

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Кафедра ___ комп'ютерних систем, мереж та кібербезпеки _____

“ЗАТВЕРДЖУЮ”

Декан факультету інформаційних технологій



проф. О.Г. Глазунова
_____ 2023 р.

СХВАЛЕНО

на засіданні кафедри
комп'ютерних систем, мереж та кібербезпеки

Протокол № 10 від «17» травня 2023 р.

Касаткін Д.Ю. Завідувач кафедри
(доц. Касаткін Д.Ю.)

РОЗГЛЯНУТО

Гарант ОП

«Кібербезпека»

Гарант ОП

(проф. Лахно В.А.)

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
“Системне програмування”**

спеціальність _____ 125 – «Кібербезпека» _____

освітня програма _____ «Кібербезпека» _____

Факультет (ННІ) _____ інформаційних технологій _____

Розробники: _____ доцент, к.т.н., доцент Шкарупило В.В. _____

Київ – 2023 р.

1. Опис навчальної дисципліни

Системне програмування

(назва)

Галузь знань, спеціальність, освітня програма, освітній ступінь		
Галузь знань	Інформаційні технології	
Спеціальність	125 – «Кібербезпека»	
другий (магістерський) рівень	Бакалавр	
Характеристика навчальної дисципліни		
Вид	Обов'язкова	
Загальна кількість годин	150	
Кількість кредитів ECTS	5	
Кількість змістових модулів	2	
Курсовий проект (робота) (якщо є в робочому навчальному плані)	-	
Форма контролю	Іспит	
Показники навчальної дисципліни для денної та заочної форм навчання		
	денна форма навчання	заочна форма навчання
Рік підготовки (курс)	2023-2024 (3)	-
Семестр	6	-
Лекційні заняття	45 год.	-
Практичні, семінарські заняття	-	-
Лабораторні заняття	45 год.	-
Самостійна робота	60 год.	-
Індивідуальні завдання	-	-
Кількість тижневих аудиторних годин для денної форми навчання	6 год.	-

2. Мета, завдання та компетентності навчальної дисципліни.

Мета – набуття здобувачами знань та вмінь створення і застосування системних програм на основі низькорівневих і високорівневих засобів; вмінь створювати багатомодульні програми; використовувати контейнери даних, засоби низькорівневого та високорівневого введення/виведення; опрацьовувати переривання, виключні ситуації, помилки; визначати конфігурацію обладнання; працювати з відеопідсистемою; створювати і застосовувати динамічно приєднані бібліотеки DLL; застосовувати засоби інтерфейсу Win32 API.

Завдання навчальної дисципліни «Системне програмування» – теоретична та практична підготовка здобувачів до розроблення та застосування сучасних системних програм у різних установах та на підприємствах, зокрема АПК.

Місце і роль дисципліни в системі підготовки фахівців відповідно до навчального плану. Дана навчальна дисципліна є теоретичною та практичною основою сукупності знань та вмінь, що формують профіль фахівця в області кібербезпеки. Дисципліна сприяє здачі єдиного державного кваліфікаційного

іспиту зі спеціальності 125 «Кібербезпека» для першого (бакалаврського) рівня вищої освіти.

Вимоги щодо знань і вмінь, набутих внаслідок вивчення дисципліни.

Внаслідок вивчення дисципліни студенти повинні:

знати: основи низькорівневого програмування мовою Асемблера; принципи створення системних програм; моделі організації пам'яті; механізми визначення конфігурації обладнання, на якому виконується системна програма; механізми забезпечення взаємодії системної програми з відео підсистемою; засоби здійснення низькорівневого та високорівневого введення/виведення; принципи реалізації та застосування контейнерів даних; засоби створення та використання динамічно приєднаних бібліотек DLL (Dynamically Linked Libraries); засоби організації міжмодульної взаємодії та взаємодії з бібліотеками середовища програмування; засоби опрацювання переривань, виключних ситуацій, помилок; засоби інтерфейсу Win32 API (Application Programming Interface).

вміти: створювати системні програми на мові Асемблера; працювати зі стеком, оперативною пам'яттю; здійснювати високорівневе та низькорівневе введення/виведення; визначати конфігурацію обладнання, на якому виконується системна програма; створювати системні програми, де передбачається взаємодія з відео підсистемою; створювати багатомодульні програми; створювати та використовувати динамічно приєднані бібліотеки DLL; організовувати міжмодульні взаємодії та взаємодії з бібліотеками середовища програмування; опрацювати переривання; перехоплювати та опрацювати виключні ситуації; опрацювати помилки; створювати віконні додатки на основі засобів інтерфейсу Win32 API.

