

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет прикладної математики та інформатики
Кафедра програмування

Затверджено

на засіданні кафедри програмування
факультету прикладної математики
Львівського національного університету
імені Івана Франка
(протокол № 1 від 31 серпня 2021 року)



Зав. кафедри: к. ф.-м. н., доц. Ярошко С. А.

Силабус навчальної дисципліни

«Інформаційні технології та системи»,

викладається в межах ОПШ “Комп’ютерна алгебра, криптологія і теорія ігор”, “Комп’ютерний аналіз математичних моделей”, “Математика. Математична економіка та економетрика”

першого (бакалаврського) рівня вищої освіти для здобувачів зі спеціальності 111 Математика

Львів 2022

Назва дисципліни	Інформаційні технології та системи
Адреса викладання дисципліни	Львівський національний університет імені Івана Франка, вул. Університетська 1, м. Львів, Україна, 79000
Факультет та кафедра, за якою закріплена дисципліна	Факультет прикладної математики та інформатики, кафедра програмування
Галузь знань, шифр та назва спеціальності	11 – математика та статистика 111 – математика
Викладачі дисципліни	Черняхівський Володимир Вікторович, к. ф.-м. н., доц., доцент кафедри програмування
Контактна інформація викладачів	Електронна пошта: volodymyr.chernyakhivskyy@lnu.edu.ua веб-сторінка: https://ami.lnu.edu.ua/employee/chernyakhivskyy
Консультації з питань навчання по дисципліні відбуваються	Консультації проводять раз на тиждень згідно з оприлюдненим розкладом консультацій викладача. Можливі он-лайн консультації через Zoom чи Microsoft Teams. Для погодження часу он-лайн консультацій потрібно писати на електронну пошту викладача.
Сторінка курсу	https://ami.lnu.edu.ua/course/operating-systems-for-pc-informatics
Інформація про дисципліну	Курс “Інформаційні технології та системи” є вибірковою дисципліною зі спеціальності 111 – математика для освітньої програми Математика, яку викладають у п'ятому та шостому семестрі в обсязі 8 кредитів (за Європейською кредитно-трансферною системою ECTS)
Коротка анотація дисципліни	Вивчення елементів операційної системи, необхідних для оволодіння прийомами системного програмування. Формування системи знань про методи, алгоритми і прийоми програмування з використанням системних базових засобів комп'ютера. Формування практичних навичок складання системних програм шляхом моделювання реальних системних програм. Вивчення типових сучасних системних засобів програмування задач різного призначення, отримання знань і практичних навичок проектування системних програм, структур даних і побудови алгоритмічних моделей. Курс використовує низку знань і навичок, які були отримані студентами в інших раніше викладених курсах за розділами програмування і суміжних з ними.
Мета та цілі дисципліни	Метою нормативної дисципліни «Інформаційні технології та системи» є: <ul style="list-style-type: none"> ● вивчення елементів операційної системи, пов'язаних з практичним програмуванням; ● розуміння предмета дисципліни, зв'язку з іншими суміжними засобами і методами програмування, вивчення загальних принципів програмування задач на системному рівні операційної системи комп'ютера; ● вивчення математичних та інформаційних основ системного програмування, як методу реалізації робочого середовища комп'ютера; ● вміння застосовувати алгоритмічні мови асемблера, C++, C# та інші в обсязі задач розділів курсу; ● вивчення сучасних алгоритмів програмування системних задач для типових застосувань в програмних проектах; ● отримання навичок розробки процедур проектування, програмування, тестування і налагодження системних програм; ● вміння використовувати стандартні базові бібліотеки операційної системи комп'ютера і середовищ програмування різних алгоритмічних мов.

