



Системне програмування  
Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення комп'ютерних систем</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>очна(денна), заочна</i>
Рік підготовки, семестр	<i>2 курс, весінній</i>
Обсяг дисципліни	<i>4,5 кредитів /135 год. Денна форма: лекцій 36 год., лаб.робіт 18 год, СРС 81 год.</i>
Семестровий контроль/ контрольні заходи	<i>Екзамен</i>
Розклад занять	<i>Згідно розкладу на весняний семестр поточного навчального року за адресою <a href="http://rozklad.kpi.ua/">http://rozklad.kpi.ua/</a></i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектори: к.т.н., доцент, Павлов Валерій Георгійович, <a href="mailto:Pavlov.Valeriy@Ill.kpi.ua">Pavlov.Valeriy@Ill.kpi.ua</a></i>
Розміщення курсу	<i>На платформі дистанційного навчання «Сікорський»: <a href="https://do.ipu.kpi.ua/course/view.php?id=6880">https://do.ipu.kpi.ua/course/view.php?id=6880</a></i>

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Навчальна дисципліна "Системне програмування" відноситься до нормативних освітніх компонент навчальної програми, а саме до циклу професійної підготовки. Вона має номер **ПО 14** у переліку компонент освітньої програми.

**Причини та мотивація до вивчення:** необхідність розуміння принципів програмного керування безпосередньо процесору. Ця дисципліна надасть розуміння, яким чином у комп'ютерній системі відбувається взаємодія апаратної та програмної складових, тобто поєднує у єдиний комплекс знання архітектури комп'ютерних систем та їхнє програмування.

**Мета навчальної дисципліни:** вивчення базові архітектури та програмне оточення процесорів Intel 64 та IA-32, структуру форматів шістнадцяткових та двійкових чисел для цілих та дробових значень, систему команд процесору та їхню структуру, основи нізкорівневого програмування на мові Assembler.

**Метою навчальної дисципліни** є формування у студентів низки компетентностей, а саме:

**ЗДАТНІСТЬ:**

- розуміти принципи керування процесором на програмному рівні;
- розуміти послідовність дій під час розробки програм на машинній мові;
- аналізувати структуру програм на машинній мові;
- розробляти програми на мові Assembler;
- аналізувати процеси, які здійснюються під час компіляції та лінковки програм;
- використовувати можливості макроасемблерів, у програмуванні;
- розробляти мобільні системи, вбудовані системи та системи реального часу (ФК16);
- розробляти та використовувати методи і алгоритми високопродуктивних обчислень (ФК17) за рахунок керування ресурсами пам'яті.

У відповідності до вищезазначеного, студенти отримують наступні результати навчання:

**ЗНАННЯ:**

- структури та форматів команд на машинній мові;
- основних форматів зберігання даних відповідно стандарту IEEE 754-2008;
- структури програм на мові Assembler;
- знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення (ПРН07);

**УМІННЯ:**

- визначати послідовність дій для розробки системної програми на мові Assembler;
- використати інтегровані середовища розробки (IDE) системних програм;
- аналізувати повідомлення компілятора під час відладки програми;
- аналізувати лістинг програми на машинній мові;
- шукати та виправляти синтаксичні та логічні помилки у системної програмі;
- мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення (ПРН15);
- застосовувати методи компонентної розробки програмного забезпечення (ПРН18).

**ЗНАННЯ:**

- структури та форматів команд на машинній мові;
- основних форматів зберігання даних відповідно стандарту IEEE 754-2008;
- структури програм на мові Assembler;
- знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення (ПРН07);

**УМІННЯ:**

- визначати послідовність дій для розробки системної програми на мові Assembler;
- використати інтегровані середовища розробки (IDE) системних програм;
- аналізувати повідомлення компілятора під час відладки програми;
- аналізувати лістинг програми на машинній мові;
- шукати та виправляти синтаксичні та логічні помилки у системної програмі;
- мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення (ПРН15);
- застосовувати методи компонентної розробки програмного забезпечення (ПРН18).

