

ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

СИСТЕМА УПРАВЛІННЯ ЯКІСТЮ

Система забезпечення якості освітньої діяльності та якості вищої освіти
сертифікована на відповідність ДСТУ ISO 9001:2015 / ISO 9001:2015

Кафедра комп'ютерних наук та інформаційних систем

ЗАТВЕРДЖЕНО

вченою радою

(пост. п. 8 від «27» 10 20 22р.)

Ректор



Анаторій МАЗАРАКІ

СИСТЕМНЕ ПРОГРАМУВАННЯ /
SYSTEM PROGRAMMING

РОБОЧА ПРОГРАМА /
COURSE OUTLINE

освітній ступінь	бакалавр	/	bachelor
галузь знань	<u>12 Інформаційні технології</u>	/	<u>Information Technology</u>
спеціальність	<u>126 Інформаційні системи та технології</u>	/	<u>Information Systems and Technologies</u>
освітня програма	<u>Інформаційні системи та технології</u>	/	<u>Information Systems and Technologies</u>

Київ 2022

**Розповсюдження і тиражування без офіційного дозволу ДТЕУ
заборонено**

Автор: Паращак О.М., кандидат технічних наук, доцент

Робочу програму розглянуто і затверджено на засіданні кафедри комп'ютерних наук та інформаційних систем 28 червня 2022р., протокол № 35

Рецензенти: Демідов П.Г., кандидат технічних наук, доцент
Козлов В.В., кандидат технічних наук, доцент
Івашина В. Ф., керівник проектів програм у сфері нематеріального виробництва «ІнфоПлюс Центр комп'ютерних технологій»

**СИСТЕМНЕ ПРОГРАМУВАННЯ /
SYSTEM PROGRAMMING**

**РОБОЧА ПРОГРАМА /
COURSE OUTLINE**

освітній ступінь	бакалавр	/ Bachelor
галузь знань	<u>12 Інформаційні технології</u>	/ <u>Information Technology</u>
спеціальність	<u>126 Інформаційні системи та технології</u>	/ <u>Information systems and technologies</u>
освітня програма	<u>Інформаційні системи та технології</u>	/ <u>Information systems and technologies</u>

1. СТРУКТУРА ДИСЦИПЛІНИ ТА РОЗПОДІЛ ГОДИН ЗА ТЕМАМИ (ТЕМАТИЧНИЙ ПЛАН)

Назва теми	Кількість годин				
	Усього годин / кредитів	з них			Форми контролю
		лекції	лабораторні заняття/ МК	самостійна робота студентів	
1	2	3	5	6	7
Тема 1. Проблеми системного програмування і вибір засобів для їх вирішення.	8	1	2	5	О, ПСР, ППР
Тема 2. Архітектура мікропроцесора Intel. Подання даних у комп'ютері..	8	1	2	5	О, ПСР, ППР
Тема 3. Мова асемблера як основа системного програмування	10	2	2	6	О, ПСР, ППР
Тема 4. Основні директиви та операнди Асемблера	8	1	2	5	О, ПСР, ППР
Тема 5. Операції переселення даних	10	2	2	6	О, ПСР, ППР
Тема 6. Основні арифметичні операції	11	2	2	7	О, ПСР, ППР
Тема 7. Логічні команди та команди зсуву	11	2	2	7	О, ПСР, ППР
Тема 8. Команди передавання керування	10	2	2	6	О, ПСР, ППР
Тема 9. Процедури	14	2	4	8	О, ПСР, ППР
Тема 10. Організація введення-виведення у Win32	14	2	4	8	О, ПСР, ППР
Тема 11. Математичний співпроцесор	14	2	4	8	О, ПСР, МК
Тема 12. Рядки	9	1	2	6	О, ПСР,
Тема 13. Макровизначення	14	2	4	8	О, ПСР, ППР
Тема 14. Динамічні бібліотеки.	11	2	2	7	О, ПСР, ППР
Тема 15. Файли.	10	2	2	6	О, ПСР, ППР
Тема 16. Логічні конструкції високого рівня	9	1	2	6	О, ПСР, ППР

Тема 17. Структури	9	1	2	6	О, ПСР, ППР
Разом	180/6	28	42	110	

Умовні позначення: ПСР – перевірка самостійної роботи; МК – модульний контроль; ППР – перевірка практичної роботи; О – опитування.