Набуття компетентностей:

Відповідно до освітньої програми підготовки фахівців за спеціальністю 125 «Кібербезпека» навчальна дисципліна забезпечує формування загальних і фахових компетентностей:

Загальні компетентності (КЗ):

КЗ 1. Здатність застосовувати знання у практичних ситуаціях.

КЗ 4. Вміння виявляти, ставити та вирішувати проблеми за професійним спрямуванням.

Спеціальні (фахові, предметні) компетентності спеціальності (СК):

СК 2. Здатність до використання інформаційно-комунікаційних технологій, сучасних методів і моделей інформаційної безпеки та/або кібербезпеки.

СК 3. Здатність до використання програмних та програмно-апаратних комплексів засобів захисту інформації в інформаційно-телекомунікаційних (автоматизованих) системах.

СК 5. Здатність забезпечувати захист інформації, що обробляється в інформаційно-телекомунікаційних (автоматизованих) системах з метою реалізації встановленої політики інформаційної та/або кібербезпеки.

СК 7. Здатність впроваджувати та забезпечувати функціонування комплексних систем захисту інформації (комплекси нормативно-правових, організаційних та технічних засобів і методів, процедур, практичних прийомів та ін.).

СК 11. Здатність виконувати моніторинг процесів функціонування інформаційних, інформаційно-телекомунікаційних (автоматизованих) систем згідно встановленої політики інформаційної та/або кібербезпеки.

Програмні результати навчання (ПРН):

ПРН 11. Виконувати аналіз зв'язків між інформаційними процесами на віддалених обчислювальних системах.

ПРН 13. Аналізувати проекти інформаційно-телекомунікаційних систем, базуючись на стандартизованих технологіях та протоколах передачі даних.

ПРН 14. Вирішувати завдання захисту програм та інформації, що обробляється в інформаційно-телекомунікаційних системах програмно-апаратними засобами та давати оцінку результативності якості прийнятих рішень.

ПРН 17. Забезпечувати процеси захисту та функціонування інформаційно-телекомунікаційних (автоматизованих) систем на основі практик, навичок та знань, щодо структурних (структурно-логічних) схем, топології мережі, сучасних архітектур та моделей захисту електронних інформаційних ресурсів з відображенням взаємозв'язків та інформаційних потоків, процесів для внутрішніх і віддалених компонент.

ПРН 27. Вирішувати задачі захисту потоків даних в інформаційних, інформаційно-телекомунікаційних (автоматизованих) системах.

3. Програма та структура навчальної дисципліни для:

– повного терміну денної форми навчання.

Змістовий модуль №1. Асемблер. Базові команди, переривання, макроси, процедури.

Тема лекційного заняття 1. Об'єкт, предмет, зміст, завдання та структура курсу. Базові принципи низькорівневого програмування.

Розглядаються об'єкт, предмет та структура курсу. Вивчаються засади низькорівневого програмування на прикладі мови Асемблер. Аналізується архітектура обчислювальної системи. Здійснюється порівняння високорівневої і низькорівневої мов програмування. Розглядається структура програми на мові Асемблер.

Тема лекційного заняття 2. Моделі пам'яті, реєстри, робота зі стеком.

Вивчаються реєстри обчислювальної системи, моделі пам'яті. Вивчається система команд. Розглядаються приклади системних програм, у яких здійснюється робота зі стеком, оперативною пам'яттю.

Тема лекційного заняття 3. Системні переривання, засоби низькорівневого введення/виведення.

Вивчається робота системних переривань, зокрема переривання int 21h. Вивчаються засоби здійснення низькорівневого введення/виведення. Розглядаються та аналізуються відповідні приклади системних програм.

Тема лекційного заняття 4. Арифметичні операції, переходи.

Вивчаються команди виконання арифметичних операцій. Аналізуються аспекти швидкодії відповідних системних програм. Розглядаються приклади системних програм.

Тема лекційного заняття 5. Використання циклів.

Вивчаються засоби реалізації циклічних конструкцій – на основі команд умовного, безумовного переходів, а також на основі спеціалізованих команд. Розглядаються та аналізуються приклади відповідних системних програм.

Тема лекційного заняття 6. Робота з макросами. Макровизначення, макropідстановка, макророзширення.