<p>Література для вивчення дисципліни</p>	<p><i>Основна література</i></p> <ol style="list-style-type: none"> 1. Шеховцов В.А. Операційні системи: підручник / В.А.Шеховцов // К.: Видавничка група ВНУ. – 2005. – 576с. – Режим доступу: http://www.dut.edu.ua/ua/lib/1/category/729/view/1382 2. Грайворонський М.В. Операційні системи. Конспект лекцій [Електронний ресурс] // НТУУ "КПІ" Фізико-Технічний Інститут. Кафедра інформаційної безпеки. – Режим доступу: http://is.ipt.kpi.ua/operatsijni-sistemi http://is.ipt.kpi.ua/wp-content/uploads/sites/4/2015/04/Conspect_OS.pdf 3. Галісеєв Г. Системне програмування / Геннадій Галісеєв // Університет "Україна". – 2019. – 113с. 4. Дерев'янку О.С. Системне програмування. Системні сервісні компоненти [Електронний ресурс]: навч. посібник / О.С.Дерев'янку, С.Г.Межерицький, С.Ю.Гавриленко, А.М.Клименко // Харків. – НТУ "ХПІ". – 2009. – 160с. – Режим доступу: http://khpi-iip.mipk.kharkiv.edu/library/sp/sp_book/index.html#1 5. Портал знань. Асемблер (assembler) і системне програмування [Електронний ресурс]. – Режим доступу: http://www.znannya.org/?view=asm 6. Microsoft Ignite. What is a DLL [Електронний ресурс]. – Режим доступу: https://support.microsoft.com/uk-ua/help/815065/what-is-a-dll 7. Вікіпедія. Inkscape — вільний редактор векторної графіки [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Inkscape 8. Вікіпедія. Інтерпретатор мови програмування [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Інтерпретатор 9. Microsoft Ignite. Get Started with Win32 and C++ [Електронний ресурс]. – Режим доступу: https://docs.microsoft.com/en-us/windows/win32/learnwin32/learn-to-program-for-windows <p><i>Додаткова література</i></p> <ol style="list-style-type: none"> 10. Інтерпретатор [Електронний ресурс]. – Режим доступу: http://kytok.org.ua/?p=732 11. Microsoft Ignite. Visual Studio documentation [Електронний ресурс]. – Режим доступу: https://docs.microsoft.com/en-us/visualstudio/windows/?f1url=%3FappId%3DDev15IDEF1%261%3Den-US%26k%3Dk(MSDNSTART)%26rd%3Dtrue&view=vs-2019
<p>Обсяг курсу</p>	<p>8 кредитів ЄКТС – 240 годин. З них 48 години лекцій, 64 години лабораторних занять та 128 годин самостійної роботи.</p>
<p>Очікувані результати навчання</p>	<p>Після завершення цього курсу студент буде:</p> <p><i>знати</i></p> <ul style="list-style-type: none"> • принципи будови і функціонування сучасних операційних систем; • математичні та інформаційні основи системного програмування, як методу реалізації робочого середовища комп'ютера; • засоби алгоритмічних мов асемблера, C++, C# та інших в обсязі задач розділів курсу; • типові сучасні алгоритми програмування основних системних задач; • процедури проектування, програмування, тестування і налагодження системних програм; • стандартні базові бібліотеки операційної системи комп'ютера і середовищ програмування різних алгоритмічних мов. <p><i>вміти</i></p> <ul style="list-style-type: none"> • застосовувати засоби різних алгоритмічних мов для програмування системних задач; • складати, тестувати і налагоджувати системні програми загального і прикладного характеру; • аналізувати і модифікувати системні програми до зміни вимог; • реалізувати вимоги користувачів до будови робочого середовища; • застосовувати стандартні бібліотеки функцій операційної системи.
<p>Ключові слова</p>	<p>Операційна система, системне програмування, мікропроцесор, асемблер, компонування, зв'язування, об'єктний файл, параметри функцій, DLL, бібліотека функцій, комп'ютерна графіка, графічна операція, графічний</p>

