

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
«ПРИКАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВАСИЛЯ СТЕФАНІКА»**

Фізико-технічний факультет  
Кафедра комп'ютерної інженерії та електроніки

**СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
Системне програмування**

Освітня програма Бакалавр  
Галузь знань 12 Інформаційні технології  
Спеціальність 123 Комп'ютерна інженерія

Затверджено на засіданні кафедри  
Протокол № 12 від “30” червня 2023 р.

Івано-Франківськ – 2023 рік

## ЗМІСТ

1. Загальна інформація
2. Анотація до курсу
3. Мета та цілі курсу
4. Компетентності
5. Результати навчання
6. Організація навчання курсу
7. Система оцінювання курсу
8. Політика курсу
9. Рекомендована література

## 1. Загальна інформація про викладача і дисципліну

<b>Назва дисципліни</b>	Системне програмування
<b>Рівень вищої освіти</b>	Перший рівень вищої освіти
<b>Викладач</b>	Доцент, кандидат технічних наук Голота Віктор Іванович
<b>Контактний телефон викладача</b>	(0342) 59-60-07
<b>Е-mail викладача</b>	victor.holota@pnu.edu.ua
<b>Формат дисципліни</b>	Двосеместровий
<b>Обсяг дисципліни</b>	6 кредитів
<b>Посилання на сайт дистанційного навчання</b>	<a href="http://www.d-learn.pu.if.ua">http://www.d-learn.pu.if.ua</a>
<b>Консультації</b>	Відповідно до графіку індивідуальних консультацій, який розміщений на інформаційному стенді кафедри комп'ютерної інженерії та електроніки

## 2. Анотація до курсу

Дисципліна “Системне програмування” належить до переліку обов’язкових навчальних дисциплін підготовки за освітнім рівнем «бакалавр», що пропонуються в рамках циклу професійної підготовки студентів за освітньо-професійною програмою “Комп’ютерна інженерія”. Вона забезпечує формування у студентів загальних і спеціальних (професійно-орієнтованих) компетенцій. Предметом вивчення навчальної дисципліни є асемблер Intel x86-64 та принципи організації і побудови трансляторів та інтерпретаторів.

Силабус навчальної дисципліни “Системне програмування” складений відповідно до освітньо-професійної програми підготовки бакалавра спеціальності 123 “Комп’ютерна інженерія”.

## 3. Мета та цілі курсу

**Мета:** сформувати у студентів сучасні теоретичні уявлення та практичні знання із програмування на асемблері Intel x86-32/64 та загальних принципів організації і побудови трансляторів.

**Завдання:**

- знайомство з сучасним станом, тенденціями та перспективами програмування на асемблері Intel x86-64;
- вивчення системи команд асемблера Intel x86-64 для роботи з цілочисловим процесором та процесором з плаваючою крапкою;
- вивчення принципів організації і побудови багатомодульних системних програм на мові асемблера;
- вивчення загальних принципів організації і побудови трансляторів.
- вивчення принципів лексичного, синтаксичного і семантичного аналізу, генерації машинного коду.

У результаті вивчення навчальної дисципліни студент повинен

**знати:**

- основні принципи розроблення системних програм;
- програмну архітектуру процесора Intel x86-64;
- систему команд цілочислового процесора та процесора з плаваючою крапкою;
- системи команд MMX і XMM.
- визначення і властивості скінченних автоматів;
- методи проектування лексичних аналізаторів;
- принципи низхідного і висхідного синтаксичного аналізу;
- способи подання семантики;
- методи генерації машинного коду.

**вміти:**

- створювати системні програми на асемблері в ОС Linux;
- розрізняти лексичні, синтаксичні і семантичні помилки в програмах;
- проектувати лексичні аналізатори на основі скінченних автоматів;
- будувати формальні граматики;
- побудувати дерево синтаксичного розбору;
- проектувати синтаксичний аналізатор на основі рекурсивного спуску.

**4. Компетентності****Загальні**

- ЗК2. Здатність вчитися і засвоювати сучасні знання.

**Спеціальні (фахові)**

- Р2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.
- Р3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.
- Р9. Здатність системно адмініструвати, використовувати, адаптувати та експлуатувати наявні інформаційні технології та системи.
- *Здатність розробляти системні програми на мові асемблера Intel x86-32.64.*
- *Здатність проектувати лексичні аналізатори на основі скінченних автоматів.*
- *Здатність проектувати синтаксичні аналізатори на основі рекурсивного спуску.*
- *Здатність працювати з лексичними і синтаксичними генераторами.*

**5. Результати навчання**

- N10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати типове для спеціальності обладнання.