2. ТЕМАТИКА ТА ЗМІСТ ЛЕКЦІЙНИХ, ПРАКТИЧНИХ (СЕМІНАРСЬКИХ), ЛАБОРАТОРНИХ ЗАНЯТЬ, САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ

Результати навчання	Навчальна діяльність*	Робочий час студента, год
1	2	3
<p>Знати: теоретичні та методологічні принципи створення програм, які взаємодіють з операційною системою, або апаратним забезпеченням комп'ютера.</p> <p>Вміти: застосовувати машинно-орієнтовані мови для розробки системних програм для ОС Windows</p>	<p>Тема 1. Проблеми системного програмування і вибір засобів для їх вирішення. Лекція № 1. Проблеми системного програмування і вибір засобів для їх вирішення. План лекції: 1. Функції системних програм. 2. Машинно-орієнтовані мови і їх використання в системному програмуванні. 3. Системне програмування в ОС Windows. 4. Структура ОС Windows. 5. Системні виклики, переривання, виняткові ситуації.</p> <p><i>Список рекомендованих джерел:</i> <i>Основний: 1 [с.86-91]</i> <i>Додатковий: 5 [с.51-67]</i></p>	1
	<p>Самостійна робота студентів. Питання для самостійного опрацювання: 1. Функції системних програм? 2. Класифікація машинно-орієнтовані мов програмування? 3. Які переваги програм написаних на Асемблері? 4. Які суттєві недоліки асемблерних програм? 5. Для яких типів задач використовується мова програмування Асемблер?</p> <p><i>Список рекомендованих джерел:</i> <i>Основний: 1 [с.86-91]</i> <i>Додатковий: 5 [с.51-67]</i></p>	5
	<p>Тема. Проблеми системного програмування і вибір засобів для їх вирішення. Лабораторне заняття №1. Проблеми системного програмування і вибір засобів для їх вирішення. Завдання до заняття: 1. Вивчити структуру та архітектурні особливості ОС Windows. 2. Знайомство з сучасними тенденціями в розробці ОС.</p>	2

	<p>3. Технології створення асемблерних програм для розширення функціональних можливостей ОС.</p> <p>4. Методи програмування на Асемблері апаратних засобів комп'ютера.</p>	
<p>Знати: архітектуру та організацію пам'яті персонального комп'ютера, а також способи адресації та представлення даних в комп'ютері.</p> <p>Вміти: при написанні програм використовувати реєстри за призначенням, способи адресації та подання даних в комп'ютері.</p>	<p>Тема 2. Архітектура мікропроцесора (МП) Intel. Подання даних у комп'ютері.</p> <p>Лекція № 2. Архітектура мікропроцесора (МП) Intel. Подання даних у комп'ютері.</p> <p>План лекції:</p> <ol style="list-style-type: none"> 1. Архітектурні особливості МП Intel та його типи даних. 2. Регістри та моделі пам'яті МП Intel. 3. Способи адресації. 4. Позиційні системи числення. 5. Двійкові, вісімкові та шістнадцяткові числа. 6. Формати представлення в комп'ютері цілих чисел та плаваючою комою. <p><i>Список рекомендованих джерел:</i> Основний: 1 [с.7-16], 2 [с.27-53] Додатковий: 5 [с.9-50]</p>	1
	<p>Самостійна робота студентів.</p> <p>Питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> 1.Програмна архітектура МП? 2.На які групи поділяються реєстри МП? 3.Режими роботи МП? 4.Організація пам'яті та способи адресації? 5.Представлення даних в двійковій та шістнадцятковій системі числення та алгоритм переходу з однієї системи в іншу ? 6. Формат представлення даних в комп'ютері? <p><i>Список рекомендованих джерел:</i> Основний: 1 [с.7-16], 2 [с.27-53] Додатковий: 5 [с.9-50]</p>	5
	<p><i>Тема.</i> Архітектура мікропроцесора (МП) Intel. Подання даних у комп'ютері.</p> <p>Лабораторне заняття № 2. Архітектура мікропроцесора (МП) Intel. Подання даних у комп'ютері.</p> <p>Завдання до заняття:</p> <ol style="list-style-type: none"> 1.Вивчення програмної моделі МП Intel. 2.Аналіз режимів роботи МП. 3.Вивчення реєстрів МП та їх використання при розробці асемблерних програм. 4. Способи адресації та методи їх ефективного використання. 5. Формати представлення даних в комп'ютері . 	2
<p>Знати: асемблер Masm32, інтегроване середовище для розробки</p>	<p>Тема 3. Мова асемблера як основа системного програмування.</p> <p>Лекція №3. Мова асемблера як основа системного програмування.</p> <p><i>План лекції:</i></p> <ol style="list-style-type: none"> 1.Критерії вибору асемблера. 	2