	редактор, електронна таблиця, сканер, лексичний аналіз, семантичний аналіз, інтерпретатор.
Формат курсу	Очний: проведення лекцій, лабораторних робіт та консультацій в приміщеннях університету, а в умовах карантину – онлайнний на платформі Microsoft Teams.
Теми	
Підсумковий контроль, форма	Залік в кінці семестру
Пререквізити	Для вивчення курсу студенти потребують базових знань з дисциплін: дискретна математика, архітектура обчислювальних систем, програмування, алгоритми і структури даних, програмна інженерія.
Навчальні методи та техніки, які використовують під час викладання курсу	Лекції з мультимедійними презентаціями та з демонстрацією прийомів програмування системних засобів; лабораторні заняття у вигляді проектування алгоритмів і програм, виконання практичних завдань та проектів; самостійне опрацювання навчальних матеріалів: підручників, конспектів лекцій, електронних ресурсів, готових програм різними алгоритмічними мовами, додаткових матеріалів, розміщених у хмарному сховищі (Microsoft Teams, Google Диск). Обговорення теоретичного та практичного матеріалу в онлайн сервісах, формулювання творчих завдань для студентів, виконання яких готує до вивчення нового теоретичного і практичного матеріалу.
Необхідне обладнання	Для проведення лекцій: комп'ютер, проектор, доступ до мережі інтернет. Для проведення лабораторних та виконання завдань: комп'ютер; ОС Windows/Linux; доступ до інтернету; середовища програмування мовою C++ (Microsoft Visual Studio, Code Blocks тощо), мовою C#, мовою Python (IDLE Python тощо), мовою асемблера, мовою Java. Можливі використання інших мов програмування. Уся література і робочі матеріали, які студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	Оцінювання проводиться за 100-бальною шкалою. 80 балів нараховують за виконання лабораторних завдань та індивідуальних проектів впродовж семестра, ще 20 балів – за виконання індивідуального тестового завдання. Лабораторні завдання індивідуальні. Проекти можуть бути індивідуальні та командні. Упродовж семестру студент виконує до 12 лабораторних робіт і до 4 проектів (у власній комбінації, лабораторні/проекти є альтернативними), які оцінюють різною шкалою залежно від складності (шкала оцінювання і критерії надаються студентам на початку семестра). Оцінка за екзаменаційне завдання може отримати додаткові бали, якщо проекти студента, виконані впродовж семестра, мали закінчений характер і високий рівень фахової реалізації. Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Активність під час проведення лекцій і лабораторних заохочується балами при оцінюванні відповідного лабораторного завдання чи проекту. У будь-якому випадку студенти зобов'язані дотримуватися усіх строків визначених для виконання усіх видів робіт, передбачених курсом. Виконані роботи завантажують у відповідне хмарне сховище. Альтернативою відвідування лабораторних занять в університеті може бути дистанційна онлайн робота за розкладом проведення занять. Академічна доброчесність: очікується, що роботи студентів будуть їхнім оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів, здавання чужих комп'ютерних програм як своїх становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману.
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано після завершення курсу.

Схема курсу

Тиж-день	Тема, план, короткі тези	Форма заняття	Тривалість год	Термін виконання
1	Будова асемблерних програм. Схема трансляції, компанування і виконання програми. Основні реєстри мікропроцесора. Позиційна незалежність програми в однопрограмих ОС. Режими адресування (однопрограмих ОС). Моделі адресування. Моделі структури програм.	Лекція	2	
	Віконна програма перегляду і редагування текстових файлів. Файли Unicode і ASCII. Таблиці кодування символів.	Лабораторна робота	4	Наступне лабораторне заняття
2	Бітова структура команд мікропроцесора. Загальний формат команди процесора Intel. Повна форма адресування операндів в пам'яті. Приклади трансляції команд в 32-бітовому режимі. Особливості адресування. Таблиці кодів команд та їх швидкодії.	Лекція	2	
	Програмний аналіз атрибутів файлів. Програмування сервісних засобів опрацювання файлів. Консольні і віконні реалізації програм. Процедури програмування, тестування і випуску програмного продукту.	Лабораторна робота	4	Наступне лабораторне заняття
3	Трансляція і завантаження програм. Схеми виконання команд мікропроцесора. Комбінації операндів в командах. Форми запису констант в командах мовою асемблера. Директиви асемблера. Трансляція програм за першим і другим переглядом.	Лекція	2	
	Модель СОМ-програми. Зовнішня модель реалізації асемблера. Трансляція і виконання асемблерної програми. Перевірка моделі виконання СОМ-програми.	Лабораторна робота	4	Наступне лабораторне заняття
4	Загальні принципи компонування програм. Об'єктні файли. Компонувальники і принципи їх роботи. Завантаження виконуваних файлів за статичного компонування.	Лекція	2	
	Трансляція ASM-команд. Схема трансляції. Підготовка даних до програмної реалізації. Визначення окремих асемблерних команд. Приклади для тестування асемблера.	Лабораторна робота	4	Наступне лабораторне заняття
5	Динамічні бібліотеки. Загальні принципи будови. Переваги і недоліки використання динамічних бібліотек. Зворотня сумісність динамічних бібліотек. Неявне і явне зв'язування. Динамічні бібліотеки та адресний простір процесу. Особливості об'єктного коду динамічних бібліотек. Структура виконуваних файлів. Виконувані файли в Linux. Формат ELF. Виконувані файли у Windows. Формат PE. Динамічне компонування у Windows.	Лекція	2	
	Створення DLL. Будова бібліотеки DLL мовою С++. Проектування, програмування і тестування функцій DLL. Параметри функцій: типи Windows і стандартні типи С++. Компіляція функцій в бібліотеку DLL.	Лабораторна робота	4	Наступне лабораторне заняття
6	Створення і використання функцій DLL. Стандартні типи даних Windows. Підготовка функцій для DLL. Компіляція функцій в бібліотеку DLL (створення DLL). Використання DLL. Неявне зв'язування, явне зв'язування.	Лекція	2	
	Використання бібліотеки DLL в програмах С++ методами неявного і явного зв'язування. Файл імпорту, файл бібліотеки. Тестування прикладної програми, яка використовує DLL.	Лабораторна робота	4	Наступне лабораторне заняття
7	Методи застосування DLL. Загальна оцінка типів даних функцій DLL. Функції DLL в програмах С#. Функції високої швидкодії і точності. Зовнішні функції в Python. Типи параметрів зовнішніх функцій Python. Функції DLL в програмах мовою асемблера. Явне і неявне зв'язування з DLL мовою асемблера.	Лекція	2	
	Використання DLL мовами програмування. Функції DLL в програмах С# (неявне зв'язування з бібліотеками DLL). Функції DLL в програмах компільованих алгоритмічних мов. Ефективне використання DLL мовами інтерпретованого типу (Python, Visual Basic, Java).	Лабораторна робота	4	Наступне лабораторне заняття
8	Принципи програмування графіки в операційній системі. Відображення малюнків на екрані дисплея. Векторна і растрова графіка. Спрощена масштабована векторна графіка в JavaScript. Формати графічних файлів. Перетворення графічних форматів.	Лекція	2	

