

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ
КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ПРОГРАМНИХ СИСТЕМ**

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи

_____ Кашпур О.Ф.

«__» _____ 2019 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
СУЧАСНІ ОПЕРАЦІЙНІ СИСТЕМИ
для студентів**

галузь знань	12 Інформаційні технології
спеціальність	121 Інженерія програмного забезпечення
освітній рівень	магістр
освітня програма	Програмне забезпечення систем
вид дисципліни	обов'язкова

Форма навчання	денна
Навчальний рік	2021/2022
Семестр	3
Кількість кредитів ECTS	5
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладач: **к. ф.-м. н., доцент Ченцов О.І.** (лекції, лабораторні заняття).

Пролонговано: на 20__/20__ н. р. _____ (_____) «__» 20__ р.

на 20__/20__ н. р. _____ (_____) «__» 20__ р.

Розробник: Ченцов Олексій Ілліч, к. ф.-м. н., доцент, доцент кафедри інтелектуальних програмних систем.

ЗАТВЕРДЖЕНО

Завідувач кафедри інтелектуальних програмних систем

_____ О.І. Провотар

Протокол № __ від «__» _____ 2019 р.

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від «__» _____ 2019 року №__

Голова науково-методичної комісії _____ Л.Л. Омельчук

«__» _____ 2019 року

Затверджено вченою радою факультету комп'ютерних наук та кібернетики

Протокол від «__» _____ 2019 року №__

Голова вченої ради факультету _____ А.В. Анісімов

1. Мета дисципліни – всебічний розгляд принципів розробки та реалізації сучасних операційних систем, базових алгоритмів управління апаратними та віртуальними ресурсами. Розуміння характерних спільних рис та відмінностей сучасних операційних систем, тенденцій подальшого розвитку системного програмного забезпечення.

2. Попередні вимоги до опанування навчальної дисципліни. Для успішного вивчення дисципліни «Сучасні операційні системи» студенти повинні відповідати наступним вимогам:

1. **Мати уявлення:** про архітектуру сучасної обчислювальної техніки.
2. **Знати:** загальні відомості з галузі знань про ОС та з системного програмування, окремі питання в галузі розподілених обчислень.
3. **Вміти:** читати код на мові програмування С.
4. **Володіти елементарними навичками:** налагодження та тестування програмного забезпечення.

3. Анотація навчальної дисципліни. Навчальна дисципліна «Сучасні операційні системи» є складовою освітньо-наукової програми підготовки фахівців за другим (магістерським) рівнем вищої освіти у галузі знань 12 Інформаційні технології за спеціальністю 121 Інженерія програмного забезпечення в рамках освітньо-наукової програми «Програмне забезпечення систем».

Дана дисципліна є нормативною навчальною дисципліною в рамках освітньої програми «Програмне забезпечення систем». Викладається у 3 семестрі в **обсязі – 150 год. (5 кредитів ECTS)** зокрема: лекції – 24 год., лабораторні заняття – 24 год., самостійна робота – 100 год., консультації – 2 год. У курсі передбачено 2 змістовні частини та 1 контрольна робота. Завершується дисципліна – **іспитом**.

В результаті вивчення навчальної дисципліни студент повинен:

знати: принципи проектування та реалізації сучасних операційних систем, структуру таких ОС, типові алгоритми, що в них використовуються, віртуальні ресурси та абстракції, що ними створюються, а також нові нестандартні підходи до побудови операційних систем.

вміти: ефективно працювати із вихідним кодом сучасних операційних систем, робити їх збірку, реалізовувати драйвери та розширення сучасних операційних системи, ефективно використовувати інструментарій адміністрування.

4. Завдання (навчальні цілі). Основними завданнями дисципліни «Сучасні операційні системи» є набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень в області операційних систем відповідно до освітньої кваліфікації магістр з інженерії програмного забезпечення. Зокрема, розвивати:

- Здатність проведення теоретичних та прикладних досліджень на відповідному рівні (ЗК03).
- Здатність проектувати програмне забезпечення включаючи проведення моделювання його архітектури, поведінки та процесів функціонування окремих підсистем і модулів (СК03).
- Здатність розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення (СК04).

5. Результати навчання за дисципліною.

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
РН1.1	Знати структуру, цільові системні характеристики та спосіб їх забезпечення у сучасних ОС.	Лекції.	Контрольна робота, 60% балів, іспит.	20%
РН1.2	Знати ефективні алгоритми управління системними ресурсами, основні способи віртуалізації ресурсів.	Лекції.	Контрольна робота, 60% балів, іспит.	36%
РН2.1	Вміти конфігурувати, збирати ядро, адмініструвати сучасні ОС.	Лабораторні заняття, самостійна робота.	Захист лабораторної роботи, іспит.	18%
РН2.2	Вміти ефективно працювати з вихідним кодом операційної системи, розробляти розширення.	Лабораторні заняття, самостійна робота.	Захист лабораторної роботи, іспит.	18%
РН4.1	Ініціювання комплексних проектів, лідерство під час їх реалізації.	Самостійна робота.	Захист лабораторної роботи.	8%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання.