<p>програм на мові асемблер (SASM), етапи створення програм та методи їх налагодження</p> <p>Вміти: інсталювати асемблер Masm32, установити та налаштувати інтегроване середовище SASM з допомогою якого пройти всі етапи створення програми.</p>	<p>2. Інсталяція асемблера Masm32 (Microsoft Macro Assembler). 3. Установка кросплатформеного середовища для розробки програмного забезпечення на мові асемблер (SASM). 4. Етапи створення програми. 5. Зменшення розміру програми за рахунок ключів компіляції та лінкування в Masm32. 6. Налагодження програм.</p> <p>Список рекомендованих джерел: <i>Основний:</i> 2[с.19-28,7] <i>Додатковий:</i> 6[с.5-17] <i>Internet-ресурси:</i> 7,8</p>	
	<p>Самостійна робота студентів. Питання для самостійного опрацювання: 1. Критерії вибору асемблера? 2. Призначення асемблера Masm32? 3. Функції кросплатформеного середовища для розробки програм на мові асемблер (SASM)? 4. Етапи створення програми? 5. Способи зменшення розміру програми ? 6. Функції програми ollydbg.exe для налагодження програм?</p> <p>Список рекомендованих джерел: <i>Основний:</i> 2[с.19-28,7] <i>Додатковий:</i> 6 [с.5-17] <i>Internet-ресурси:</i> 7,8</p>	6
	<p>Тема. . Мова асемблера як основа системного програмування. Лабораторне заняття №3. Мова асемблера як основа системного програмування. Завдання до заняття: 1. Інсталювати асемблера Masm32 (Microsoft Macro Assembler). 2. Установити та налагодити кросплатформене середовище для розробки програм SASM. 3. Написати програму в SASM 4. Провести налагодження програми в SASM та за допомогою програми ollydbg.exe.</p>	2
<p>Знати: директиви та операнди Асемблера. Вміти: правильно застосовувати директиви та операнди Асемблера при розробці структури асемблерних програм і визначенні</p>	<p>Тема 4. Основні директиви та операнди Асемблера. Лекція № 4. Основні директиви та операнди Асемблера. План лекції: 1. Директиви керування моделями пам'яті MODEL та сегментами DATA, CONST, STACK, CODE. 3. Директиви початку і закінчення сегмента SEGMENT та ENDS. 4. Директиви початку і закінчення процедури PROC та ENDP. 5. Директива визначення пам'яті та призначення сегментів ASSUME. 6. Директива для зміни вмісту програмного лічильника ORG. 7. Директиви визначення даних DB, DW, DD, DF, DP, DQ, DT. 8. Директиви символічних констант: = , EQU, TEXTEQU. 9. Директива завершення програми END. 10. Директива призначена для встановлення відповідності між</p>	1

даних.	<p>іменем і типом змінних LABEL.</p> <p>11. Директива вирівнювання адреси ALIGN.</p> <p>12. Директива визначення локальності ідентифікаторів LOCALS.</p> <p>13. Директиви керування файлами INCLUDE, INCLUDELIB.</p> <p>14. Оператори мови асемблера.</p> <p>Список рекомендованих джерел: <i>Основний: 2[с.56-69]</i></p>	
	<p>Самостійна робота студентів.</p> <p>Питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> 1. Для чого використовуються директиви асемблера? 2. Для чого використовуються директиви керування моделями пам'яті? 3. Яку модель пам'яті використовує ОС Windows? 4. Для яких цілей використовують директиви визначення даних? 5. Які директиви визначення даних використовують для визначення числа зі знаком? 6. Які директиви визначення даних використовують для визначення дійсних чисел? 7. Для чого використовують директиви символічних констант? <p>Список рекомендованих джерел: <i>Основний: 2[с.56-69]</i></p>	5
	<p>Тема. Основні директиви та операнди Асемблера.</p> <p>Лабораторне заняття №5. Основні директиви та операнди Асемблера.</p> <p>Завдання до заняття:</p> <p>Завдання 1. Внутрішнє подання цілочислових даних</p> <ol style="list-style-type: none"> 1. Вибрати для свого варіанта цілі числа. 2. Перевести їх із десяткової у двійкову систему числення. 3. Одержати їх внутрішнє подання. 4. Написати програму опису цих чисел мовою асемблера й одержати лістинг програми. 5. Перевірити правильність своїх викладок. <p>Завдання 2. Внутрішнє подання дійсних чисел:</p> <ol style="list-style-type: none"> 1. Вибрати для свого варіанта дійсні числа. 2. Перевести їх із десяткової у двійкову систему числення і зробити нормалізацію чисел. 3. Одержати їх внутрішнє подання. 4. Написати програму опису цих чисел мовою асемблера й одержати лістинг програми. 5. Перевірити правильність своїх викладок. 	2
<p>Знати: всі типи команд пересилання даних та їх призначення.</p> <p>Вміти: застосовувати команди</p>	<p>Тема 5. Операції пересилання даних.</p> <p>Лекція № 5. Операції переселення даних.</p> <p>План лекції:</p> <ol style="list-style-type: none"> 1. Команди пересилання даних загального призначення. 2. Організація роботи зі стеком. 3. Команди роботи з адресами і вказівником. 4. Команди перетворення даних. 5. Команди введення та виведення в порт. 6. Команди пересилання бітів умов. 	2

