному змістовому модулю;
- ректорська контрольна робота;
- підсумкова оцінка за комплексне практичне індивідуальне завдання (КПІЗ), враховуючи поточне опитування;
- підсумковий письмовий екзамен.

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

Заліковий модуль 1	Заліковий модуль 2	Заліковий модуль 3 (підсумкова оцінка за КПІЗ)	Заліковий модуль 4 (екзамен)	Разом
20 %	20 %	20 %	40 %	100%
1. Усне опитування під час заняття (5 тем по 5 бали = 25 балів) 2. Письмова робота = 75 балів	1. Усне опитування під час заняття (9 тем по 5 бали = 45 балів) 2. Письмова робота = 55 балів	1. Написання та захист КПІЗ = 80 балів. 2. Виконання завдань під час тренінгу = 20 балів	1. Тестові завдання (25 тестів по 2 бали за тест) – макс. 50 балів 2. Завдання. 1 – макс. 25 балів 3. Завдання. 2 – макс. 25 балів	100

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	Задовільно	D (задовільно)
60-64		E (достатньо)
35-59	Незадовільно	FX (незадовільно, з можливістю повторного складання)
1-34		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. Roldan Pozo, Ronald Kriemann, et al. C++ Programming for Scientists NIST, Ohio Supercomputer Center
11. Dr. Rian Quinn. Hands-On System Programming with C++: Build Performant and Concurrent Unix and Linux Systems with C++17. Packt Publishing 2018, 522.
12. Florian Jatton The Constitution of Algorithms: Ground-Truthing, Programming, Formulating. The MIT Press, 2021, 154.
13. Xinyu Liu. Elementary Algorithms. GitHub.com, 2021, 503.
14. Granville Barnett, Luca Del Tongo, John Bullinaria. Data Structures and Algorithms: Annotated Reference with Examples. University of Birmingham, 2019, 112.