

**ТАВРІЙСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ В. І. ВЕРНАДСЬКОГО**
Навчально-науковий інститут муніципального управління
та міського господарства
Кафедра комп'ютерних та інформаційних технологій



ЗАТВЕРДЖУЮ

Директор інституту
Володимир КИСЕЛЬОВ

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

СИСТЕМНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

першого (бакалаврського) рівня вищої освіти

Галузь знань: 12 «Інформаційні технології»

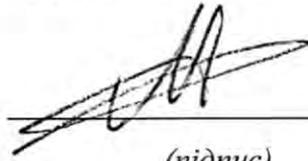
Спеціальність: 122 «Комп'ютерні науки»

Освітньо–професійна програма: «Комп'ютерні науки»

Форма здобуття освіти: очна (денна)

КИЇВ 2025

УКЛАДАЧ силабусу к.т.н., доцент, доцент кафедри комп'ютерних та інформаційних технологій

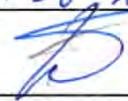


Андрій МОШЕНСЬКИЙ

(підпис)

Розглянуто та схвалено на засіданні кафедри комп'ютерних та інформаційних технологій

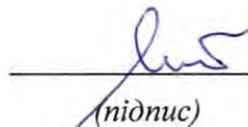
Протокол № 1 від 26.08.2025

Завідувач кафедри  Олександр ГУЙДА

(підпис)

Гарант освітньо-професійної програми «Комп'ютерні науки»

к.т.н., доцент



Сергій ЛІСОВЕЦЬ

(підпис)

1. Загальна інформація про навчальну дисципліну	
1. Назва навчальної дисципліни, код в ОПП	СИСТЕМНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ Код 1.2.14
2. Статус навчальної дисципліни	Навчальна дисципліна Професійної підготовки
3. Рік навчання, семестр у якому викладається дисципліна	3 рік навчання, 5 семестр
4. Обсяг навчальної дисципліни (кількість кредитів, загальна кількість годин (аудиторних за видами занять, самостійної роботи здобувача вищої освіти)	4 кредити загальна кількість годин: 120 год аудиторних: 44 год лекцій: 16 год практичні: 28 год самостійна робота: 76 год
5. Вид підсумкового (семестрового) контролю	екзамен
6. Інформація про консультації	За графіком проведення консультацій викладачами кафедри
7. Мова викладання	українська
8. Прізвище, ім'я, по батькові викладача (науковий ступінь, вчене звання, посада)	к.т.н., доцент, доцент кафедри комп'ютерних та інформаційних технологій А.О. Мошенський
9. Контактна інформація викладача	moshenskyi.andrii@tnu.edu.ua
10. Посилання на силабус на веб-сайті Університету	https://tnu.edu.ua/
2. Опис навчальної дисципліни	
Анотація дисципліни	Навчальна дисципліна «Системне програмне забезпечення» є складовою циклу професійної підготовки фахівців освітньо-кваліфікаційного рівня «бакалавр». Курс підпорядковано розробці ефективної конфігурації вбудованої системи Linux для створення середовища для запуску прикладних програм; застосування принципів розробки модулів ядра Linux для організації взаємодії з апаратними засобами системи.

<p>Мета, завдання та цілі вивчення дисципліни</p>	<p><u>мета дисципліни</u> необхідність формування у студентів знань примітивів виконання та синхронізації в ядрі Linux, деталі їх реалізації, вплив на швидкодію системи; алгоритмів функціонування різних типів модулів пристроїв в ядрі Linux, способи їх використання для організації взаємодії з системним програмним забезпеченням; особливостей взаємодії з апаратними засобами спеціалізованих систем.</p> <p><u>завдання і цілі дисципліни</u> дисципліна сприяє формуванню алгоритмічного мислення майбутнього фахівця, створює базу, яка необхідна при вивченні багатьох наступних дисциплін. Виходячи з цього викладання дисципліни «Системне програмне забезпечення» підпорядковане вирішенню таких основних задач, як з'ясування концептуальних принципів проектування ОС Linux та її вбудовані та серверні програми, що є критично важливими компонентами сьогодишньої ключової програмної інфраструктури в децентралізованому та мережевому всесвіті.</p>
<p>Пререквізити</p>	<p>Дослідження операцій, Програмування</p>
<p>Постреквізити</p>	<p>Системне програмування, Організація баз даних</p>
<p>Формат проведення дисципліни</p>	<p>Змішаний; У разі роботи в дистанційному режимі використовується корпоративне середовище Google Classroom; Лекції та практичні у дистанційному режимі будуть вестися через Google Meet; поточна комунікація з викладачем здійснюється корпоративною поштою.</p>