## **2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)**

*При вивченні дисципліни «Системне програмування» студенти доцільно використовувати знання, отримані при вивченні попередніх дисциплін «Історія науки і техніки», «Комп'ютерна дискретна математика», « Основи комп'ютерних систем і мереж».*

*Дисципліна «Системне програмування» є базовою для курсів: «Компоненти програмної інженерії», «Операційні системи», «Безпека програмного забезпечення», а також під час курсового та дипломного проектування.*

## **3. Зміст навчальної дисципліни**

*Вступ.*

*Розділ 1. Базові поняття мови Assembler*

*Тема 1.1 Процесор, як об'єкт програмування*

- *Місце мови Assembler у класифікації мов програмування.*
- *Структура процесора у концепції фон Неймана.*
- *Хронологія розвитку процесорних архітектури INTEL та AMD.*
- *Принципи організації та режими використання комп'ютерної пам'яті.*
- *Організація регістрової пам'яті мікропроцесора x86-64.*

*Тема 1.2. Формати представлення чисел за стандартом IEEE 754-2008.*

- *Цілі числа без знаку.*
- *Цілі числа зі знаком.*
- *Дійсні числа.*
- *Формат BCD двійково-десяткових чисел.*

*Тема 1.3. Формати команд процесорів x86*

- *Загальна структура команди у машинній мові*
- *Поняття коду операції.*
- *Поняття префіксу команди.*
- *Різновиди адресації у командах.*
- *Залежність формату команди від види адресації.*
- *Поле Mod/RM.*
- *Поле SIB.*
- *Формати запису команд на мові Assembler.*
- *Інваріантність кодів команд Assembler.*

*Розділ 2. Програмування на мові Assembler*

*Тема 2.1. Розробка програм на мові Assembler.*

- *Етапи розробки програми.*
- *Основні режими роботи компілятора.*
- *Основні режими роботи компонувальника.*
- *Використання відладчиків та дізасемблерів.*

- Структура програми на мові *Assembler*.
- Директиви компілятора.
- Регістр станів та основні прапорці.

Тема 2.2. Застосування макроасемблерів.

- Макрокоманди та макровизначення.
- Макрозмінні та макрообчислення.

Тема 2.3. Застосування процедур та підпрограм.

- Принципи організації стекової пам'яті.
- Внутрішні та зовнішні процедури.
- Способи передачі параметрів до процедури та їхнє повернення з процедури.
- Рекурсивний виклик процедур.
- Розробка бібліотек динамічної компоновки.

## 4. Навчальні ресурси та матеріали

### 4.1. Базова література

1. Рисований О.М. Системне програмування. Підручник. В 2-х т. – Харків, : “Слово”, 2015.– 576 с.
2. Галісеєв. Г Системне програмування. – К., : Видавництво Університет "Україна", 2019. - 113 с.
3. Тонкошкур О.С, Гниленко, О.Б, Матвєєва Н.О, Морозов О.С. Архітектура комп'ютерів. Машинні команди та програмування на асемблері: навчальний посібник - Д.: Вид-во «Нова Ідеологія», 2018, 179 с.
4. Kip R. Irvine. *Assembly Language for x86 Processors*. – Florida International University: Pearson, Seventh Edition 2014, - 876 p.
5. Rudolf Marek. *Učíte se programovat v jazyce Assembler pro PC*. Computer Press, 2003. – 228 p.

### 4.2. Додаткова література

1. Intel Corporation. *Intel 64 and IA-32 Architectures Software Developer's Manual. Volumes 1–3* [Електронний ресурс]:– 2014. <http://www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html>.
2. Microsoft *Portable Executable and Common Object File Format Specification*. [Електронний ресурс]:– <http://www.osdever.net/documents/PECOFF.pdf>.
3. Рисований О.М. Системне програмування. – Харків, : НТУ «Харківський політехнічний інститут», 2012, 912 с.
4. Соколов А. Системне програмування: лабораторний практикум. – Lambert, 2018, - 68 с.

### 4.3. Інформаційні ресурси

1. Дистанційні курси на платформі дистанційного навчання «Сікорський» в середовищі MOODLE: Системне програмування. <https://do.ipk.kpi.ua/course/view.php?id=6880> .
2. Згідно обмеженням, накладеним адміністраторами MOODLE на обсяг. Відео-ролики лекцій будуть розміщені на YouTube, а вказаний ресурс містить посилання на них.