**6. Організація навчання курсу**

Обсяг курсу	
Вид заняття	Загальна кількість годин
лекції	42

семінарські заняття / практичні / лабораторні		48			
самостійна робота		90			
<b>Ознаки курсу</b>					
Семестр	Спеціальність	Курс (рік навчання)		Нормативний/ вибірковий	
5, 6	123 Комп'ютерна інженерія	3		нормативний	
<b>Тематика курсу</b>					
Тема	Форма заняття, год.	Література	Кількість годин	Вага оцінки	Термін виконання
<b>5-й семестр</b>					
<b>Змістовий модуль 1. Основи асемблера NASM</b>					
Тема 1. Вступ в системне програмування.	лекція	1-7, 9-11	2	3	Згідно розкладу
Тема 2. Режими роботи і моделі пам'яті процесорів intel x86-32/64.	лекція	1-7, 9-11	2	3	Згідно розкладу
Тема 3. Основи Nasm асемблера.	лекція	1-7, 9-11	2	3	Згідно розкладу
Тема 4. Nasm макропроцесор і макродирективи.	лекція	1-7, 9-11	2	3	Згідно розкладу
Тема 5. Класифікація цілочислових команд. Команди пересилання даних, арифметичні, логічні, ланцюгові.	лекція	1-7, 9-11	2	3	Згідно розкладу
Модульний контроль 1			2		
<b>Практичний модуль</b>					
Тема 6. Команди передачі керування.	лекція	1-7, 9-11	2	3	Згідно розкладу
Тема 7. Процеси, системні виклики і підпрограми.	лекція	1-7, 9-11	2	3	Згідно розкладу
Тема 8. Співпроцесор.	лекція	1-7, 9-11	2	3	Згідно розкладу
Тема 9. Технологія MMX.	лекція	1-7, 9-11	2	3	Згідно розкладу
Тема 10. Технологія SSE.	лекція	1-7, 9-11	2	3	Згідно розкладу
Тема 11. Технологія AVX.	лекція	1-7, 9-11		2	Згідно розкладу
Модульний контроль 2			2		

1. Програмна модель процесора, типи даних, директиви і адресація пам'яті	лаб. робота	1-7, 9-11	2	4	Згідно розкладу
2. Синтаксис і формати команд асемблерів NASM і GAS.	лаб. робота	1-7, 9-11	2	4	Згідно розкладу
3. Команди mov, lea, цілочисельні арифметичні.	лаб. робота	1-7, 9-11	2	4	Згідно розкладу
4. Безумовні та умовні переходи, цикли, логічні побітові команди і операції над виразами, зсуви, деасемблери.	лаб. робота	1-7, 9-11	2	4	Згідно розкладу
5. Ланцюгові команди.	лаб. робота	1-7, 9-11	2	4	Згідно розкладу
6. Директиви і макроси препроцесора.	лаб. робота	1-7, 9-11	2	4	Згідно розкладу
7. Виклики підпрограм.	лаб. робота	1-7, 9-11	2	4	Згідно розкладу
8. Взаємодія з ОС. Переривання.	лаб. робота	1-7, 9-11	2	4	Згідно розкладу
9. Структури даних.	лаб. робота	1-7, 9-11	2	4	Згідно розкладу
10. Модульне програмування.	лаб. робота	1-7, 9-11	2	4	Згідно розкладу
11. Співпроцесор.	лаб. робота	1-7, 9-11	2	4	Згідно розкладу
<b>Самостійна робота</b>					
Тема 1. Макроси.	самостій на робота	1-11	6	3	Впродовж семестру
Тема 2. Введення даних з консолі	самостій на робота	1-11	6	3	Впродовж семестру
Тема 3. Робота з файлами.	самостій на робота	1-11	6	3	Впродовж семестру
Тема 4. Команди налагоджувача GDB.	самостій на робота	1-11	6	3	Впродовж семестру
Тема 5. Налгоджувач з графічним інтерфейсом DDD	самостій на робота	1-11	6	3	Впродовж семестру