3. Перелік компетентностей, яких набувають здобувачі вищої освіти при вивченні навчальної дисципліни.

ІК. Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК8. Здатність генерувати нові ідеї (креативність)

ЗК10. Здатність бути критичним і самокритичним.

ЗК11. Здатність приймати обґрунтовані рішення.

ЗК16. Здатність розробляти й управляти проектами.

ФКЗ. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

ФК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

4. Програмні результати навчання відповідно до освітньо-професійної програми.

ПРН8. Використовувати методологію системного аналізу об'єктів, процесів і систем для задач аналізу, прогнозування, управління та проектування динамічних процесів в макроекономічних, технічних, технологічних і фінансових об'єктах.

ПРН9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

ПРН13. Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення.

5. Тематика та зміст навчальної дисципліни

Номер та назва розділу теми, перелік основних питань	Вид навчального заняття	Форми і методи контролю знань	Кількість годин Лекція/ практичне заняття
Модуль 1 – Система Linux			
<p>Тема 1. Архітектура системи Linux. Технічні вимоги. Linux і операційна система Unix. Три стандартні канали input/output. Зручне комбінування інструментів. Модульна конструкція, призначена для використання іншими користувачами. Псевдокод. Архітектура системи Linux. Application Binary Interface. Доступ до вмісту реєстру через вбудовану збірку. Доступ до вмісту реєстра управління через вбудовану збірку. Рівні привілеїв ЦП. Рівні привілеїв або кільця на x86. Архітектура Linux. Бібліотеки. Системні виклики. Контексти виконання в ядрі. Контекст процесу. Контекст переривання.</p>	Лекція/ практичне заняття	усне опитування/розв'язок задач за темою	4/4
<p>Тема 2. Віртуальна пам'ять. Технічні вимоги. Віртуальна пам'ять. Рішення 1 - спрощений помилковий підхід. Непряме звернення. Переклад адрес. Переваги використання VM. Ізоляція процесу. Захист області пам'яті. Тестування програми memsru(). Структура пам'яті процесу. Сегменти або зіставлення. Текстовий сегмент. Сегменти даних. Сегменти бібліотеки. Сегмент стека.</p>		усне опитування/розв'язок задач за темою	4/4

<p>Тема 3. Обмеження ресурсів. Ліміти ресурсів. Деталізація лімітів ресурсів. Типи ресурсів. Доступні ліміти ресурсів. Жорсткі і м'які обмеження. Запит і зміна значень ліміту ресурсів. Застереження. Утиліта prlimit. Використання prlimit(1). API-інтерфейси. Приклади коду.</p>		<p>усне опитування/розв'язок задач за темою</p>	<p>2/6</p>
<p>Модуль 2 - Розподіл пам'яті</p>			
<p>Тема 4. Розподіл динамічної пам'яті. Сімейство API glibc malloc (3). API-інтерфейс malloc (3). malloc (3) - деякі поширені запитання. malloc (3) - короткий опис. Calloc API. Realloc API. API-інтерфейс realloc array. Використання sbrk() API. Як насправді поводить себе malloc (3). Приклад коду - malloc(3). Резидент. Блокування пам'яті. Ліміти і привілеї. Блокування всіх сторінок. Захист пам'яті. Захист пам'яті - приклад коду. Журнали LSM, Ftrace. Експеримент: запуск програми memprot на ARM-32. Ключі захисту пам'яті. Використання alloca для автоматичного виділення пам'яті.</p>	<p>Лекція/ практичне заняття</p>	<p>усне опитування/розв'язок задач за темою</p>	<p>2/4</p>
<p>Тема 5. Проблеми з пам'яттю в Linux. Загальні проблеми з пам'яттю. Неправильний доступ до пам'яті. Доступ і/або використання неініціалізованих змінних. Тестовий приклад 1: не ініціалізований доступ до пам'яті. Доступ до пам'яті поза межами. Тестовий набір. Тестові приклади. Помилки використання після звільнення/використання після повернення. Вітік.</p>		<p>усне опитування/розв'язок задач за темою</p>	<p>2/4</p>

