МІЖНАРОДНИЙ ЄВРОПЕЙСЬКИЙ УНІВЕРСИТЕТ

Навчально-науковий інститут
«Європейська школа бізнесу»

Кафедра інформаційних технологій

ОПЕРАЦІЙНІ СИСТЕМИ

Методичні рекомендації
до опанування лекційних занять
з навчальної дисципліни
для здобувачів вищої освіти спеціальності
F2 «Інженерія програмного забезпечення»

Київ – 2025

УДК 37.026:004.4:007

Рекомендовано до друку Вченою радою
Навчально-наукового інституту
«Європейська школа бізнесу»
(протокол № 1 від 26.08.2025)

Рецензент:
Федоров В.В., к.ф.-м.н.,
доцент кафедри інформаційних технологій МЄУ

Укладач:
Нестеренко О.В., д.т.н., професор,
професор кафедри інформаційних технологій МЄУ

Операційні системи: Методичні рекомендації до опанування лекційних занять з навчальної дисципліни для здобувачів вищої освіти спеціальності F2 «Інженерія програмного забезпечення» / Нестеренко О.В. Київ: МЄУ, 2025. 34 с.

Запропоноване видання узгоджене з навчальною програмою з дисципліни «Операційні системи» та може використовуватися як посібник для студентів, які розпочинають навчання на бакалавраті в галузі інформаційних технологій, зокрема за спеціальністю F2 «Інженерія програмного забезпечення».

Для студентів вищих навчальних закладів, які навчаються за напрямком підготовки з галузі знань „Інформаційні технології".

ЗМІСТ

ВСТУП..... 4
1. Завдання та структура навчальної дисципліни..... 6
2. Результати навчання..... 10
3. Методичні рекомендації до лекцій..... 13
4. Методичні рекомендації до самостійної роботи..... 19
5. Засоби діагностики, критерії та процедури оцінювання лекційних модулів..... 23
6. Глосарій..... 26
Література..... 33

ВСТУП

Головною ознакою сьогодення є повсюдне застосування інформаційних технологій і систем, динамічною складовою яких виступає програмне забезпечення. Розвиток обчислювальної техніки пов'язаний не тільки з вдосконаленням комп'ютерів та їх складових, але й з розвитком технологій програмування.

Тому в сучасних умовах актуальною є відносно нова для технологічного розвитку проблема, суть якої полягає у створенні усе більшої кількості програм, і, відповідно, новий технологічний процес, спрямований на вирішення проблеми керування багатьма програмами в єдиному комп'ютерному середовищі. Стрімке зростання об'ємів програм передусім пов'язане з розподілом пам'яті та інших ресурсів комп'ютера між багатьма програмними процесами. У зв'язку із цим в контексті цих проблем особливе значення набуває розуміння структури і алгоритмів функціонування сучасних операційних систем (ОС) та їх роль в технологіях розробки програмного забезпечення (ПЗ).

Отже, в теперішній час підтримка виконання програмних засобів – це складний автоматизований технологічний процес. Тому питання його організації, управління програмами в багатопроцесорному та розподіленому середовищі, а також методи та засоби планування, диспетчеризації та контролю використання комп'ютерних ресурсів є дуже важливими для досягнення мети отримання в процесі розробки якісного програмного продукту. Ці та інші дотичні питання вивчаються в дисципліні «Операційні системи».

Масштабні і складні зміни, які несе нова промислова революція, пов'язана з масовою роботизацією, злиттям технологій і стиранням граней між фізичними і цифровими сферами і, як наслідок, майбутня інтелектуалізація суспільства потребують переважання високопрофесійних фахівців-програмістів. Вони повинні не лише вміти працювати з комп'ютерним обладнанням та різноманітними програмними засобами, розуміти роль інформаційних технологій у

суспільній діяльності, а й бути обізнаними в сучасних технологіях керування програмними процесами, мати цілісний погляд на основи операційних систем та їх функціонування з метою створення якісних програмних продуктів.

Саме тому метою навчальної дисципліни «Операційні системи» є оволодіння студентами знаннями про основні принципи і алгоритми, покладені в основу розробки операційних систем, вивчення їх внутрішньої будови, набуття умінь і практичних навичок у роботі з основними засобами ОС, що використовуються в діяльності програмістів.