Тема 6. Інтерфейси з мовами високого рівня.	самостійна робота	1-11	6	3	Впродовж семестру
Тема 7. Команди MMX, SSE розширення процесорів x86_64.	самостійна робота	1-11	6	3	Впродовж семестру
Тема 8. Команди SSE2 процесорів x86_64.	самостійна робота	1-11	4	3	Впродовж семестру
Контроль самостійної роботи			2		Згідно розкладу
<b>Підсумковий контроль (залік)</b>				100	Згідно розкладу
6-й семестр					
<b>Змістовий модуль 3. Лексичний і синтаксичний аналіз</b>					
Тема 1. Лексичний аналіз. Специфікація і розпізнання токенів.	лекція	12-22	2	2	Згідно розкладу
Тема 2. Лексичний аналіз. Недетерміновані і детерміновані скінченні автомати.	лекція	12-22	2	2	Згідно розкладу
Тема 3. Синтаксичний аналіз. Контекстно вільні граматики.	лекція	12-22	2	2	Згідно розкладу
Тема 4. Синтаксичний аналіз. Метод рекурсивного спуску.	лекція	12-22	2	2	Згідно розкладу
Тема 5. Синтаксичний аналіз. Висхідний синтаксичний аналіз.		12-22	2	2	
Модульний контроль			2		Згідно розкладу
<b>Змістовий модуль 4. Трансляція і генерація машинного коду</b>					
Тема 6. Синтаксично керована трансляція.	лекція	12-22	2	2	Згідно розкладу
Тема 7. Генерація проміжного коду.	лекція	12-22	2	2	Згідно розкладу
Тема 8. Середовище часу виконання.	лекція	12-22	2	2	Згідно розкладу
Тема 9. Оптимізація і генерація машинного коду.	лекція	12-22	2	2	Згідно розкладу
Тема 10. Методи і засоби генерації машинного коду.	лекція	12-22	2	2	Згідно розкладу
Модульний контроль			2		Згідно

					розкладу
<b>Практичний модуль</b>					
1. Регулярні вирази.	лаб. робота	12-27	2	2	Згідно розкладу
2. Пошук ключових слів у стрічках.	лаб. робота	12-27	2	2	Згідно розкладу
3. Недетермінований і детермінований скінченні автомати.	лаб. робота	12-27	2	2	Згідно розкладу
4. Побудова лексичного аналізатора на мові C.	лаб. робота	12-27	2	2	Згідно розкладу
5. Побудова лексичного аналізатора на мові Python.	лаб. робота	12-27	2	2	Згідно розкладу
6. Побудова лексичного аналізатора з використанням FLEX і PLY.	лаб. робота	12-27	2	2	Згідно розкладу
7. Трансляція арифметичних виразів з інфіксною у постфіксну/префіксну нотацію.	лаб. робота	12-27	2	2	Згідно розкладу
8. Синтаксичний аналізатор простих арифметичних виразів.	лаб. робота	12-27	2	2	Згідно розкладу
9. Низхідний рекурсивний синтаксичний аналізатор.	лаб. робота	12-27	2	2	Згідно розкладу
10. Предиктивний синтаксичний аналізатор LL(1).	лаб. робота	12-27	2	2	Згідно розкладу
11. Побудова таблиці предиктивного синтаксичного аналізу.	лаб. робота	12-27	2	2	Згідно розкладу
12. Побудова синтаксичного аналізатора з використанням генераторів	лаб. робота	12-27	2	1	Згідно розкладу
13. Генерація машинного коду з використанням генераторів.	лаб. робота	12-27		1	Згідно розкладу
<b>Самостійна робота</b>					
Тема 1. Мінімізація кількості станів детермінованого скінченного автомата	самостій на робота	12-27	7	1	Впродовж семестру
Тема 2. Мови і контекстно вільні граматики	самостій на	12-27	7	1	Впродовж семестру



	робота				
Тема 3. Розроблення граматики мови програмування.	самостійна робота	12-27	7	1	Впродовж семестру
Тема 4. Функції FIRST і FOLLOW в низхідному синтаксичному аналізі.	самостійна робота	12-27	7	1	Впродовж семестру
Тема 5. LR синтаксичні аналізатори.	самостійна робота	12-27	7	1	Впродовж семестру
Тема 6. Генерація проміжкового і цільового машинного коду.	самостійна робота	12-27	8	1	Впродовж семестру
Контроль самостійної роботи			2		Згідно розкладу
<b>Підсумковий контроль (екзамен)</b>				50	Згідно розкладу