пересилання даних при розробці асемблерних програм.	<p>Список рекомендованих джерел: <i>Основний: 1 [с.57-62], 2 [с.80-90]</i></p> <p>Самостійна робота студентів.</p> <ol style="list-style-type: none"> 1. Команди пересилання даних їх призначення та синтаксис? 2. Які команди пересилання даних використовують для передання даних за умовою? 3. Призначення команд роботи зі стеком? 4. Для яких цілей використовують команди роботи з адресами і вказівником? 5. Призначення команд перетворення даних? 6. Команди введення та виведення в порт та їх призначення? 7. Для чого використовуються команди пересилання бітів умов? <p>Список рекомендованих джерел: <i>Основний: 1 [с.57-62], 2 [с.80-90]</i></p>	6
	<p>Тема. Операції пересилання даних. Лабораторне заняття №5. Операції пересилання даних.. <i>Завдання до заняття:</i></p> <ol style="list-style-type: none"> 1. Відповідно до свого варіанту завдання, використовуючи команди пересилання даних та прямої, базової, індексно-базової адресації, розробити алгоритм і структуру даних програми. 2. Написати програму за розробленим алгоритмом. 3. Розробити тест для перевірки роботи програми 4. Перевірити роботу програми. 5. Оформити звіт. 	2
<p>Знати: типи команди арифметичних операцій та їх призначення. Вміти: застосовувати команди арифметичних операцій при розробці асемблерних програм.</p>	<p>Тема 6. Основні арифметичні операції. Лекція №6. Основні арифметичні операції. План лекції:</p> <ol style="list-style-type: none"> 1. Команди додавання чисел. 2. Команди перетворення типів. 3. Команди віднімання та множення. 4. Команди ділення. 5. Команди двійково-десятькової арифметики. <p>Список рекомендованих джерел: <i>Основний: 2 [с.92-118]</i> <i>Додатковий: 6 [с. 27-39]</i></p>	2
	<p>Самостійна робота студентів. Питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> 1. Які вимоги ставляться до операндів операцій додавання і віднімання? 2. У чому відмінність виконання команд ADD і ADC та в якому випадку обидві команди дають однаковий результат? 3. Які команди використовуються для додавання (віднімання) двох чисел, якщо їх розмір перевищує розрядність процесора? 4. Які прапори встановлюються при виконанні операції додавання? 	7

	<p>5. При виконанні операції множення, які регістри використовуються для зберігання першого співмножника?</p> <p>6. У яких регістрах формується добуток двох цілих чисел?</p> <p>7. Після виконання операції множення які прапори встановлюються у процесорі?</p> <p>8. У чому відмінність команд MUL і IMUL?</p> <p>9. Які прапори встановлюються у процесорі після виконання ділення цілих чисел?</p> <p>10. За допомогою якої команди можна отримати залишок від ділення двох чисел?</p> <p>11. Для чого служить команда CWD?</p> <p>Список рекомендованих джерел: <i>Основний: 2 [с.92-118]</i> <i>Додатковий: 6 [с. 27-39]</i></p>	
	<p>Тема. Основні арифметичні операції</p> <p>Лабораторне заняття №6. Основні арифметичні операції</p> <p><i>Завдання до заняття:</i></p> <p>1. Відповідно до заданого варіанту розробити програму для обчислення арифметичного виразу.</p> <p>2. Розробити структуру даних та алгоритм програми та написати програму, що реалізує обчислення арифметичного виразу. При виконанні обчислень здійснити контроль переповнювання цілих чисел та ділення на нуль.</p> <p>3. Розробити три тести для перевірки роботи програми: перший тест для перевірки працездатності, другий тест – для перевірки ділення на нуль, третій – для перевірки виявлення переповнювання при виконанні додавання і віднімання.</p> <p>4. Перевірити роботу програми.</p> <p>5. Оформити звіт.</p>	2
<p>Знати: типи логічних команди, команди зсуву та їх призначення.</p> <p>Вміти: застосовувати логічні команди та команди зсуву при розробці асемблерних програм.</p>	<p>Тема 7. Логічні команди та команди зсуву.</p> <p>Лекція №7. Логічні команди та команди зсуву.</p> <p>План лекції:</p> <p>1. Команди булевих операцій.</p> <p>2. Команди перевірки і модифікації бітів.</p> <p>3. Команди сканування бітів.</p> <p>4. Команди зсуву.</p> <p>5. Команди циклічного зсуву.</p> <p>6. Команди подвійного зсуву.</p> <p>7. Команди установки байта за умовою.</p> <p>8. Команда перевірки.</p> <p>Список рекомендованих джерел: <i>Основний: 2 [с. 123-131]</i> <i>Додатковий: 5 [с. 130-134]</i></p>	2
	<p>Самостійна робота студентів.</p> <p>Питання для самостійного опрацювання:</p>	7

