

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

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 № від р.

Івано-Франківськ – 2020 рік

## ЗМІСТ

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

## 1. Загальна інформація про викладача і дисципліну

Назва дисципліни	Системне програмування
Рівень вищої освіти	Перший рівень вищої освіти
Викладач	Доцент, кандидат технічних наук Голота Віктор Іванович
Контактний телефон викладача	(0342) 59-60-07
Е-mail викладача	victor.holota@pnu.edu.ua
Формат дисципліни	Двосеместровий
Обсяг дисципліни	6 кредитів
Посилання на сайт дистанційного навчання	<a href="http://www.d-learn.pu.if.ua">http://www.d-learn.pu.if.ua</a>
Консультації	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 x86-32/64 та принципи організації і побудови трансляторів.

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 x86-32/64;
- вивчення системи команд асемблера Intel x86-32/64 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 x86-32/64;
- систему команд цілочислового процесора та процесора з плаваючою крапкою;
- системи команд MMX і XMM.
- визначення і властивості скінчених 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 скінчених автоматів;
- будувати формальні граматики;
- будувати дерево синтаксичного розбору;
- проектувати синтаксичний аналізатор на основі рекурсивного спуску.

#### 4. Результати навчання (компетентності)

**Інтегральна**

- І. Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми під час професійної діяльності в комп'ютерній галузі або навчання, що передбачає застосування теорій та методів комп'ютерної інженерії і характеризується комплексністю та невизначеністю умов.

**Загальні**

- ЗК1. Здатність до абстрактного мислення, аналізу і синтезу.
- ЗК3. Здатність застосовувати знання у практичних ситуаціях.
- ЗК7. Вміння виявляти, ставити та вирішувати проблеми.

**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.
- Р7. Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності.
- *Здатність розробляти системні програми на мові асемблера Intel x86-32.64.*
- *Здатність проектувати лексичні аналізатори на основі скінчених автоматів.*
- *Здатність проектувати синтаксичні аналізатори на основі рекурсивного спуску.*
- *Здатність працювати з лексичними і синтаксичними генераторами.*

## 5. Організація навчання курсу

<b>Обсяг курсу</b>					
Вид заняття			Загальна кількість годин		
лекції			40		
семінарські заняття / практичні / <u>лабораторні</u>			46		
самостійна робота			94		
<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
1-й семестр					
<b>Змістовий модуль 1. Основи асемблера NASM</b>					
Тема 1. Вступ в системне програмування.	лекція	1, 2, 3, 8, 11	2	3	Згідно розкладу
Тема 2. Режим роботи і моделі пам'яті процесорів intel x86-32/64.	лекція	6, 9, 10	2	3	Згідно розкладу
Тема 3. Основи Nasm асемблера.	лекція	4	2	3	Згідно розкладу
Тема 4. Nasm макропроцесор і макродирективи	лекція	4	2	3	Згідно розкладу
Тема 5. Класифікація цілочислових команд. Команди пересилання даних, арифметичні, логічні, ланцюгові.	лекція	6, 7	2	3	Згідно розкладу
Модульний контроль 1			2		
<b>Змістовий модуль 2. Додаткові можливості асемблера NASM</b>					
Тема 6. Команди передачі керування.	лекція	5, 6, 7	2	3	Згідно розкладу
Тема 7. Процеси, системні виклики і підпрограми.	лекція	5, 6, 7, 9, 10	2	3	Згідно розкладу
Тема 8. Співпроцесор.	лекція	5, 6, 7, 9, 10	2	3	Згідно розкладу
Тема 9. Технологія MMX	лекція	5, 6, 9, 12	2	3	Згідно розкладу
Тема 10. Технологія SSE	лекція	12	2	3	Згідно розкладу