## 7. Система оцінювання курсу

<b>Загальна система оцінювання курсу</b>
<p><i>Поточний контроль</i> здійснюється під час проведення лекційних, лабораторних, індивідуальних занять і має на меті перевірку знань студентів з окремих тем навчальної дисципліни та рівня їх підготовленості до виконання конкретної роботи. Оцінки у національній шкалі («відмінно» - 5, «добре» - 4, «задовільно» - 3, «незадовільно» - 2), отримані студентами, виставляються у журналах обліку відвідування та успішності академічної групи.</p> <p><i>Модульний контроль</i> (сума балів за окремий змістовий модуль) проводиться (виставляється) на підставі оцінювання результатів знань студентів після вивчення матеріалу з логічно завершеної частини дисципліни – змістового модуля.</p> <p>Завданням модульного контролю є перевірка розуміння та засвоєння певного матеріалу (теми), вироблення навичок проведення розрахункових робіт, вміння вирішувати конкретні ситуативні задачі, самостійно опрацьовувати тексти, здатності осмислювати зміст даної частини дисципліни, уміння публічно чи письмово подати певний матеріал.</p> <p><i>Семестровий (підсумковий) контроль</i> проводиться у формі заліку (5-й семестр), екзамену (6-й семестр).</p> <p><i>Екзамен</i> – форма підсумкового контролю, яка передбачає перевірку розуміння студентом теоретичного та практичного програмного матеріалу з усієї дисципліни, здатності творчо використовувати здобуті знання та вміння, формувати власне ставлення до певної проблеми тощо.</p> <p><i>Залік</i> – форма підсумкового контролю, що полягає в оцінюванні засвоєння здобувачем навчального матеріалу з певної дисципліни, і складається із зданих залікових змістових модулів, виконаних тестових завдань, ситуаційних робіт, опрацювання завдань робочих зошитів, тематичних рефератів, лабораторних робіт тощо, передбачених навчальною програмою.</p>

<b>Шкала оцінювання: національна та ECTS</b>			
Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проєкту (роботи), практики	для заліку
90-100	A	відмінно	зараховано
80-89	B	добре	
70-79	C		
60-69	D	задовільно	
50-59	E		
26-49	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-25	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни
<b>Вимоги до письмової роботи</b>			
Підсумкова письмова роботи виконується з використанням білетів. Кількість білетів 25. Кількість запитань у білеті 3.			
<b>Лабораторні заняття</b>			
До початку лабораторної роботи студент має отримати допуск на основі усної співбесіди з викладачем. На лабораторній роботі кожен студент отримує варіант завдання до виконання на занятті і домашнього завдання для самостійної роботи. Після завершення роботи студент захищає поточне завдання і домашні завдання попередніх лабораторних робіт оформленні як звіти.			
<b>Умови допуску до підсумкового контролю</b>			
Студент допускається до складання екзамену, якщо впродовж семестру він за змістові модулі набрав сумарно 25 балів і вище.			
Студент не допускається до складання екзамену, якщо впродовж семестру він за змістові модулі набрав менше 25 балів. У цьому випадку студенту у відомості робиться запис “не допущений” і виставляється набрана кількість балів. Допускається, як виняток, з дозволу декана факультету за заявою, погодженою з відповідною кафедрою, одноразове виконання студентом додаткових видів робіт з навчальної дисципліни (відпрацювання пропущених занять, перескладання змістових модулів, виконання індивідуальних завдань тощо) для підвищення оцінок за змістові модулі.			
Напередодні екзамену викладач подає доповідну декану про не допуск студентів академічної групи (груп). Відмітка про не допуск у відомості робиться при наявності розпорядження декана.			

## 8. Політика курсу

Студент зобов'язаний відвідувати заняття відповідно до встановленого розкладу, не запізнюватися, мати відповідний зовнішній вигляд. У разі відсутності через хворобу надається відповідна довідка.

Пропущена лекція відпрацьовується студентом самостійно і оформляється як короткий конспект за темою заняття.

Пропущена лабораторна робота виконується студентом самостійно вдома або в комп'ютерному класі, результати оцінюються викладачем.

У випадку, коли студент приймав участь у програмі мобільності, можливе зарахування отриманих оцінок в іншому навчальному закладі за умови відповідності навчальних планів.

Можливе зарахування результатів неформальної освіти згідно з Положенням про порядок зарахування неформальної освіти у ДВНЗ «Прикарпатський національний університет імені Василя Стефаника».

### Політика академічної поведінки і етики

Студент повинен бути толерантним і поважати думку інших.

Заперечення повинні формулюватися тільки в коректній формі.

Академічна доброчесність здобувачами вищої освіти передбачає:

- самостійне виконання навчальних завдань, завдань поточного та підсумкового контролю результатів навчання;
- посилення на джерела інформації у разі використання ідей, розробок, тверджень, відомостей;
- дотримання норм законодавства про авторське право і суміжні права;
- надання достовірної інформації про результати власної навчальної (наукової, творчої) діяльності, використані методики досліджень і джерелі інформації.

Не допускається підказування та списування під час здачі будь-яких робіт поточного, рубіжного, підсумкового контролю.