Внаслідок сучасних тенденції розвитку практики викладання у вищій освіті, пов'язаних із студентоцентрованим навчанням та викладанням як колективним процесом і розподілом відповідальності між викладачем та студентом, зростає активна роль студента та фокусування на його самостійності у процесі навчання – з одного боку, а з іншого – більшає значення вдосконалення та зросту якості дидактичних компетентностей викладача.

У зв'язку із цим у відповідь на зростаючі потреби в інноваціях та інтелектуалізації необхідно впроваджувати інноваційні змісти та моделі навчання. Враховуючи до того ж широке різноманіття тем, пов'язаних понять і визначень, теоретичних знань і практичних навичок, передбачених навчальною дисципліною «Операційні системи», вважається за доцільне надати студентам та викладачам додаткові методичні вказівки до опанування лекційних занять, що має сприяти більш швидкому зануренню у проблемну область і глибшому засвоєнню навчального матеріалу.

1. Завдання та структура навчальної дисципліни

Основними завданнями вивчення дисципліни «Операційні системи» є:

• ознайомити студентів з основними архітектурними рішеннями побудови операційних систем, з різними видами операційних систем та їх архітектурними рішеннями;

• оволодіння студентами знання про основні принципи, концепції та алгоритми, покладені в основу розробки операційних систем;

• вивчення внутрішньої будови ОС, набуття умінь і практичних навичок у роботі з основними засобами операційних систем, що використовуються в діяльності програмістів;

• розширити знання студентів про керування роботою програм, як прикладних, так і системних, про інтерфейс між програмами і апаратним забезпеченням комп'ютера;

• розкрити місце і значення операційних систем в розвитку інформаційних технологій і систем..

З завдань випливає мета викладання навчальної дисципліни «Основи програмної інженерії», яка полягає у формуванні у студентів здатностей оволодіння знаннями про принципи та процеси розробки програмних продуктів та програмних систем, сучасні методології та практики у розробці програмного забезпечення, набуття умінь і практичних навичок щодо використання основних засобів організації процесу розробки програмного забезпечення.

На вивчення навчальної дисципліни «Операційні системи» зазвичай відводиться 120 годин 4 кредити ЄКТС. Обсяг дисципліни поділено на два розділи (рис. 1, 2). Перелік розділів і тем дисципліни наведено в табл. 1. При викладанні навчальної дисципліни «Операційні системи» крім інформаційних методів навчання, таких як класичні лекції, навчальним планом передбачено час на самостійну роботу

студентів. Завдання, що виносяться на самостійне опрацювання, наведені в табл. 2. Рекомендовані матеріали розміщено на платформі Distance Learning МЄУ

Рис. 1. Структура першого розділу навчальної дисципліни

Рис. 2. Структура другого розділу навчальної дисципліни

Таблиця 1

Розділи і теми навчальної дисципліни «Операційні системи»

№ з/п	Тема	Перелік основних питань
РОЗДІЛ 1. ОСНОВНІ ФУНКЦІЇ ОС		
1	1. Функції та архітектура операційних систем	Функції, класифікації і складові ОС. Структура операційних систем. Системні виклики. Поняття операційного і програмного середовища. Архітектура операційних систем (windows, unix і linux, android). Основні етапи розвитку операційних систем.
2	2. Процеси і потоки	Основні відомості про процеси. Управління процесами. Взаємодія процесів. Примітиви синхронізації. Взаємні блокування. Класичні проблеми міжпроцесної взаємодії. Потоки. Сигнали
3	3. Управління пам'яттю в операційних системах	Загальні відомості. Технології розподілу пам'яті. Віртуальна пам'ять
4	4. Введення/ виведення та файлова система	Система введення/виведення. Файлова система. Поняття файлу. Логічна організація файлової системи
РОЗДІЛ 2. СПЕЦІАЛЬНІ ЗАСТОСУВАННЯ ОС		
5	5. Мульти-програмування	Основні визначення і характеристики. Тупики. Мультипрограмування на основі переривань
6	6. Паралельна робота	Організація паралельної роботи пристроїв і процесора. Багатопроцесорні системи. Мультикомп'ютери
7	7. Віртуалізація та хмари	Засади віртуалізації. Технології ефективної віртуалізації. Віртуалізація пам'яті та введення/виведення. Розподілені системи. Хмари як послуга. Міграція віртуальних машин
8	8. Безпека ОС	Загрози і зловмисники. Засади безпеки ОС. Керування доступом до ресурсів. Автентифікація. Методи злому. Зловредні програми. Методи і засоби захисту