Програмні результати навчання	Результати навчання дисципліни				
	РН1.1	РН1.2	РН2.1	РН2.2	РН4.1
ПРН06. Аналізувати, оцінювати і обирати методи, сучасні програмно-апаратні інструментальні та обчислювальні засоби, технології, алгоритмічні та програмні рішення для ефективного виконання конкретних виробничих задач з програмної інженерії	+	+		+	
ПРН07. Обґрунтовано обирати парадигми і мови програмування для вирішення прикладних завдань; застосовувати на практиці системні та спеціалізовані засоби, компонентні технології (платформи) та інтегровані середовища розробки програмного забезпечення.			+	+	+

7. Схема формування оцінки.

7.1 Форми оцінювання студентів.

Семестрове оцінювання:

1. Контрольна робота: РН1.1, РН1.2 – **24 балів/12 балів**.
2. Лабораторні роботи 1-6: РН2.1, РН2.2, РН4.1 – **36 балів/18 балів**.
3. Допускна контрольна робота: РН4.1 – зараховано-незараховано/60 %.

Підсумкове оцінювання (у формі іспиту):

- Максимальна кількість балів які можуть бути отримані студентом: 40 балів.
- Результати навчання які будуть оцінюватись: РН1.1, РН1.2, РН2.1, РН2.2.
- Форма проведення і види завдань: письмова робота.
- Види завдань: 5 письмових завдань.

Студенти, що **не виконали** допускну контрольну роботу, відповідають на **два додаткових письмових завдання** з тематики загальних знань про операційні системи.

Студент **не допускається** до іспиту, якщо під час семестру набрав **менше ніж 20 балів**.

7.2 Організація оцінювання.

Терміни проведення форм оцінювання:

1. Допускна контрольна робота: до 3 тижня семестру.
2. Контрольна робота: до 12 тижня семестру.
3. Лабораторна робота 1: до 2 тижня семестру.
4. Лабораторна робота 2: до 4 тижня семестру.
5. Лабораторна робота 3: до 6 тижня семестру.
6. Лабораторна робота 4: до 8 тижня семестру.
7. Лабораторна робота 5: до 10 тижня семестру.
8. Лабораторна робота 6: до 12 тижня семестру.

На початку семестру студенти пишуть допускну контрольну роботу з загальних відомостей про операційні системи, що включено у попередні вимоги дисципліни. Написання відповідей на допускну контрольну роботу виконується у рамках самостійної роботи. Оцінка допускної роботи має пороговий характер. Студенти, які не набрали за роботу хоча б 60 зі 100 балів, на іспиті відповідають на 2 додаткових питання з зазначеної тематики.

Якщо студент з поважних причин, які підтверджено документально, був відсутній при написанні контрольної роботи, він має право на одне перескладання з можливістю отримання максимальної кількості балів. Термін перескладання визначається викладачем.

Студент має право на одне перескладання кожної контрольної роботи із можливістю отримання максимумо 80% початково визначених за цю контрольну роботу балів. Термін перескладання визначається викладачем.

У разі неякісного виконання лабораторної роботи, викладач має право не зарахувати лабораторну роботу, або знизити за неї бали.

Студент має право здавати лабораторні роботи після закінчення визначеного для них терміну, але з втратою одного балу за кожен тиждень, який пройшов з моменту закінчення терміну її здачі.

7.3 Шкала відповідності оцінок.

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59

8. Структура навчальної дисципліни. Тематичний план лекцій і лабораторних занять.

№ лекції	Назва теми/лекції	Кількість годин		
		Лекції	Лабораторні заняття	Самостійна робота
Частина 1. Операційна система Linux.				
1	Тема 1. Загальний огляд ОС Linux.	2		12
2	Тема 2. Методологія роботи із кодом ядра ОС Linux. Лекція 2. Основні обчислювальні структури ядра Linux	2	2	6
3	Тема 2. Методологія роботи із кодом ядра ОС Linux. Лекція 3. Фреймворк локальних змінних процесора	2	2	6
4	Тема 3. Процеси та потоки в ОС Linux	2	2	6
5	Тема 3. Процеси та потоки в ОС Linux Лекція 5. Підсистема ідентифікаторів процесів Linux	2	2	6
6	Тема 4. Планувальник ОС Linux. Лекція 6. Справедливий планувальник ОС Linux	2	2	6
7	Тема 4. Планувальник ОС Linux. Лекція 7. Допоміжні планувальники ОС Linux	2	2	6
Всього по частині 1		14	12	48
Частина 2. Операційна система Windows.				
8	Тема 5. Загальний огляд ОС Windows. Лекція 8. Загальні відомості щодо ОС Windows	2	2	6
9	Тема 5. Загальний огляд ОС Windows. Лекція 9. Структура ОС Windows	2	2	6
10	Тема 6. Системні механізми ОС Windows. Лекція 10. Класифікація пасток та обробка переривань	2	2	10
11	Тема 6. Системні механізми ОС Windows. Лекція 11. Синхронні пастки в ОС Windows	2	2	10
12	Тема 7. Нові підходи до побудови ОС	2	2	20

Контрольна робота		2	
Всього по частині 2	10	12	52
Консультації	2		
ВСЬОГО	24	24	100

Загальний обсяг – **150** год., в тому числі:

Лекції – **24** год.