Модульний контроль 2			2		
<b>Практичний модуль</b>					
1. Програмна модель процесора, типи даних, директиви і адресація пам'яті	лаб. робота	5, 6, 7, 9	2	4	Згідно розкладу
2. Синтаксис і формати команд асемблерів NASM і GAS.	лаб. робота	4, 12	2	4	Згідно розкладу
3. Команди mov, lea, цілочисельні арифметичні.	лаб. робота	5, 6, 7, 9, 10	2	4	Згідно розкладу
4. Безумовні та умовні переходи, цикли, логічні побігові команди і операції над виразами, зсуви, деасемблери.	лаб. робота	5, 6, 7, 9, 10	2	4	Згідно розкладу
5. Ланцюгові команди.	лаб. робота	5, 6, 7, 9, 10	2	4	Згідно розкладу
6. Директиви препроцесора.	лаб. робота	5, 6, 7, 9, 10	2	4	Згідно розкладу
7. Виклики підпрограм.	лаб. робота	5, 6, 7, 9, 10	2	4	Згідно розкладу
8. Взаємодія з ОС. Переривання.	лаб. робота	5, 6, 7, 9, 10	2	4	Згідно розкладу
9. Структури даних.	лаб. робота	5, 6, 7, 9, 10	2	4	Згідно розкладу
10. Модульне програмування.	лаб. робота	2, 5, 6, 7, 9, 10, 12	2	5	Згідно розкладу
11. Співпроцесор.	лаб. робота	5, 6, 7, 9, 10, 12	2	5	Згідно розкладу
<b>Самостійна робота</b>					
Тема 1. Макроси.	самостій на робота	4, 9, 10	6	3	Впродовж семестру
Тема 2. Введення даних з консолі	самостій на робота	5, 6, 7, 9, 12	6	3	Впродовж семестру
Тема 3. Робота з файлами.	самостій на робота	5, 6, 7, 9, 12	6	3	Впродовж семестру
Тема 4. Команди зневадника GDB.	самостій на робота	12	6	3	Впродовж семестру
Тема 5. Налаштувач з	самостій	12	6	3	Впродовж

графічним інтерфейсом DDD	на робота				семестру
Тема 6. Інтерфейси з мовами високого рівня.	самостій на робота	5, 6, 7	6	3	Впродовж семестру
Тема 7. Команди SSE розширення процесорів x86_64.	самостій на робота	5, 7, 9, 12	6	3	Впродовж семестру
Тема 8. Команди SSE2 процесорів x86_64.	самостій на робота	5, 7, 9, 12	6	3	Впродовж семестру
Контроль самостійної роботи			2		Згідно розкладу
<b>Підсумковий контроль (залік)</b>				100	Згідно розкладу
2-й семестр					
<b>Змістовий модуль 3. Лексичний і синтаксичний аналіз</b>					
Тема 1. Лексичний аналіз. Специфікація і розпізнання токенів	лекція	20, 21, 23	2	3	Згідно розкладу
Тема 2. Лексичний аналіз. Недетерміновані і детерміновані скінченні автомати.	лекція	21, 22, 25	2	3	Згідно розкладу
Тема 3. Синтаксичний аналіз. Контекстно вільні граматики.	лекція	20, 21, 23	2	3	Згідно розкладу
Тема 4. Синтаксичний аналіз. Метод рекурсивного спуску.	лекція	21, 23	2	3	Згідно розкладу
Тема 5. Синтаксичний аналіз. Висхідний синтаксичний аналіз.		21, 23, 24	2	3	
Модульний контроль			2		Згідно розкладу
<b>Змістовий модуль 4. Трансляція і генерація машинного коду</b>					
Тема 6. Синтаксично керована трансляція.	лекція	21, 23, 27	2	3	Згідно розкладу
Тема 7. Генерація проміжного коду.	лекція	21, 23, 29	2	3	Згідно розкладу
Тема 8. Середовища часу виконання.	лекція	21, 23, 30	2	3	Згідно розкладу
Тема 9. Генерація машинного коду.	лекція	21, 23, 30	2	3	Згідно розкладу
Тема 10. Інструменти для		25, 26, 27		3	