	<ol style="list-style-type: none"> 1. Наведіть приклади використання логічних команд та команд зсуву? 2. Поясніть різницю між арифметичним зсувом та логічним? 3. Наведіть приклади використання подвійного зсуву та їх практичне значення? 4. Як встановити необхідні розряди змінної в нульове значення? 5. Які прапори змінюються у процесорі при виконанні логічних операцій? 6. На яку максимальну кількість розрядів можна зсунути значення змінної? 7. У чому відмінність логічного зсуву від циклічного? 8. Чим відрізняється виконання команд ROR і RCR? <p>Список рекомендованих джерел: Основний: 2 [с. 123-131] Додатковий: 5 [с.130-134]</p>	
	<p>Тема. Логічні команди та команди зсуву. Лабораторне заняття №7. Логічні команди та команди зсуву. Завдання:</p> <ol style="list-style-type: none"> 1. Відповідно до свого варіанту завдання, використовуючи логічні команди та команди зсуву, розробити алгоритм і структуру даних програми. 2. Написати програму за розробленим алгоритмом. 3. Розробити тест для перевірки роботи програми. 4. Перевірити роботу програми з використанням тесту. 5. Оформити звіт. 	2
<p>Знати: типи команд передачі управління та їх призначення. Вміти: застосовувати команди передачі управління при розробці асемблерних програм.</p>	<p>Тема 8. Команди передачі управління Лекція № 8. Команди передачі управління. План лекції.</p> <ol style="list-style-type: none"> 1. Команда безумовного переходу. 2. Команди умовного передання управління. 3. Команди керування циклами. <p>Список рекомендованих джерел: Основний: 2 [с.133-143] Додатковий: 6 [с.18-26]</p> <p>Самостійна робота студентів. Питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> 1. Що розуміється під терміном "непряма передача управління"? 2. Яка команда здійснює передачу управління, якщо попередня команда дала результат у формі позитивного числа? 3. За допомогою яких команд можна побудувати цикл у програмі на асемблері? 4. У чому відмінність команди LOOPE від LOOP? 5. Яку максимальну кількість циклів повторення може забезпечити команда LOOP? <p>Список рекомендованих джерел:</p>	2
		6

	<p><i>Основний: 2 [с.133-143],5</i> <i>Додатковий: 6 [с.18-26]</i></p>	
	<p>Тема. Команди передачі управління. Лабораторне заняття №8. . Команди передачі управління. Завдання: 1. Відповідно до свого варіанту завдання розробити алгоритм і структуру даних для програми обчислення виразу. 2. Написати програму за розробленим алгоритмом з використанням команд передачі управління. 3. Розробити тест для перевірки роботи програми. 4. Перевірити роботу програми з використанням тесту. 5. Оформити звіт.</p>	2
<p>Знати: призначення процедур і їх використання в процедурно-орієнтованому програмуванні Вміти: використовувати процедури при проектуванні та розробці програм</p>	<p>Тема 9. Процедури Лекція №10. Процедури План лекції 1. Виклик (команда CALL) процедури та її повернення (команда RET). 2. Команди керування стеком при виконанні процедур. 3. Директиви LOCAL, INVOKE, PROTO. 4. Оператори ADDR, USES. 5. Організація процедур та їх дослідження. Windows API-подібні процедури. 6. Непрямий виклик процедур. 7. Використання загальних змінних у процедурах.</p> <p>Список рекомендованих джерел: <i>Основний: 2 [с.148-182]</i></p>	2
	<p>Самостійна робота студентів. Питання для самостійного опрацювання: 1. Які переваги та недоліки використання процедур? 2. Як оголошуються локальні і глобальні змінні в процедурах? 3. Які переваги дає використання директиви invoke? 4. Який оператор дозволяє перерахувати імена всіх регістрів, значення яких змінюється в процедурі? 6. Як організовується та виконується процедура? 7. Які процедури є Windows API-подібними? 8. Як виконується непрямої виклик процедур? 10. Які директиви використовуються для роботи із загальними змінними? 11. Який модуль при об'єднанні декількох модулів записується першим? 12. Які файли мають розширення .INC? 13. За допомогою якої директиви виконується підключення додаткового файлу?</p> <p>Список рекомендованих джерел: <i>Основний: 2[с.148-182]</i></p>	8
	<p>Тема. Процедури. Лабораторне заняття №9. Лабораторна робота Використання</p>	4

	<p>процедур в програмах. <i>Завдання до заняття:</i> Завдання 1. 1. Відповідно до свого варіанту завдання розробити алгоритм і структуру даних для програми обчислення складного виразу з використанням процедур. 2. Написати програму за розробленим алгоритмом. 3. Розробити тест для перевірки роботи програми. 4. Перевірити роботу програми з використанням тесту. 5. Оформити звіт.</p> <p>Завдання 2. 1. Згідно з отриманим варіантом розробити алгоритм і структуру даних програми оформленої у вигляді процедури з параметрами. 2. Написати програму за розробленим алгоритмом. 3. Розробити тест для перевірки роботи програми. 4. Перевірити роботу програми з використанням тесту. 5. Оформити звіт.</p>	
<p>Знати: API-функції Windows, і їх використання для введення-виведення даних</p> <p>Вміти: застосовувати API-функції Windows в асемблерних програмах для введення-виведення даних</p>	<p>Тема 10. Організація введення-виведення у Win32. Лекція №10. Організація введення-виведення у Win32. План лекції. 1. Загальні відомості про введення – виведення у Win32. 2. Набір символів. API-функції Windows. 3. Типи даних Windows, дескриптори консолі, API-функції консолі. 4. Виведення повідомлень і чисел. 5. Введення з консолі. 6. Керування кольором.</p> <p><i>Список рекомендованих джерел:</i> <i>Основний: 2 [с.187-215], 3 [с.7-33], 4 [с.42-96]</i></p>	2
	<p>Самостійна робота студентів. Питання для самостійного опрацювання: 1. Що таке дескриптор консолі? 2. Які функції Win32 API використовуються для виведення повідомлень? 4. Яка функція Win32 API здійснює перетворення 32-розрядного числа в послідовність символів? 5. Яка функція Win32 API створює спрощене вікно?</p> <p><i>Список рекомендованих джерел:</i> <i>Основний: 2[с.187-215],3 [с.7-33], 4 [с.42-96]</i></p>	8
	<p>Тема. Організація введення-виведення у Win32. Лабораторне заняття №10. Лабораторна робота Введення-виведення даних <i>Завдання до заняття:</i> 1. Відповідно до варіанта завдання розробити алгоритм і структуру даних програми виведення даних, використовуючи</p>	4