Не допускається користування телефонами та будь-якими іншими електронними засобами під час здачі будь-яких робіт поточного, рубіжного, чи підсумкового контролю.

## 9 Рекомендована література

### Основна (1-й семестр)

1. Randall Hyde. The Art of 64-Bit Assembly, Volume 1: x86-64 Machine Organization and Programming. No Starch Press, 2021. – 1032 p.
2. Daniel Kusswurm. Modern X86 Assembly Language Programming: Covers x86 64-bit, AVX, AVX2, and AVX-512. Apress, 2018. – 625 p.
3. Jo Van Hoey. Beginning x64 Assembly Programming: From Novice to AVX Professional. Apress, 2019. – 438 p.
4. Jo Van Hoey. Beginning x64 Assembly Programming: From Novice to Professional. CA: Apress, 2019. – 413 p.
5. Jeff Duntemann. Assembly Language Step-by-Step: Programming with Linux 3rd Edition. Wiley, 2009. – 656 p.

### Додаткова

6. NASM [Інтернет ресурс]: NASM on-line компілятор. Режим доступу: <https://www.jdoodle.com>
7. NASM [Інтернет ресурс]: Сайт розробників NASM асемблера. Режим доступу: <https://www.nasm.us>.
8. YASM [Інтернет ресурс]: Сайт розробників YASM асемблера. Режим доступу: <http://www.tortall.net/>
9. Flat Assembler (FASM) [Інтернет ресурс]: Сайт розробників FASM асемблера. Режим доступу: <http://www.flatassembler.net/>
10. “Calling Convention” [Інтернет ресурс]: Домовленість про виклики підпрограм. Режим доступу: <https://learn.microsoft.com/en-us/docs/>
11. Intel [Інтернет ресурс]: Технічна документація для розробників. Режим доступу: <https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html>

### Основна (2-й семестр)

12. Основи проектування трансляторів: Інструкції та завдання до виконання лабораторних робіт та розрахунково-графічної роботи з дисципліни «Основи проектування трансляторів» : [Електронний ресурс] : навч. посіб. для студ. спеціальності 123 – «Комп’ютерна інженерія» / О. І. Марченко, О. О. Марченко ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 1,08 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 118 с.
13. Формальні мови, граматики та автомати: Навчальний посібник/ Гавриленко С.Ю. – Харків: НТУ «ХПІ», 2021. – 133 с.
14. Clinton L. Jeffery. Build Your Own Programming Language: A programmer's guide to designing compilers, interpreters, and DSLs for solving modern computing problems. Packt Publishing, 2021. – 494 p.
15. Douglas Thain. Introduction to Compilers and Language Design: Second Edition. Independently published, 2020. – 247 p.
16. Anthony J. Dos Reis. Writing Interpreters and Compilers for the Raspberry Pi Using Python: Second Edition. Independently published, 2020. – 247 p.
17. Watt, David A. Programming language processor in Java: compilers and interpreters / Davit A. Watt and Deryck F. Brown. GB: Pearson EL, 2000. – 424 p.
18. Alfred Aho, Jeffrey Ullman, Ravi Sethi, Monica Lam. Compilers: Principles, Techniques, and Tools 2nd Edition. Addison-Wesley, 2007. – 1040 p.

### Додаткова

19. Keith D. Cooper and Linda Torczon. Engineering a Compiler. 2012 Elsevier, Inc., 2012. – 800 p.
20. Andrew W. Appel, Maia Ginsburg. Modern Compiler implementation in C. Cambridge University press, 1998. – 544 с.
21. Allen I. Holub. Compiler design in C. Prentice-Hall, Inc. , 1990. – 984 p.

22. Jean-Paul Tremblay, Paul G. Sorenson. The theory and practice of compiler writing. McGraw-Hill Book Company, 1985. – 796 p.

23. Flex [Інтернет ресурс]: Генератор лексичних аналізаторів. Режим доступу: <https://sourceforge.net/p/re-flex/wiki/Home/>

24. GNU Bison [Інтернет ресурс]: Генератор парсерів загального призначення. Режим доступу: <https://www.gnu.org/software/bison/>

25. PLY (Python Lex-Yacc) [Інтернет ресурс]: Лексичний і синтаксичний аналізатор на мові Python. Режим доступу: <https://www.dabeaz.com/py>

26. ANTLR [Інтернет ресурс]: Потужний генератор парсерів. Режим доступу: <https://wwwantlr.org/>

27. LLVM [Інтернет ресурс]: Оптимізатор і генератор коду. Режим доступу: <https://llvm.org/>

Викладач



Голота В.І.