генерації машинного коду.					
Модульний контроль			2		Згідно розкладу
<b>Практичний модуль</b>					
1. Регулярні вирази.	лаб. робота	21, 23	2	4	Згідно розкладу
2. Пошук ключових слів у стрічках.	лаб. робота	21, 23	2	4	Згідно розкладу
3. Недетермінований і детермінований скінченні автомати.	лаб. робота	21, 23, 24	2	4	Згідно розкладу
4. Побудова лексичного аналізатора на мові C, Python.	лаб. робота	28, 29	2	4	Згідно розкладу
5. Побудова лексичного аналізатора з використанням FLEX.	лаб. робота	21	2	4	Згідно розкладу
6. Побудова лексичного аналізатора з використанням PLY.	лаб. робота	31	2	4	Згідно розкладу
7. Трансляція арифметичних виразів з інфіксною у постфіксну/префіксну нотацію.	лаб. робота	21, 22, 24	2	4	Згідно розкладу
8. Синтаксичний аналізатор простих арифметичних виразів.	лаб. робота	21, 23, 25, 26	2	4	Згідно розкладу
9. Низхідний рекурсивний синтаксичний аналізатор.	лаб. робота	21, 25, 27	2	4	Згідно розкладу
10. Предиктивний синтаксичний аналізатор LL(1).	лаб. робота	21, 23, 25,	2	4	Згідно розкладу
11. Побудова таблиці предиктивного синтаксичного аналізу.	лаб. робота	21, 23, 24, 25, 26	2	4	Згідно розкладу
12. . Побудова синтаксичного аналізатора з використанням генераторів	лаб. робота	21, 23, 24, 25, 26	2	4	Згідно розкладу
<b>Самостійна робота</b>					
Тема 1. Мінімізація кількості станів детермінованого скінченного автомата	самостій на робота	21, 23	6	4	Впродовж семестру

Тема 2. Мови і контекстно вільні граматики	самостійна робота	21, 23, 24	8	4	Впродовж семестру
Тема 3. Розроблення граматики мови програмування.	самостійна робота	21, 23, 26	8	4	Впродовж семестру
Тема 4. Функції FIRST і FOLLOW в низхідному синтаксичному аналізі.	самостійна робота	21, 23, 26	8	4	Впродовж семестру
Тема 5. LR синтаксичні аналізатори.	самостійна робота	21, 23, 25, 26	8	4	Впродовж семестру
Тема 6. Генерація проміжкового і цільового машинного коду.	самостійна робота	21, 23, 24, 25	8	2	Впродовж семестру
Контроль самостійної роботи			2		Згідно розкладу
<b>Підсумковий контроль (екзамен)</b>				100	Згідно розкладу

## 6. 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 (сума балів за окремий змістовий модуль)</i> 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 (1-й семестр), екзамену (2-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>		
<b>Шкала оцінювання: національна та ECTS</b>		
Сума балів за	Оцінка ECTS	Оцінка за національною шкалою

всі види навчальної діяльності		для екзамену, курсового проекту (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

## 7. 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 врахування отриманих оцінок в іншому навчальному закладі за умови відповідності навчальних планів.