Невизначена поведінка. Фрагментація.			
Тема 6. Інструменти налагодження для проблем з пам'яттю. Типи інструментів. Valgrind. Використання інструменту Memcheck від Valgrind. Зведена таблиця Valgrind. Valgrind за і проти: короткий опис. Збірка програм для використання з ASan. Запуск тестових прикладів з ASan. Зведена таблиця AddressSanitizer (ASan). AddressSanitizer за і проти. Параметри Malloc. Покриття коду при тестуванні.		усне опитування/розв'язок задач за темою	2/6

6. Інформація про індивідуальне завдання

Курс Системне програмне забезпечення передбачає виконання індивідуальних завдань у вигляді самостійної роботи.

Провідна мета організації самостійної роботи полягає у необхідності широкого огляду тематики курсу з використанням основної та додаткової літератури, набуття навичок пошуку необхідної інформації, її аналітичного осмислення.

В процесі цієї роботи студенти повинні навчитися робити узагальнюючі висновки, оформляти результати роботи та планувати свою діяльність по вивченню дисципліни.

Контроль за самостійною роботою студентів – поточний контроль, тестування, контрольна робота.

Завдання контрольної роботи для студентів заочної форми навчання містять індивідуальні завдання для кожного студента.

Докладна інформація щодо змісту, варіантів завдань, порядку оформлення та захисту контрольної роботи міститься в методичних вказівках.

7. Технічне обладнання та програмне забезпечення

У звичайному режимі навчання вивчення навчальної дисципліни передбачає приєднання кожного здобувача до навчального середовища Google

Classroom, оскільки там розміщуються навчальні матеріали, проводиться тестування, ведеться журнал оцінювання навчальних досягнень.

У режимі дистанційного навчання - вивчення курсу додатково передбачає приєднання кожного здобувача вищої освіти до програм Google Meet (для занять у режимі відеоконференцій).

Для комунікації та опитувань, виконання домашніх завдань, виконання завдань самостійної роботи, проходження тестування (поточний, підсумковий контроль) тощо, здобувачу пропонується самостійно потурбуватися про якість доступу до інтернету, ноутбук або персональний комп'ютер, мобільний пристрій (телефон, планшет) з підключенням до Інтернет.

8. Політика дисципліни.

Політика щодо академічної доброчесності: списування під час тесту, іспиту заборонені.

Жодні форми порушення академічної доброчесності не толеруються. У випадку таких подій – реагування відповідно до Методичних рекомендацій для закладів вищої освіти з підтримки принципів академічної доброчесності.

Правила перезарахування кредитів у випадку мобільності, правила перескладання або відпрацювання пропущених занять тощо: відбувається згідно з Положення про організацію освітнього процесу у Таврійському національному університет ім. В.І. Вернадського.

Політика щодо дедлайнів та перескладання: роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку (до - 50% від можливої максимальної кількості балів за вид діяльності).

Перескладання тесту відбувається за наявності поважних причин (наприклад, лікарняний).

Політика щодо відвідування: відвідування занять є обов'язковим компонентом. За об'єктивних причин (наприклад, хвороба, працевлаштування, міжнародне стажування) навчання може відбуватись в онлайн-формі за погодженням.