## Навчальний контент

### 5. Методика опанування навчальної дисципліни (освітнього компоненту)

Розподіл навчального часу за видами занять і завдань з дисципліни згідно з робочим навчальним планом. На вивчення дисципліни виділено 135 годин/4,5 кредити.

Назви розділів і тем	Кількість годин				
	Всього	у тому числі			
		Лекції	Практичні (семінарські)	Лабораторні (комп'ютерній практикум)	СРС
<b>Розділ 1. Базові поняття мови Assembler</b>					
Тема 1.1. Процесор, як об'єкт програмування	8	4	–	–	4
Тема 1.2. Формати представлення числових констант за стандартом IEEE 754-2008	14	4	–	2	8
Тема 1.3. Формати команд процесорів x86	38	10	–	6	22
Модульна контрольна робота 1	1	–	–	–	1
Разом за розділом 1	61	18	–	8	35
<b>Розділ 2 Програмування на мові Assembler</b>					
Тема 2.1. Розробка програм на мові Assembler.	18	6	–	2	10
Тема 2.2. Застосування макроасемблерів.	18	6	–	2	10
Тема 2.3. Застосування процедур та підпрограм.	30	6	–	6	18
Модульна контрольна робота 2	1	–	–	–	1
Разом за розділом 2	67	18	–	10	39
Підготовка до екзамену	7	–	–	–	7
<b>Всього годин</b>	<b>135</b>	<b>36</b>	<b>–</b>	<b>18</b>	<b>81</b>

### Лабораторні заняття

Метою лабораторних робіт є отримання студентами необхідних практичних навичок розробки окремих модулів керування обчислювальними процесами, забезпечення виконання спеціальних функцій у мові Assembler та низькорівневому програмуванні.

№ з/п	Назва лабораторної роботи (комп'ютерного практикуму)	Кількість ауд. годин
1	Внутрішнє представлення цілих і дійсних даних у процесорі ix86.	4
2	Дослідження структури файлів формату COM.	4
3	Дослідження структури файлів формату EXE.	4
4	Використання макросів у середовищі MASM.	4
5	Арифметичні і логічні операції з цілими числами.	4
6	Використання команд співпроцесора.	4
7	Модульне програмування. Використання процедур.	4
8	Розробка і використання динамічних бібліотек.	8
	<b>Разом:</b>	36

### 6. Самостійна робота здобувача вищої освіти денної форми навчання

№ з/п	Назви тем і питань, що виносяться на самостійне опрацювання	Кількість годин СРС
1	Виконання завдань за темою кожного лекційного заняття (див. розділ 5) – 0,5 год. на 1 годину лекції	18
2	Підготовка до лабораторних занять (див. розділ 5) – 1,5 год. на 1 годину лабораторного заняття	27
3	Підготовка до модульних контрольних (див. розділ 5) – 3 години на кожну контрольну	6
4	Підготовка до екзамену	30
	<b>Разом:</b>	81

### Політика та контроль

#### 7. Політика навчальної дисципліни (освітнього компонента)

При зарахування та оцінювання лабораторних робіт беруться до уваги наступні чинники:

- Повнота виконання завдання на лабораторну роботу за індивідуальним варіантом;
- Своєчасність виконання лабораторної роботи згідно графіку;
- Самостійність виконання лабораторної роботи та відсутність ознак плагіату;
- Відповіді на питання щодо змісту лабораторної роботи під час її захисту.

При оцінюванні контрольних робіт до уваги приймаються:

- Правильність та повнота виконання завдань;
- Кількість виконаних завдань в умовах обмеженого часу;
- Самостійність виконання завдань та відсутність ознак плагіату;

- Кількість спроб виконання контрольних, які передують тій, що оцінюється.

Для підготовки до модульних контрольних студенти отримують перелік теоретичних питань та зміст типових задач, які будуть у завданнях на контрольних.

При першій та другій атестації до уваги приймається кількість лабораторних робіт та модульних контрольних робіт зарахованих на час проведення атестації.

