



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)</i>
Рік підготовки, семестр	<i>2 курс, весняний семестр</i>
Обсяг дисципліни	<i>4.5 кредити (ECTS). Обсяг дисципліни - 135 годин, у тому числі 54 аудиторних години та 81 години самостійної роботи студентів.</i>
Семестровий контроль/ контрольні заходи	<i>4 семестр – екзамен,</i>
Розклад занять	<i>3 години у тиждень за розкладом http://rozklad.kpi.ua/</i>
Мова викладання	<i>Українська / Англійська</i>
Інформація про керівника курсу / викладачів	<i>Лекції та лабораторні роботи: к.т.н., доцент Павлов Валерій Георгійович, pavlovvg@ukr.net</i>
Розміщення курсу	

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 низкорівневого 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.

1.2. Основні завдання навчальної дисципліни.

Після засвоєння навчальної дисципліни студенти мають продемонструвати такі 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 системній 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)

Міждисциплінарні зв'язки: Для успішного вивчення курсу «Системне програмування» студенти повинні засвоїти матеріал та мати певні знання, вміння та навички з таких дисциплін, як «Історія науки і техніки», «Комп'ютерна дискретна математика», «Основи комп'ютерних систем і мереж», «Основи програмування», «Об'єктно-орієнтоване програмування»

Знання та навички, які отримуються під час вивчення дисципліни «Системне програмування», можуть бути використані в подальшому при опануванні наступних курсів: «Компоненти програмної інженерії», «Операційні системи», «Безпека програмного забезпечення», а також під час курсового та дипломного проектування.

3. Зміст навчальної дисципліни

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 виду 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 компілятору.
- 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 матеріали та ресурси

4.1. Базова

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. Кравець В. О., Рисований О.М. Системне програмування. Програмування на мові асемблера: навч. посіб. – Х. : НТУ «ХПІ», 2007. – 448 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 Допоміжна

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.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Назви розділів і тем	Кількість годин				
	Всього	у тому числі			
		Лекції	Практичні (семінарські)	Лабораторні (комп'ютерний і практикум)	СРС
Розділ 1. Базові поняття мови Assembler					
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
Контрольна робота 1	1	–	–	–	1
Разом за розділом 1	61	18	–	8	35
Розділ 2 Програмування на мові Assembler					
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
Контрольна робота 2	1	–	–	–	1
Разом за розділом 2	67	18	–	10	39
Підготовка до екзамену	7	–	–	–	7
Всього годин	135	36	–	18	81

Лабораторні заняття

Основні завдання циклу лабораторних занять є отримання студентами необхідних практичних навичок розробки окремих модулів керування обчислювальними процесами, забезпечення виконання спеціальних функцій у мові Assembler та низькорівневого 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 MASM32.	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
	Разом:	36

6. Самостійна робота

№ 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) – 1 год. на 1 годину лекції	36
2	Підготовка до лабораторних занять (див. розділ 5) – 1 год. на 1 годину лабораторного заняття	36
3	Підготовка до контрольних (див. розділ 6) – 1 година на кожну контрольну	2
4	Підготовка до екзамену	7
	Разом:	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 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 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), а саме Рейтингової системи 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 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.

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

Робочу програму навчальної дисципліни (силабус):

Складено к.т.н., доцент, Павлов В.Г.

Ухвалено кафедрою обчислювальної техніки (протокол № 10 від 25 травня 2022)

Погоджено Методичною комісією факультету ФІОТ (протокол № 10 від 09 червня 2022)