9. Система оцінювання та вимоги

З дисципліни ЗВО (здобувач вищої освіти) може набрати до 60% підсумкової оцінки за виконання всіх видів робіт, що виконуються протягом семестру і до 40% підсумкової оцінки – на екзамені.

Поточний контроль проводиться шляхом спілкування із ЗВО під час лекцій та консультацій та опитувань.

Результати поточного контролю за відповідний модуль оприлюднюються викладачем на наступному аудиторному занятті. Бали, які

набрані ЗВО під час аудиторних занять, складають оцінку поточного контролю.

Семестровий контроль у вигляді екзамену проводиться під час сесії з трьома практичними завданнями(40 балів максимум). Оцінка за результатами вивчення дисципліни формується шляхом додавання підсумкових результатів поточного контролю до екзаменаційної оцінки. Взаємозв'язок між набраними балами і оцінкою наведено у розділі 9.1.

Приклади екзаменаційного білету знаходяться у пакеті документів на дисципліну.

У випадку, якщо ЗВО протягом семестру не виконав в повному обсязі передбачених робочою програмою всіх видів навчальної роботи, має невідпрацьовані роботи або не набрав мінімально необхідну кількість балів (20), він не допускається до складання екзамену під час сесії, але має право ліквідувати академічну заборгованість.

Повторне складання екзамену з метою підвищення позитивної оцінки не дозволяється.

9.1.Шкала та схема формування підсумкової оцінки

Модулі		Сума
Модуль 1	Модуль 2	60 балів
30 балів	30 балів	
Підсумковий контроль		40 балів
Максимальна сума балів		100 балів

Загальна система оцінювання курсу	Підсумкова оцінка з дисципліни є сумою оцінок з відповідною вагою за кожен з таких видів робіт: активна робота на лабораторних та практичних заняттях, тести та підсумковий контроль (екзамен). Підсумкова оцінка визначається відповідно до поданої нижче таблиці оцінювання за різними шкалами (100-бальна, ECTS, національна).
Розрахункова графічна-робота	В рамках курсу не передбачено виконання РГР.
Лабораторні та практичні роботи	Критерії оцінювання лабораторних та практичних робіт: 1. Підготовленість до лабораторних/практичних занять 2. Самостійність виконання лабораторних/практичних робіт.

	<p>3. Повнота виконання завдань</p> <p>4. Своєчасність виконання та захисту лабораторних/практичних робіт</p> <p>Максимальний бал за кожну лабораторну/практичну роботу – 5 балів</p>
Тест	Проміжний тест проводиться по кожному розділу курсу та оцінюється максимально в 10 балів.
Екзамен	Екзамен проводиться в кінці курсу, включає три практичних завдання. Максимально оцінюється в 40 балів.
Умови допуску до підсумкового контролю	Позитивна оцінка за всіма обов'язковими видами робіт (лабораторні та практичні роботи)

10.Рекомендовані джерела інформації

Назва теми	Рекомендовані джерела інформації до теми (основна література; допоміжна література; інформаційні ресурси в мережі Інтернет)
Модуль 1 – Система Linux	
<p>Тема 1. Архітектура системи Linux. Технічні вимоги. Linux і операційна система Unix. Три стандартні канали input/output. Зручне комбінування інструментів. Модульна конструкція, призначена для використання іншими користувачами. Псевдокод. Архітектура системи Linux. Application Binary Interface. Доступ до вмісту реєстру через вбудовану збірку. Доступ до вмісту реєстра управління через вбудовану збірку. Рівні привілеїв ЦП. Рівні привілеїв або кільця на x86. Архітектура Linux. Бібліотеки. Системні виклики. Контексти виконання в ядрі. Контекст процесу. Контекст переривання.</p> <p>Тема 2. Віртуальна пам'ять. Технічні вимоги. Віртуальна пам'ять. Рішення 1 - спрощений помилковий підхід. Непряме звернення. Переклад адрес. Переваги використання VM. Ізоляція процесу. Захист області пам'яті. Тестування програми memsru(). Структура пам'яті процесу. Сегменти або зіставлення. Текстовий сегмент. Сегменти даних.</p>	<p>Основна література: 1-4</p> <p>Додаткова література: 1-5</p>