	<p>функцію MessageBoxIndirect з повідомленнями свого прізвища та номера своєї навчальної групи.</p> <ol style="list-style-type: none"> Написати програму за розробленим алгоритмом. Розробити тест для перевірки роботи програми. Перевірити роботу програми з використанням тесту. Оформити звіт. 	
<p>Знати: архітектуру математичного співпроцесора та систему його команд</p> <p>Вміти: отримати навички роботи з дійсними числами за допомогою математичного співпроцесора, вивчити систему команд арифметики дійсних чисел.</p>	<p>Тема 11. Математичний співпроцесор Лекція №11 Математичний співпроцесор План лекції.</p> <ol style="list-style-type: none"> Архітектура і типи даних математичного співпроцесора. Система команд співпроцесора. Адміністративні команди. Цикли і розгалуження при роботі з співпроцесором. Програмування співпроцесора з використанням операцій обчислення математичних функцій. <p>Список рекомендованих джерел: <i>Основний: 2 [с.267-313]</i></p>	2
	<p>Самостійна робота студентів. Питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> Яку архітектуру має співпроцесор? Для чого служить співпроцесор? Як взаємодіє центральний процесор з співпроцесором при виконанні його команд? Які типи даних обробляє співпроцесор? Який діапазон дійсних чисел допустимий у співпроцесорі? Для чого служить регістровий стек співпроцесора? Що зберігає регістр ознак співпроцесора? Для чого служить управляюче слово співпроцесора? Яка інформація записується в слово стану співпроцесора? Які арифметичні команди підтримуються співпроцесором? Що таке команди із зворотним порядком виконання? За допомогою яких команд здійснюється порівняння дійсних чисел? <p>Список рекомендованих джерел: <i>Основний: 2 [с.267-313]</i></p>	8
	<p>Тема. Математичний співпроцесор Лабораторне заняття №11. Лабораторна робота Обробка даних у співпроцесорі. <i>Завдання до заняття:</i></p> <ol style="list-style-type: none"> Відповідно до варіанта завдання розробити алгоритм і структуру даних програми обчислення виразу із використанням дійсних чисел та виведенням результатів обчислень на екран. Написати програму за розробленим алгоритмом. Розробити тест для перевірки роботи програми. Перевірити роботу програми з використанням тесту. Оформити звіт. 	4
Знати:	Тема 12. Рядки.	1

<p>команди обробки блоків пам'яті</p> <p>Вміти: скласти, налагоджувати і виконувати програми, написаних мовою асемблера, з використанням команд обробки</p>	<p>Лекція №11 Рядки. План лекції.</p> <ol style="list-style-type: none"> 1.Призначення команд обробки рядків. 2.Команда переміщення рядків з однієї частини пам'яті у іншу MOVS(B/W/D/Q). 3.Команда порівняння рядків CMPS(B/W/D/Q). 4.Команда сканування рядків SCAS (B/W/D/Q). 5.Команда загрузки даних з пам'яті у регістр LODS (B/W/D/Q). 6.Команда загрузки вмісту регістра у пам'ять STOS (B/W/D/Q). <p>Список рекомендованих джерел: Основний:1 [с.84-85], 2 [с.328-346]</p>	
<p>команд обробки рядків.</p>	<p>Самостійна робота студентів. Питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> 1. Які команди належать до команд обробки рядків? 2. В яких випадках використовується префікс команд та для чого він призначений? 3. Яка різниця між зчепленням та конкатенацією? 4. Яка різниця між командами CMPS та SCAS? <p>Список рекомендованих джерел: Основний:1 [с.84-85], 2 [с.328-346]</p>	6
	<p>Тема. Рядки. Лабораторне заняття №12. Лабораторна робота Команди обробки рядків. <i>Завдання до заняття:</i></p> <ol style="list-style-type: none"> 1. Відповідно до варіанта завдання програму розробити алгоритм і структуру даних програми обробки рядків, з обов'язковим виведення у спрощене вікно результатів обчислень. 2. Написати програму за розробленим алгоритмом. 3. Розробити тест для перевірки роботи програми. 4. Перевірити роботу програми з використанням тесту. 5. Оформити звіт. 	2
<p>Знати: призначення макросів і їх використання в процедурно-орієнтованому програмуванні</p> <p>Вміти: використовувати макроси при проектуванні та розробці</p>	<p>Тема 13. Макровизначення. Лекція №13 Макровизначення. План лекції.</p> <ol style="list-style-type: none"> 1.Введення в макровизначення. 2.Порівняльний аналіз процедур і макрозасобів. 3.Місця використання макровизначень. 4.Директиви MACRO, ENDM і LOCAL. 5.Булеві вирази. 6.Оператори в макровизначеннях. 7.Додаткові макровизначення і директиви. 8.Макроси циклів: FOR, FORC, REPEAT, WHILE . 9.Категорії макросів в masm32. 10.Макрос @ для запису команд в один рядок. 	2

