



# Системне програмування в середовищі Unix

## Робоча програма навчальної дисципліни (Силабус)

### Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>123 Комп'ютерна інженерія</i>
Освітня програма	<i>Комп'ютерні системи та мережі</i>
Статус дисципліни	<i>Вибіркова</i>
Форма навчання	<i>Очна (денна) / Заочна</i>
Рік підготовки, семестр	<i>4 курс, осінній семестр</i>
Обсяг дисципліни	<i>4 кредити. Всього – 108 годин, з них 36 години лекції, 18 годин лабораторні заняття, 54 годин самостійної роботи.</i>
Семестровий контроль/ контрольні заходи	<i>Залік</i>
Розклад занять	<i>1.5 заняття на тиждень за розкладом <a href="http://rozklad.kpi.ua/">http://rozklad.kpi.ua/</a></i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: старший викладач, Андрій Сімоненко, <a href="mailto:comsys.spz@gmail.com">comsys.spz@gmail.com</a> Лабораторні: старший викладач, Андрій Сімоненко, <a href="mailto:comsys.spz@gmail.com">comsys.spz@gmail.com</a></i>
Розміщення курсу	<i><a href="https://drive.google.com/drive/folders/1d2xx7CZ9lgMH_Zy9YT6h-Ya0RGQ9uKaT">https://drive.google.com/drive/folders/1d2xx7CZ9lgMH_Zy9YT6h-Ya0RGQ9uKaT</a></i>

### Програма навчальної дисципліни

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

**Що буде вивчатися.** Вивчатиметься системне програмування в середовищі Unix, тобто розробка системних програм для Unix-подібних операційних систем на рівні використання системних викликів для взаємодії з ядром. Надається ґрунтовна інформація про функції POSIX (та достатня інформація про реалізації відповідних системних викликів), які використовуються при розробці системних програм. Дисципліна не орієнтована на системне програмування в якийсь конкретній реалізації Unix-подібної операційної системи, вивчатиметься переносне системне програмування. Дисципліна складається з наступних тем: середовище виконання програми, управління процесами, робота з файлами, робота з сигналами, робота з pipe, просунуте введення-виведення, управління пам'яттю процесу, програмування псевдотерміналів та інші.

**Чому це цікаво/треба вивчати.** Цю дисципліну доцільно вивчати тим, хто буде розробляти системні програми для Unix-подібних операційних систем. Програмування завдань виконується мовою програмування C, або C++, або Rust, але набуті знання будуть корисні при вирішенні деяких системних задач для Unix-подібних операційних систем іншими мовами програмування.

**Чому можна навчитися (результати навчання).** Розробляти системні програми для Unix-подібних операційних систем мовою програмування C, або C++, або Rust, які управляють процесами, працюють з файлами, працюють з сигналами, використовують просунуте введення-виведення, працюють з pipe, працюють з псевдотерміналами.

**Як можна користуватися набутими знаннями і уміннями (компетентності).** Набуті знання можливо використовувати при розробці системних програм для Unix-подібних операційних

систем, для супроводження вихідного коду наявних системних програм для Unix-подібних операційних систем, при розробці ефективніших прикладних програм.

**Кредитний модуль забезпечує наступні компетентності і програмні результати освітньо-практичної програми першого рівня вищої освіти: ФК2, ФК3, ФК15, ПРН8, ПРН11.**

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

Вміння програмувати на C, або C++, або Rust та вміння працювати в Unix-подібній операційній системі на рівні користувача. Базові знання з дисциплін Програмування, Структури даних та алгоритми.