	Базові елементи програмування графіки. Підготовка основи для програмування графічних операцій. Стандартні класи бібліотеки, які забезпечують програмне малювання. Методи реалізації графічних операцій: векторна графіка, растрова графіка. Реалізація дигамічного малювання окремих фігур.	Лабораторна робота	4	Наступне лабораторне заняття
9	Алгоритми виконання графічних операцій. Параметри стандартного графічного інструментарія. Схема будови програмованих класів малювання. Приклад будови активних елементів на час виконання.	Лекція	4	
	Алгоритми і програмна реалізація окремих графічних операцій – параметри і сервісні можливості. Скасування операцій. Копіювання і пересування окремих фігур чи ділянок малюнка.	Лабораторна робота	4	Наступне лабораторне заняття
10	Методи і технології програмування графічних операцій. Інтерфейсна частина проєкта. Меню вікон верхнього рівня. Вікна з меню і з панеллю інструментів. Підхід до функцій опрацювання графіки. Графіка на основі пакетів бібліотечних функцій типових технологій (PIL, Pillow, numpy, інші).	Лекція	6	
	Додаткові графічні операції. Перегляд і редагування великих за розмірами малюнків. Програмування операцій фільтрування. Програмування операцій обертання малюнка.	Лабораторна робота	6	Наступне лабораторне заняття
11	Обчислення формул інтерпретацією. Різновиди інтерпретаторів. Проміжний код. Правила будови синтаксичних визначень. Інтерпретація формул загальних алгебраїчних правил. Приклад програмної реалізації. Розширення змісту формул. Операторне виконання.	Лекція	6	
	Модель інтерпретатора формул для операцій без рангів. Доповнення інтерпретатора бінарними і унарними операціями. Вибір системи числення. Опрацювання помилок в формулах.	Лабораторна робота	6	Наступне лабораторне заняття
12	Побудова сканера для формул загального виду. Структури даних сканера. Граматичні правила для чисел і формул. Лексеми. Формати лексем і алгоритми розпізнавання. Визначення класу сканера.	Лекція	4	
	Модель інтерпретатора формул за алгебраїчними правилами. Бінарні і унарні операції. Граматика виразів. Метод рекурсивного спуску для граматичного розбору. Розширення змісту формул. Доповнення інтерпретатора.	Лабораторна робота	4	Наступне лабораторне заняття
13	Перетворення виразів з інфіксної форми в постфіксу. Граматичні правила для формул. Програмування перетворювача.	Лекція	6	
	Програмування і застосування сканера формул формату електронної таблиці. Формати лексем і алгоритми розпізнавання. Діаграми станів і скінчені автомати. Перетворення в постфіксу форму. Обчислювач формул.	Лабораторна робота	6	Наступне лабораторне заняття
14	Програмні проєкти за розділами системного програмування. Вибір алгоритмічної мови і середовища реалізації. Проєктування і програмування з використанням стандартних класів і бібліотек. Інкапсульовані стандартні функції операційної системи. Параметри оцінки проєктів.	Лекція	6	
	Аналіз проєктів, виконаних за курсом ОСіСП. Виправлення і доопрацювання окремих функцій. Остаточне налагодження інтерфейсних частин проєктів. Підготовка до іспиту.	Лабораторна робота	6	Початок сесії