<p>програм</p>	<p>Список рекомендованих джерел: <i>Основний: 2 [с. 353-377]</i> <i>Додатковий: 5 [с.81-84]</i></p> <p>Самостійна робота студентів. Питання для самостійного опрацювання: 1. Що називається макровизначенням? 2. Які є типи макросів? 3. Переваги та недоліки макросів в порівнянні з процедурами? 4. Для чого призначена директива LOCAL? 5. Які директиви використовуються в макровизначеннях для організації циклів? 6. Які оператори використовуються в макровизначеннях? 7. Які є категорії макросів в masm32?</p> <p>Список рекомендованих джерел: <i>Основний: 2 [с. 353-377]</i> <i>Додатковий: 5 [с.81-84]</i></p>	<p>8</p>
	<p>Тема. Макровизначення. Лабораторне заняття №13. Лабораторна робота Макровизначення. Завдання до заняття: 1. Відповідно до варіанта завдання розробити алгоритм і структуру даних програми обчислення виразу із використанням макросів та виведенням результатів обчислень на екран. 2. Написати програму за розробленим алгоритмом. 3. Розробити тест для перевірки роботи програми. 4. Перевірити роботу програми з використанням тесту. 5. Оформити звіт.</p>	<p>4</p>
<p>Знати: призначення динамічних бібліотек та їх використання при розробці програм на асемблері</p> <p>Вміти: практично складати, налагоджувати і виконувати програми з використанням DLL-файлів.</p>	<p>Тема 14. Динамічні бібліотеки. Лекція №14. Динамічні бібліотеки. План лекції. 1. Поняття динамічної бібліотеки. 2. Бібліотека DLL: створення і використання. 3. Функція точки входу в DLL.</p> <p>Список рекомендованих джерел: <i>Основний: 2 [с.227-235]</i></p>	<p>2</p>
	<p>Самостійна робота студентів. Питання для самостійного опрацювання: 1. Які переваги та недоліки використання динамічної бібліотеки? 2. Які кроки необхідно виконати для створення своєї динамічної бібліотеки? 3. Як створюється точка входу в динамічну бібліотеку? 4. Чи вимагає внесення змін в DLL перекомпіляцію всього проекту.</p> <p>Список рекомендованих джерел: <i>Основний: 2 [с.227-235]</i></p>	<p>7</p>

	<p>Тема. Динамічні бібліотеки. Лабораторне заняття №14. Динамічні бібліотеки.. <i>Завдання до заняття:</i></p> <ol style="list-style-type: none"> 1. Відповідно до варіанта завдання розробити алгоритм і структуру даних. програми обчислення виразу з використанням двох DLL-файлів з використанням точки входу. 2. Написати програму за розробленим алгоритмом. 3. Розробити тест для перевірки роботи програми. 4. Перевірити роботу програми з використанням тесту. 5. Оформити звіт. 	2
<p>Знати: функції Win32 API призначені для роботи з файлами</p> <p>Вміти: практично складати програми зі створенням файлів та запису в них інформації</p>	<p>Тема 15. Файли Лекція №15.Файли. План лекції.</p> <ol style="list-style-type: none"> 1.Створення файлу. 2.Читання файлу. 3.Отримання інформації про файл та його ідентифікатор. 4.Створення, перейменування та видалення каталогів. 5.Файлові операції засобами ShellAPI. <p>Список рекомендованих джерел: <i>Основний: 2 [с.239-263]</i> <i>Додатковий:5 [с.87-97]</i></p>	2
	<p>Самостійна робота студентів.</p> <p>Питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> 1. Яка функція Win32 API призначена для створення файлу? 2. Яка функція Win32 API призначена для запису у файл? 3. Які дії необхідно виконати для читання даних із файлу? 4. Які дії необхідно виконати для того щоб вставити дані на початок файлу? <p>Список рекомендованих джерел: <i>Основний: 2 [с.239-263]</i> <i>Додатковий:5 [с.87-97]</i></p>	6
	<p>Тема. Файли. Лабораторне заняття №15. Файли. <i>Завдання до заняття:</i></p> <ol style="list-style-type: none"> 1. Відповідно до варіанта завдання розробити алгоритм програми виведення даних у файл та в функцію MessageBox. 2. Написати програму за розробленим алгоритмом. 3. Налаштувати програму та перевірити її працездатність. 4. Оформити звіт. 	2
<p>Знати: директиви умовного асемблювання</p> <p>Вміти: практично складати</p>	<p>Тема 16. Логічні конструкції високого рівня. Лекція №16. Логічні конструкції високого рівня. План лекції.</p> <ol style="list-style-type: none"> 1.Призначення логічних конструкції високого рівня. 2.Конструкції директиви умовного асемблювання IF. 3.Логічні директиви SWITCH, SWITCH\$ та їх призначення. 4.Директиви організації циклів .REPEAT та .WHILE. 	1