Вивчаються призначення, аспекти реалізації та застосування макросів. Розглядаються та аналізуються шляхи реалізації макросів – у складі файлу основної програми; у вигляді окремого модуля. Вивчаються відповідні приклади системних програм.

Тема лекційного заняття 7. Використання процедур. Визначення процедур. Виклик процедур.

Вивчаються призначення, аспекти реалізації та застосування процедур. Розглядаються та аналізуються шляхи реалізації процедур. Вивчаються відповідні приклади системних програм.

Змістовий модуль №2. Визначення конфігурації обладнання, робота з відеопідсистемою, створення бібліотек.

Тема лекційного заняття 8. Визначення конфігурації обладнання. Використання спеціалізованих функцій і макросів.

Вивчаються методи та засоби визначення конфігурації програмно-апаратної платформи, на якій виконується системна програма. Для цього залучаються спеціалізовані функції та макроси. Розглядаються та аналізуються приклади відповідних системних програм.

Тема лекційного заняття 9. Визначення конфігурації обладнання. Зчитування вмісту комірок пам'яті.

Вивчаються засоби зчитування конфігураційних даних із залученням бітових полів, спеціалізованих структур, функцій. Розглядаються та аналізуються приклади відповідних системних програм.

Тема лекційного заняття 10. Робота з відеопідсистемою. Спеціалізовані функції і структури.

Вивчаються текстовий і графічний режими роботи з відеопідсистемою, аспекти використання відповідного системного переривання int 10h. Розглядаються та аналізуються приклади відповідних системних програм.

Тема лекційного заняття 11. Робота з відеопідсистемою. Запис до відеопам'яті.

Вивчаються, порівнюються та досліджуються метод запису даних до відеопам'яті з використанням спеціалізованих засобів та метод безпосереднього запису. Розглядаються та аналізуються приклади відповідних системних програм.

Тема лекційного заняття 12. Забезпечення високорівневого введення/виведення. Контейнери даних, ітератори.

Вивчаються засоби здійснення високорівневого введення/виведення, а також супутні засоби – контейнери даних (вектори, списки, асоціативні списки тощо) та літератори. Розглядаються та аналізуються приклади відповідних системних програм.

Тема лекційного заняття 13. Опрацювання помилок і виключних ситуацій.

Вивчаються засоби опрацювання помилок та виключних ситуацій. Розглядаються та аналізуються приклади відповідних системних програм.

Тема лекційного заняття 14. Створення і використання динамічних бібліотек.

Вивчаються засоби створення і використання динамічно приєднаних бібліотек (DLL, Dynamically Linked Libraries). Розглядаються та аналізуються приклади відповідних системних програм.

Тема лекційного заняття 15. Створення віконних застосунків засобами Win32 API.

Вивчаються засоби створення віконних застосунків для середовища Windows – на основі інтерфейсу Win32 API. Вивчаються віконні ресурси. Розглядаються та аналізуються приклади відповідних системних програм.

Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин													
	денна форма							Заочна форма						
	Тижні	Усього	у тому числі					усього	у тому числі					
			л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	
Змістовий модуль 1. Асемблер. Базові команди, переривання, макроси, процедури.														
Тема 1. Об'єкт, предмет, завдання та структура курсу. Базові принципи низькорівневого програмування.	1	10	3		3		4							
Тема 2. Моделі пам'яті, регістри, робота зі стеком.	1	10	3		3		4							

Тема 3. Системні переривання, засоби низькорівневого введення/виведення.	1	10	3	3	4								
Тема 4. Арифметичні операції, переходи.	1	10	3	3	4								
Тема 5. Використання циклів.	1	10	3	3	4								
Тема 6. Робота з макросами. Макровизначення, макropідстановка, макророзширення.	1	10	3	3	4								
Тема 7. Використання процедур. Визначення процедур. Виклик процедур.	1	10	3	3	4								
Разом за змістовим модулем 1	7	70	21	21	28								
Змістовий модуль 2. Визначення конфігурації обладнання, робота з відеопідсистемою, створення бібліотек.													
Тема 8. Визначення конфігурації обладнання. Використання спеціалізованих функцій і макросів.	1	10	3	3	4								
Тема 9. Визначення конфігурації обладнання. Зчитування вмісту комірок пам'яті.	1	10	3	3	4								
Тема 10. Робота з відеопідсистемою. Спеціалізовані функції і структури.	1	10	3	3	4								
Тема 11. Робота з відеопідсистемою. Запис до відеопам'яті.	1	10	3	3	4								
Тема 12. Забезпечення високорівневого введення/виведення.	1	10	3	3	4								