<p>Сегменти бібліотеки. Сегмент стека. Тема 3. Обмеження ресурсів. Ліміти ресурсів. Деталізація лімітів ресурсів. Типи ресурсів. Доступні ліміти ресурсів. Жорсткі і м'які обмеження. Запит і зміна значень ліміту ресурсів. Застереження. Утиліта prlimit. Використання prlimit(1). API-інтерфейси. Приклади коду.</p>	
<p>Модуль 2 - Розподіл пам'яті</p>	
<p>Тема 4. Розподіл динамічної пам'яті. Сімейство API glibc malloc (3). API-інтерфейс malloc (3). malloc (3) - деякі поширені запитання. malloc (3) - короткий опис. Calloc API. Realloc API. API-інтерфейс realloc array. Використання sbrk() API. Як насправді поводить себе malloc (3). Приклад коду - malloc(3). Резидент. Блокування пам'яті. Ліміти і привілеї. Блокування всіх сторінок. Захист пам'яті. Захист пам'яті - приклад коду. Журнали LSM, Ftrace. Експеримент: запуск програми memprot на ARM-32. Ключі захисту пам'яті. Використання alloca для автоматичного виділення пам'яті.</p> <p>Тема 5. Проблеми з пам'яттю в Linux. Загальні проблеми з пам'яттю. Неправильний доступ до пам'яті. Доступ і/або використання неініціалізованих змінних. Тестовий приклад 1: не ініціалізований доступ до пам'яті. Доступ до пам'яті поза межами. Тестовий набір. Тестові приклади. Помилки використання після звільнення/використання після повернення. Витік. Невизначена поведінка. Фрагментація.</p> <p>Тема 6. Інструменти налагодження для проблем з пам'яттю. Типи інструментів. Valgrind. Використання інструменту Memcheck від Valgrind. Зведена таблиця Valgrind. Valgrind за і проти: короткий опис. Збірка програм для використання з ASan. Запуск тестових прикладів з ASan. Зведена таблиця AddressSanitizer (ASan). AddressSanitizer за і проти. Параметри Malloc. Покриття коду при тестуванні.</p>	<p>Основна література: 1-4 Додаткова література: 1,2,5</p>

Основна література

- Jonathan Corbet, Alessandro Rubini, and Greg Kroah-Hartman. Linux Device Drivers, Third Edition - 2005, O'Reilly Media, Inc. - ISBN: 0-596-00590-3
- Ulrich Drepper. What Every Programmer Should Know About Memory - 2007 - <https://people.freebsd.org/~lstewart/articles/cpumemory.pdf>
- Paul E. McKenney. Is Parallel Programming Hard, And, If So, What Can You Do About It? - Linux Technology Center, 2017 - <https://mirrors.edge.kernel.org/pub/linux/kernel/people/paulmck/perfbook/perfbook.html>

4. Andrew Sloss, Dominic Symes, Chris Wright. ARM System Developer's Guide: Designing and Optimizing System Software. - Morgan Kaufmann; 2004 - ISBN-13: 978-1558608740

Додаткова література

1. Mastering Linux Kernel Development by Raghu Bharadwaj.
2. The Art of Linux Kernel Design Illustrating the Operating System Design Principle and Implementation by Lixiang Yang.
3. Hands On: C Programming and Unix Application Design: UNIX System Calls and Subroutines using C c A. D. Marshall 1998-2004.
4. Practical Linux Security Cookbook - Second Edition. Tajinder Kalsi ISBN: 978-1-78913-839-9.
5. Hands-On Linux Administration on Azure Frederik Vos ISBN: 978-1-78913-096-6