## **3. Зміст навчальної дисципліни**

Тема 1. Середовище виконання програми

Тема 2. Дерево файлової системи

Тема 3. Дескриптор файлу

Тема 4. Користувачі та облікові дані

Тема 5. Сигнали

Тема 6. Управління процесами

Тема 7. Pipe та FIFO

## **4. Навчальні матеріали та ресурси**

1. The Open Group. Single UNIX specification, version 4 - IEEE and The Open Group, 2018.
2. W. Richard Stevens, Stephen A. Rago. Advanced Programming in the UNIX Environment, 3rd edition. - Addison-Wesley Professional, 2013 - 1032 p.
3. Michael Kerrisk. The Linux Programming Interface: A Linux and UNIX System Programming Handbook, 1st edition. - No Starch Press, 2010. - 1552 p.

## **Навчальний контент**

### **5. Методика опанування навчальної дисципліни (освітнього компонента)**

<i>Розділи, теми</i>	<i>Всього, г</i>	<i>Лекції, г</i>	<i>Лабораторні роботи, г</i>	<i>Самостійне навчання, г</i>
<i>Тема 1. Середовище виконання програми</i>	17	6	4	7
<i>Тема 2. Дерево файлової системи</i>	12	5		7
<i>Тема 3. Дескриптор файлу</i>	17	5	4	8
<i>Тема 4. Користувачі та облікові дані</i>	12	5		7
<i>Тема 5. Сигнали</i>	14	5		9
<i>Тема 6. Управління процесами</i>	18	5	5	8
<i>Тема 7. Pipe та FIFO</i>	18	5	5	8
<i>Залік</i>	3			
<i>Всього</i>	108	36	18	54

*Лабораторні роботи:*

1. Налаштування середовища розробки (4 г)
2. Дерево файлової системи (4 г)
3. Командний інтерпретатор (5 г)
4. Розподілена спільна пам'ять (5 г)

## 6. Самостійна робота студента/аспіранта

У процесі розуміння тем із лекцій та виконання лабораторних робіт студенти мають опрацьовувати знання, що були отримані під час лекцій та лабораторних робіт, самостійно вивчаючи певні теми, використовуючи інформацію з Internet, поглиблюючи свої знання для подальшого навчання.

Самостійне вивчення полягає в наступному:

1. Вивчення та розуміння тем з попередніх лекцій.
2. Виконання завдань, наданих для самостійного вивчення.
3. Виконання лабораторних робіт.
4. Написання звітів до лабораторних робіт.

## Політика та контроль

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

Політика курсу повністю відповідає правилам та нормам, що видані КПІ. Для зарахування лабораторної роботи необхідно набрати 60% від максимальної кількості балів за неї. Щоб отримати допуск до заліку необхідно здати всі лабораторні роботи. Для отримання першої атестації необхідно мати зараховані перші дві лабораторну роботу. Для отримання другої атестації необхідно мати зараховану третю лабораторну роботу. Кількість спроб здачі будь-якої лабораторної роботи не обмежена. Здачі та перевірки лабораторних робіт за розкладом групи. Якщо у виконаній лабораторній роботі присутні помилки або не відповідність умовам завдання лабораторної роботи і якщо виконавець відмовляється виправляти помилки або не відповідності, то робота не зараховується або зараховується з меншими балами.

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

Згідно з правилами, що видані КПІ, максимальна кількість балів 100 розподіляється між лабораторними роботами. Здобувачі, які виконали всі умови допуску до заліку, тобто мають рейтингову оцінку 60 та вище балів, отримують відповідну до набраного рейтингу оцінку. Зі здобувачами, які бажають підвищити свою рейтингову оцінку, на завершальному за розкладом занятті з дисципліни в семестрі проводиться семестровий контроль у вигляді залікової контрольної роботи. Залікова контрольна робота складається з трьох питань, максимальна кількість можливих балів 100 за залікову контрольну роботу рівномірно розподіляється між цими питаннями. Здобувач отримує більшу з оцінок, що отримані за результатами залікової контрольної роботи або за рейтингом.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

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 старшим викладачем, Сімоненко А. В.

Ухвалено кафедрою Обчислювальної техніки (протокол № 10 від 25.05.2022)

Погоджено Методичною комісією факультету (протокол № 10 від 09.06.2022)