Контейнери даних, ітератори.													
Тема 13. Опрацювання помилок і виключних ситуацій.	1	10	3		3		4						
Тема 14. Створення і використання динамічних бібліотек.	1	10	3		3		4						
Тема 15. Створення віконних застосунків засобами Win32 API.	1	10	3		3		4						
Разом за змістовим модулем 2	8	80	24		24		32						
Усього годин за курс	15	150	45		45		60						
Усього годин	15	150	45		45		60						

4. Теми лабораторних занять.

№ з/п	Назва теми	Кількість годин
1	Базові принципи низькорівневого програмування.	3
2	Арифметичні операції, переходи.	6
3	Використання циклів.	6
4	Робота з макросами.	3
5	Використання процедур.	3
6	Визначення конфігурації обладнання.	6
7	Робота з відеопідсистемою	6
8	Забезпечення високорівневого введення/виведення.	3
9	Опрацювання помилок і виключних ситуацій.	3
10	Створення динамічних бібліотек.	6
	Разом за семестр	45
	Разом	45

5. Теми самостійної роботи.

№ з/п	Назва теми	Кількість годин
1.	Аспекти створення com- і exe-програм.	3
2.	Призначення і використання Program Segment Prefix.	3
3.	Робота з системним таймером. Канали таймера. Визначення у системній програмі поточних дати, часу.	3
4.	Робота з системним таймером. Програвання заданої мелодії на основі системного таймера.	3
5.	Визначення форми курсору у текстовому режимі.	3

6.	Визначення форми курсору у графічному режимі.	3
7.	Контейнери даних у середовищі C++ як елементи бібліотеки STD. Черга.	3
8.	Контейнери даних у середовищі C++ як елементи бібліотеки STD. Двонапрямлена черга.	3
9.	Контейнери даних у середовищі C++ як елементи бібліотеки STD. Вектор.	3
10.	Контейнери даних у середовищі C++ як елементи бібліотеки STD. Список.	3
11.	Контейнери даних у середовищі C++ як елементи бібліотеки STD. Асоціативний список.	3
12.	Контейнери даних у середовищі C++ як елементи бібліотеки STD. Бінарне дерево пошуку.	3
13.	Засоби C++ для створення потоків у системних програмах.	3
14.	Засоби Win32 API. Ресурси вікна. Полоси прокрутки.	3
15.	Засоби Win32 API. Ресурси вікна. Іконки.	3
16.	Засоби Win32 API. Ресурси вікна. Звукові файли.	3
17.	Засоби Win32 API. Ресурси вікна. Відеофайли.	3
18.	Системні програми середовища Windows для роботи з файлами.	3
19.	Системні програми середовища Windows для роботи з каталогами.	3
20.	Створення власних оброблювачів переривань у середовищі DOS.	3
	Разом	60

6. Зразки контрольних питань, тестів для визначення рівня засвоєння знань студентами.

Змістовий модуль 1.

1. Призначення асемблерів. Порівняльна характеристика (TASM, MASM, FASM, NASM).

2. Призначення та застосування компілятора при створенні системних програм на мові асемблера. Навести приклад застосування компілятора.

3. Призначення та застосування компонувальника (лінкера) при створенні системних програм на мові асемблера. Навести приклад застосування компонувальника.

4. Призначення та застосування інструментарію відлагодження («дебагеру») при створенні системних програм на мові асемблера. Навести приклад застосування «дебагеру».

5. Моделі організації пам'яті в TASM. Коротка характеристика.

6. Приклад асемблерної програми, побудованої на основі директиви MODEL. Навести коментарі.

7. Приклад асемблерної програми, побудованої на основі директиви ASSUME. Навести коментарі.

8. Пояснити відмінності між виконавчими файлами *.com та *.exe. Навести консольні команди одержання відповідних файлів.

9. Механізм роботи структури даних "стек". Навести програму обміну значень регістрів через стек. Прокоментувати, як при виконанні команд PUSH і POP змінюється значення сегментного регістра SP.

10. Регістри загального призначення. Приклади використання.

11. Сегментні регістри. Приклади використання.

12. Регістр прапорців. Призначення, приклад застосування.