Лабораторні заняття – **24** год.

Самостійна робота – **100** год.

Консультації – **2** год.

Теми, винесені на самостійне вивчення.

1. Фреймворк потоків в ядрі Linux.
2. Групування задач в Linux.
3. Переривання, що ініціюються повідомленням.
4. Асинхронні процедурні виклики.
5. Моделі драйверів у Windows.
6. Управління ресурсами та організація об'єктів у Windows.
7. Гібридні операційні системи на прикладах XNU, Hurd.

Умови лабораторних робіт:

- **Лабораторна робота 1:** Організація віддаленого робочого столу.
- **Лабораторна робота 2:** Інструментарій віртуальних машин.
- **Лабораторна робота 3:** Відстеження організації системних викликів. Фонові процеси.
- **Лабораторна робота 4:** Організація коду ядра та збирання ядра Linux.
- **Лабораторна робота 5:** Розробка розширення ядра Linux.
- **Лабораторна робота 6:** Емпіричне дослідження часових затримок та параметрів планувальника.

Деталізовані умови лабораторних робіт розміщено за посиланням:

https://drive.google.com/drive/folders/0B-BUpwNPP_9JZVRXV2IUcmRQdlk

9. Рекомендовані джерела.

Основні:

1. Tanenbaum A. Modern Operating Systems, 4th ed. / Tanenbaum A., Bos H. – Pearson, 2014. – 1136 p.
2. Russinovich M. Windows Internals, part 1, 6th ed. / Russinovich M., Solomon D., Ionescu A. – Microsoft Press, 2012. – 752 p.
3. Russinovich M. Windows Internals, part 1, 6th ed. / Russinovich M., Solomon D., Ionescu A. – Microsoft Press, 2012. – 672 p.
4. Mauerer W. Professional Linux c Kernel Architecture / Mauerer W. – Indianapolis: Wiley publishing, 2008. – 1368 p.
5. Love R. Linux Kernel Development, 3rd ed. / Love R. – Addison-Wesley, 2010. – 440 p.

6. Cesati M. Understanding the Linux Kernel, 3rd Ed. / Bovet D., Cesati M. – O'Reilly, 2005. – 944 p.

Додаткові:

1. Clement G. Linux Kernel and Driver Development Training / Clement G. et al [Online]. – Free-electrons, 2004-2017. – Available: <http://free-electrons.com/doc/training/linux-kernel/>
2. Greg Kroah-Hartman. Linux Kernel in a Nutshell. – O'Reilly Media, 2009. – 198 p.
3. Таненбаум Э. Современные операционные системы, 4-е изд. / Таненбаум Э., Бос Х. – СПб.: Питер, 2015. – 1120 с.
4. Руссинович М. Внутреннее устройство Microsoft Windows, 6-е изд. / Руссинович М., Соломон Д., Ионеску А. – СПб.: Питер, 2013. – 800 с.
5. Таненбаум Э. Операционные системы. Разработка и реализация. 3-е изд. / Таненбаум Э., Вудхалл А. – СПб.: Питер, 2007. – 704 с.
6. Лав Р. Разработка ядра Linux / Лав Р. – Вильямс, 2006. – 448 с.
7. Чезати М. Ядро Linux / Бовет Д., Чезати М. – БХВ-Петербург, 2007. – 1104 с.
8. Шоу А. Логическое проектирование операционных систем / Шоу А. – М.Мир, 1981. – 360 с
9. Wirth N. Good Ideas, through the Looking Glass / Niclaus Wirth // Computer. – V. 39 (2006) – no.1 – Pp. 28-39.
10. Hunt G. Singularity: rethinking the software stack / Galen Hunt, James Larus // ACM SIGOPS Operating Systems Review. – V.41. – no. 2. – Pp. 37-49.
11. Inside the Mac OS X Kernel. Debunking Mac OS Myths [Online]. – Available: http://events.ccc.de/congress/2007/Fahrplan/attachments/1053_inside-macosx-kernel.pdf.
12. Hallgren T. A principled approach to operating system construction in Haskell / Thomas Hallgren, Mark P. Jones, Rebekah Leslie, Andrew Tolmach // Proceedings of the tenth ACM SIGPLAN international conference on Functional programming (2005). – Pp. 116-128.
13. Diatchki I. Writing systems software in a functional language: an experience report / Iavor S. Diatchki, Thomas Hallgren, Mark P. Jones, Rebekah Leslie, and Andrew Tolmach // Proceedings of the Fourth Workshop on Programming Languages and Operating Systems (PLOS 2007).
14. Windows operating system internal curriculum resource kit [Online]. – Available: <http://www.microsoft.com/resources/sharedsource/windowsacademic/curriculumresourcekit.msp>

10. Додаткові ресурси.

Приклади контрольних завдань, тестових запитань, переліки контрольних запитань та тем, що виносяться на іспит розміщено за посиланням:

- https://drive.google.com/drive/folders/1vJyW1kSnW_ToCLPoqaBYmYXL7_9gIX8N