<p>програми з використанням директив умовного асемблювання</p>	<p>5.Директиви переривання та виходу з циклу .BREAK та .CONTINUE.</p> <p>Список рекомендованих джерел: Основний: 2 [с.382-407]</p>	
	<p>Самостійна робота студентів. Питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> 1. Переваги конструкцій високого рівня? 2. Які конструкції належать до логічних конструкцій високого рівня? 3. За допомогою яких операторів описуються умови порівняння? 4. Які переваги та недоліки логічної конструкції порівняння текстових рядків? 5. Директиви організації циклів? 6. Яка директива використовується для переривання виконання циклу? <p>Список рекомендованих джерел: Основний: 2[с.382-407]</p>	6
	<p>Тема. Логічні конструкції високого рівня. Лабораторне заняття №16. Логічні конструкції високого рівня.. <i>Завдання до заняття:</i></p> <ol style="list-style-type: none"> 1. Відповідно до варіанта завдання розробити алгоритм програми обчислення виразу з використанням директив умовного асемблювання та дійсних чисел з обов'язковим виведенням результатів у спрощене вікно. 2. Написати програму за розробленим алгоритмом. 3. Налаштувати програму та перевірити її працездатність. 4. Оформити звіт. 	2
<p>Знати: структури, складні структури та створення зображення іконки</p> <p>Вміти: практично створювати структури, складні структури та зображення іконки</p>	<p>Тема 17. Структури. Лекція №17. Структури. План лекції.</p> <ol style="list-style-type: none"> 1.Організація структур. 2.Структура SYSTEMTIME для визначення системного часу. 3. Структура MSGBOXPARAMS в функції MessageBoxIndirect. 4.Складні структури. 5.Створення зображення іконки. <p>Список рекомендованих джерел: Основний: 2[с.413-423]</p>	1
	<p>Самостійна робота студентів. Питання для самостійного опрацювання:</p> <ol style="list-style-type: none"> 1. Що являє собою структура? 2. Які дії необхідно виконати для використання структур в програмі? 3. Які переваги та недоліки структур? 4. Як обробляти 64-розрядні числа структур? <p>Список рекомендованих джерел: Основний: 2 [с.413-423]</p>	6

	Тема. Структури. Лабораторне заняття №17. Структури. Завдання до заняття: 1. Відповідно до варіанта завдання розробити алгоритм програми обчислення виразу з використанням структур . 2. Написати програму за розробленим алгоритмом. 3. Налаштувати програму та перевірити її працездатність. 4. Оформити звіт.	2
Разом		180

*Всі заняття проводяться в інтерактивному режимі навчання.

3. СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

Основний

1. Тонкошкур О.С, Гниленко, О.Б, Матвєєва Н.О, Морозов О.С. Архітектура комп'ютерів. Машинні команди та програмування на асемблері: навчальний посібник - Дніпро: «Нова Ідеологія», 2018. - 179 с.
2. Рисований О.М. Системне програмування [Текст]: підручник для студентів напрямку «Комп'ютерна інженерія» вищих навчальних закладів в 2-х томах. Том 1. – Видання четверте: виправлено та доповнено – Харків: «Слово», 2015.– 576 с.
3. Рисований О.М. Системне програмування [Текст]: підручник для студентів напрямку «Комп'ютерна інженерія» вищих навчальних закладів в 2-х томах. Том 2. – Видання четверте: виправлено та доповнено – Харків: «Слово», 2015.– 378 с.
4. Системне програмування Графічний інтерфейс користувача (GUI) : навчальний посібник для студентів спеціальностей 123 – «Комп'ютерна інженерія», 125 – «Кібербезпека» / О.М. Рисований – Харків : «Слово», 2018 – 160 с.

Додатковий

5. Нестеренко С.О. «Системне програмування. Архітектура ПК та асемблер MASM32». Методичні вказівки для самостійної роботи з курсу «Системне програмування», для студентів спеціальності 123 «Комп'ютерна інженерія». – Чернігів: ЧНТУ, 2018. – 163 с.
6. Методичні вказівки для виконання лабораторних робіт з дисципліни «Системне програмування» для студентів денної та заочної форми навчання розроблені у відповідності з навчальним планом спеціальності 123 «Комп'ютерна інженерія» / Уклад. Паламар А.М., Паламар М.І. – Тернопіль: ТНТУ, 2020. – 70 с.

Internet-ресурси

7. MASM32 для WINDOWS <https://www.masm32.com/download.htm>.
8. Кросплатформене середовище розробки програм для мов асемблера NASM, MASM, GAS, FASM <https://wikiprograms.org/sasm/>