13. Призначення макросів. Навести приклад макровизначення, макровиклику засобами асемблера.
14. Призначення процедур. Навести приклад визначення, виклику процедури засобами асемблера.
15. Команди асемблера. Команди побітового зсуву як аналоги операцій множення і ділення. Навести приклади, прокоментувати.
16. Команди асемблера. Команди умовного переходу як засоби організації циклів. Навести приклад, прокоментувати.
17. Команди асемблера. Команди безумовного переходу як засоби організації циклів. Навести приклад, прокоментувати.
18. Команди асемблера. Спеціалізовані команди організації циклів. Навести приклад, прокоментувати.
19. Засоби низькорівневого введення. Навести приклад відповідної програми на мові асемблера, прокоментувати.
20. Засоби низькорівневого виведення. Навести приклад відповідної програми на мові асемблера, прокоментувати.
21. Функції переривання INT 21h (DOS-рівень). Навести приклад.
22. Функції переривання INT 33h. Навести приклад.
23. Функції переривання INT 10h (BIOS-рівень). Навести приклад.

Змістовий модуль 2.

24. Засоби високорівневого введення (потокове введення). Навести приклад відповідної програми на мові C++, прокоментувати.
25. Засоби високорівневого виведення (потокове виведення). Навести приклад відповідної програми на мові C++, прокоментувати.
26. Попіксельне виведення даних засобами заголовочного файлу <windows.h>. Навести приклад, прокоментувати.
27. Визначення конфігурації обладнання. Навести і прокоментувати код програми, що дозволяє визначити обсяг доступної оперативної пам'яті.
28. Визначення конфігурації обладнання. Навести і прокоментувати код програми, що дозволяє визначити стан реєстрів клавіатури.
29. Асемблерні вставки у мові C++. Призначення. Навести приклад використання, прокоментувати.
30. Засоби опрацювання виключних ситуацій. Навести і прокоментувати код програми, де опрацьовується виключна ситуація.
31. Створення і опрацювання виключних ситуацій власних типів. Навести приклад.
32. Засоби опрацювання помилок. Навести і прокоментувати код програми, де опрацьовується помилка відкриття файлу.
33. DLL-бібліотеки. Призначення і використання. Навести і прокоментувати приклад заголовочного файлу на мові C++, де визначається склад функцій бібліотеки.
34. Реалізація клієнтського застосунку для використання функцій DLL-бібліотеки. Навести приклад, прокоментувати.
35. Програмування на основі засобів Win32 API. Функція WinMain. Призначення, використання.

36. Програмування на основі засобів Win32 API. Функція WindowProc. Призначення, використання.

37. Програмування на основі засобів Win32 API. Ресурси. Приклади ресурсів. Навести і прокоментувати приклад заголовочного файлу, де визначаються константні подання ресурсів.

38. Програмування на основі засобів Win32 API. Навести і прокоментувати приклад елементарної Win32-програми на основі MessageBox.

39. Програмування на основі засобів Win32 API. Меню. Навести приклад визначення структури меню у заголовочному файлі. Прокоментувати програмний код.

40. Програмування на основі засобів Win32 API. Меню. Навести приклад підключення меню у якості ресурсу. Прокоментувати програмний код.

41. Програмування на основі засобів Win32 API. Меню. Навести приклад опрацювання повідомлень від елементів меню. Прокоментувати програмний код.

7. Методи навчання.

Під час викладання курсу використовуються наступні методи навчання:

- розповідь – для оповідної, описової форми розкриття навчального матеріалу;
- пояснення – для розкриття сутності певного явища, закону, процесу;
- бесіда – для усвідомлення, за допомогою діалогу, нових явищ, понять;
- ілюстрація – для розкриття предметів і процесів через їх символічне зображення (рисунок, схеми, графіки);
- лабораторна робота – для використання набутих знань при виконанні лабораторних завдань;
- аналітичний метод – для мисленнєвого або практичного розкладу цілого на частини з метою вивчення їх суттєвих ознак;
- проблемний виклад матеріалу – для створення проблемної ситуації.

8. Форми контролю.

Наприкінці кожного змістовного модуля проводиться контрольна робота.

Перший змістовий модуль – захист п'яти лабораторних робіт, усне опитування, контрольна робота – тест.

Другий змістовий модуль – захист п'яти лабораторних робіт, усне опитування, контрольна робота – тест, екзамен.

9. Розподіл балів, які отримують студенти.