### **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, використанні методики досліджень і джерелі інформації.

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

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

## 8 Рекомендована література

### Основна (1-й семестр)

1. Кетков, Ю.Л. Введение в системное программирование на языке асемблера ЕС ЭВМ / Ю.Л. Кетков, В.С. Максимов, А.Н. Рябов. – М.: Наука, 2018. – 264 с.
2. Лав, Р. Linux. Системное программирование / Р. Лав. М.: Питер, 2016. – 456 с.
3. Собель, Марк. Linux. Администрирование и системное программирование / Марк Собель. – М.: Питер, 2016. – 341 с.
4. Столяров А.В. Программирование на языке асемблера NASM для ОС UNIX. Уч. пособие. – 2-е изд. – М.: Макс-пресс, 2011. – 188 с.
5. Аблязов Р.З. Программирование на асемблера на платформе x86-64. – М.: ДМК Пресс, 2011. – 304 с.
6. Юров В. Assembler. Практикум. – СПб.: Питер, 2006, – 399 с.
7. Магда Ю.С. Асемблер для процессоров Intel Pentium. – СПб.: Питер, 2006. – 410 с.
8. Харт Джонсон М. Системное программирование в среде Windows / Джонсон М. Харт ; пер. с англ. – М. : Издательский дом «Вильямс», 2005.
9. Юров В.И. Assembler. Учебник для вузов. 2-е изд. – СПб.: Питер, 2003. – 637 с.
10. Юров В. Assembler. – СПб.: Питер, 2003. – 637 с.
11. Вильямс А. Системное программирование Windows 2000. СПб.: Питер, 2001.

### Додаткова

12. Jo Van Noey. Beginning x64 Assembly Programming: From Novice to Professional. CA: Apress, 2019. – 413 p.
13. Стивенс У., Феннер Б., Рудофф Э. Разработка приложений в среде Linux. М.: Вильямс, 2007. – 544 с.
14. Бовет Д. Ядро Linux, 3-изд.: Пер. с англ. СПб: БХВ-Петербург, 2007. – 1104 с.
15. Родригес К.З., Фишер Г., Смолски С. Linux: азбука ядра / Перевод с англ. М.: Кудиц пресс, 2007. – 584 с.
16. Роберт Лав. Разработка ядра Linux, 2-изд., пер. С англ. М.: ООО “И.Д. Вильямс”, 2006. – 448 с.
17. Кулаков В. Программирование на аппаратном уровне. Спец. справ. СПб.: Питер, 2003.
18. Митчел М., Оулдем Дж., Самьюэл А. Программирование для Linux. Профессиональный подход. М. – СПб. – К.: Вильямс, 2002.
19. Кристиан К. Введение в операционную систему UNIX. М.: Финансы и статистика, 1985.

### Основна (2-й семестр)

20. Watt, David A. Programming language processor in Java: compilers and interpreters / Davit A. Watt and Deryck F. Brown. GB: Pearson EL, 2000. – 424 p.
21. Ахо Альфред, Лам Моника С., Сети Рави, Ульман Джеффри. Компиляторы: принципы, технологии, инструментарий. Пер. с англ. – М.: Изд. дом “Вильямс”, 2008. – 1184 с.

22. Опалева Э.А., Самойленко В.П. Языки программирования и методы трансляции. – СПб.: БХВ-Петербург, 2005. – 480 с., ил.
23. Альфред Ахо, Рави Сети, Джеффри Ульман. Компиляторы: принципы, технологии, инструменты. Пер. с англ. – М.: Изд. дом “Вильямс”, 2003. – 768 с.
24. Молчанов А.Ю. Системное программное обеспечение: Учебник для вузов. – СПб.: Питер, 2003. – 396 с.
25. Хангер Робин. Основные концепции компиляторов. – М.: Вильямс, 2002. – 256 с.
26. Костельцев А.В. Построение интерпретаторов и компиляторов. – СПб.: Наука и техника, 2001. – 224 с.

#### Додаткова

27. Keith D. Cooper and Linda Torczon. Engineering a Compiler. 2012 Elsevier, Inc., 2012. – 800 p.
28. Andrew W. Appel, Maia Ginsburg. Modern Compiler implementation in C. Cambridge University press, 1998. – 544 с.
29. Allen I. Holub. Compiler design in C. Prentice-Hall, Inc. , 1990. – 984 p.
30. Jean-Paul Tremblay, Paul G. Sorenson. The theory and practice of compiler writing. McGraw-Hill Book Company, 1985. – 796 p.
31. Интернет ресурс <https://www.dabeaz.com/py> – PLY (Python Lex-Yacc)

Викладач



Голота В.І.