## 8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Система оцінювання успішності студентів з дисципліни «Системне програмування» ґрунтується на «Положенні про систему оцінювання результатів навчання в КПІ ім. Ігоря Сікорського» ([https://document.kpi.ua/files/2020\\_1-273.pdf](https://document.kpi.ua/files/2020_1-273.pdf)), а саме Рейтингової системі оцінювання (PCO) другого типу (PCO-2).

PCO-2 складається з двох складових:

- стартової ( $R_S$ );
- екзаменаційної ( $R_E$ ).

Стартові бали формуються як сума балів, що отримані за результатами заходів поточного контролю (лабораторних робіт ( $R_L$ ) і контрольних ( $R_K$ )), заохочувальних ( $R_3$ ) та штрафних ( $R_{Ш}$ ) балів:

$$R_S = R_L + R_K + R_3 + R_{Ш}$$

де  $R_L$  за 8 лабораторних робіт складає  $8 \times 5 = 40$  балів,

$R_K$  за дві модульні контрольні складає  $2 \times 15 = 30$  балів.

Таким чином максимальна основна сума стартових балів складає  $40 + 30 = 70$  балів.

Поза основною шкалою оцінювання передбачені заохочувальні та штрафні бали, які враховуються у загальній сумі балів, але не входять до основної шкали PCO.

Заохочувальні бали враховують відповідь на питання та виконання завдань на лекційних заняттях, якість конспекту.

Штрафні бали передбачені за несвоєчасне виконання лабораторних робіт, тобто зі запізненням відносно графіку, якщо відносно цього відсутні рекомендації з боку ДНВР.

Оцінювання кожної лабораторної роботи в межах 5 балів виконується за двома складовими:

- зміст лабораторної роботи (в межах від 0 до 3 балів);
- вчасність виконання за дотримання графіку виконання лабораторних робіт (в межах від 0 до 2 балів), але ці бали додаються до першої складової при її наявності.

Екзаменаційні бали ( $R_E$ ), таким чином складають максимально 30 балів, але для допуску до екзамену студент повинен мати не менш ніж 60% від максимальної суми стартових балів, що складає

$$70 \times 0.6 = 42 \text{ бала.}$$

Після складання екзамену стартові бали  $R_S$  складаються зі екзаменаційними  $R_E$ . Оцінювання результатів навчання здійснюється за 100-бальною шкалою з подальшим переведенням оцінок за університетською шкалою згідно таблиці:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо

Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

## 10. Визнання результатів неформальної освіти

Визнання результатів неформальної освіти відбувається згідно Положення про визнання в КПІ ім. Ігоря Сікорського результатів навчання, набутих у неформальній / інформальній освіті [https://osvita.kpi.ua/sites/default/files/downloads/Неформ\\_інформ.pdf](https://osvita.kpi.ua/sites/default/files/downloads/Неформ_інформ.pdf)

В рамках даного курсу можуть бути визнані результати навчання здобуті у неформальній/інформальній освіті в обсязі, що не перевищує 10% від загального обсягу навчального курсу (п.2.6 Положення).

У разі виконання рекомендованого викладачем онлайн курсу додаткова валідація результатів неформального навчання не потрібна. Поточний контроль з відповідної частини курсу оцінюється відповідно до рейтингової системи оцінювання результатів навчання та політики навчальної дисципліни. В такому форматі одним онлайн курсом можна замінити одну лабораторну роботу на вибір (5 балів) і не можна замінити МКР.

У разі зарахування сторонніх результатів неформальної освіти, визнання результатів проводиться на початку семестру, у якому передбачено опанування освітнього компонента, який може бути частково зарахований. Викладач проводить аналіз їх відповідності силабусу, проводить співбесіду із студентом. Студент має підготувати і захистити звіт з результатами опанованої частини курсу. В окремих випадках може бути зарахований весь курс, або більша частина курсу. Процедура відбувається згідно Положення з дозволу декана, валідацію результатів навчання проводить комісія.

Робочу програму навчальної дисципліни (силабус):

**Складено** к.т.н, доцент, доцент кафедри ОТ Павлов Валерій Георгійович.

**Ухвалено** кафедрою обчислювальної техніки (протокол № 13 від 10.05.2023р.).

**Погоджено** методичною комісією ФІОТ (протокол № 11 від 29.06.2023 р.).