Оцінювання знань студента відбувається за 100-бальною шкалою і переводиться в національні оцінки згідно з табл. 1 «ПОЛОЖЕННЯ про екзамен та заліки у НУБіП України» (наказ про уведення в дію від «26» квітня 2023 р. протокол № 10):

Рейтинг студента, бали	Оцінка національна за результати складання	
	екзаменів	заліків

90-100	Відмінно	Зараховано
74-89	Добре	
60-73	Задовільно	
0-59	Незадовільно	Не зараховано

Для визначення рейтингу студента (слухача) із засвоєння дисципліни $R_{\text{дис}}$ (до 100 балів) одержаний рейтинг з атестації (до 30 балів) додається до рейтингу студента (слухача) з навчальної роботи $R_{\text{НР}}$ (до 70 балів): $R_{\text{дис}} = R_{\text{НР}} + R_{\text{АТ}}$.

Оцінювання виконання та захисту лабораторних робіт за кожний модуль здійснюється у наступній відповідності:

№ лабораторної роботи	Кількість балів	Загальна кількість балів
1 модуль		
Лабораторна робота № 1	12	70
Лабораторна робота № 2	12	
Лабораторна робота № 3	12	
Лабораторна робота № 4	12	
Лабораторна робота № 5	12	
Самостійна робота	10	
Модульна контрольна		30
2 модуль		
Лабораторна робота № 6	10	70
Лабораторна робота № 7	10	
Лабораторна робота № 8	10	
Лабораторна робота № 9	10	
Лабораторна робота № 10	10	
Лабораторна робота № 11	10	
Самостійна робота	10	
Модульна контрольна		30

10. Навчально-методичне забезпечення.

1. Методичні вказівки до лабораторних робіт з дисципліни "Системне програмування" для студентів спеціальності 123 "Комп'ютерна інженерія" всіх форм навчання – Частина 1 / Укл.: В.В. Шкарупило. – Київ: НУБіП, 2018. – 42 с. URL: <https://elearn.nubip.edu.ua/mod/resource/view.php?id=201679>

2. Методичні вказівки до лабораторних робіт з дисципліни "Системне програмування" для студентів спеціальності 125 "Кібербезпека" всіх форм навчання / Укл.: В.В. Шкарупило. – Київ: НУБіП, 2022. – 74 с. (прийнято до друку). URL: <https://elearn.nubip.edu.ua/mod/resource/view.php?id=240416>

3. Відеоматеріал до вступної лекції. URL: <https://elearn.nubip.edu.ua/mod/url/view.php?id=396024> (дата звернення: 08.05.2022).

4. Відеоінструкція до виконання л.р. №6. URL: <https://elearn.nubip.edu.ua/mod/url/view.php?id=394994> (дата звернення: 08.05.2022).

5. Відеоінструкція до виконання л.р. №8. URL: <https://elearn.nubip.edu.ua/mod/url/view.php?id=396647> (дата звернення: 08.05.2022).
6. Відеоінструкція до виконання л.р. №9. URL: <https://elearn.nubip.edu.ua/mod/url/view.php?id=401692> (дата звернення: 08.05.2022).

11. Рекомендовані джерела інформації.

– основні:

1. Піза Д.М. Асемблер: навчальний посібник / Д.М. Піза, Б.Т. Солдатов. – Запоріжжя: ЗНТУ, 2004. – 167 с.
2. Рисований О.М. Системне програмування : підручник для студентів напрямку "Комп'ютерна інженерія" вищих навчальних закладів в 2-х томах. Т. 1. Вид. 4-е, виправлено та доповнено. Харків : Слово, 2015, 576 с.

– допоміжні:

1. Petzold C. Programming Windows: 5th edition. Microsoft Press. 1998. 1100 p. ISBN-13: 978-1572319950
2. theForger's Win32 API Programming Tutorial. URL: <http://www.winprog.org/tutorial/> (дата звернення: 08.05.2022).
3. Getting started with Win32 API. URL: <https://riptutorial.com/winapi> (дата звернення: 08.05.2022).
4. Understanding MAP files generated by the Linker. URL: <https://community.embarcadero.com/article/technical-articles/149-tools/15481-understanding-map-files-generated-by-the-linker> (дата звернення: 08.05.2022).
5. ДСТУ ISO/IEC 2382:2017 (ISO/IEC 2382:2015, IDT) Інформаційні технології. Словник термінів.