Таблиця 2

Загальний перелік завдань, що виносяться на самостійне опрацювання

№ з/п	Теми	Завдання, що виноситься на самостійне опрацювання
1.	1.	Оглядовий реферат «Основні етапи та тенденції розвитку операційних систем»
2.	2	Алгоритм Деккера, Алгоритм Петерсона, Алгоритм розв'язання задачі про «філософів, що обідають».
3.	3	Адаптивний алгоритм заміщення кешу (ARC)
4.	4	Особливості файлових систем в UNIX та Linux
5.	5	Алгоритми для уникнення взаємних блокувань в системі
6.	6	Паралельне програмування на основі MPI
7.	7	Дослідження в сфері віртуалізації та хмар
8.	8	Дослідження в сфері віртуалізації та хмар для забезпечення безпеки

2. Результати навчання

Зміст лекційних занять з дисципліни «Операційні системи» містить навчальні елементи, необхідні для реалізації очікуваних результатів навчання, передбачених освітньою програмою спеціальності.

Згідно з вимогами стандарту освіти дисципліна має забезпечувати набуття студентами низки загальних та спеціальних (фахових, предметних) компетентностей .

На основі трансформації цих компетентностей у такі результати навчання, які студент може опанувати на лекції і під час самостійної роботи над лекційним матеріалом та продемонструвати рівень їх сформованості на контрольних заходах формуються дисциплінарні результати навчання (ДР) [1] (рис. 3). Дисциплінарні результати навчання наведено у табл. 3.

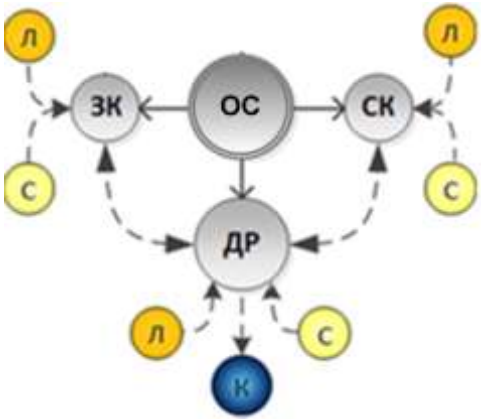

Рис. 3. Трансформація компетентностей і дисциплінарних результатів навчання дисципліни «Операційні системи» (ОС)
(Л – лекційні матеріали, С – самостійна робота, К – контрольні заходи)

Таблиця 3

Дисциплінарні результати навчання

№ з\п	Теми навчання	Знання та навички, що складають дисциплінарні результати навчання
1	2	3
Загальні компетентності:		
Здатність до абстрактного мислення, аналізу та синтезу		
1.	1	Усвідомлення операційної системи як абстрактного рівня керування ресурсами комп'ютера. Здатність формалізувати функції ОС через моделі управління процесами, пам'яттю, файлами та пристроями введення/виведення, до аналітичного мислення щодо внутрішніх механізмів ОС, абстрагування апаратного рівня засобами ОС та синтезу рішень на основі моделей ОС
Спеціальні (фахові, предметні) компетентності		
Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування		
2.	1, 3, 5, 6	Розуміння архітектури операційної системи як складової комплексу ПЗ, здатність пояснити структуру ОС у вигляді модулів (ядро, драйвери, менеджери ресурсів, оболонки) та інтерфейсів взаємодії між ними, моделювати взаємодію процесів через системні виклики, сигнали, пайпи або черги повідомлень, вміти формалізувати поведінку ОС за допомогою таблиць переходів, графів станів, мереж Петрі, моделювати паралельні та конкурентні процеси із фіксацією умов гонки, критичних секцій, семафорів та м'ютексів, використовувати мову програмування (наприклад, Python, C) для системного програмування в середовищі ОС в умовах, наближених до реальних
Здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки (в тому числі кібербезпеки)		
3.	8	Розпізнавати уразливості операційних систем, знати типові загрози безпеці ОС, аналізувати механізми захисту, реалізовані в ОС, застосовувати засоби забезпечення безпеки ОС, оцінюапти рівень захищеності ОС в контексті проєкту, дотримуватися принципів безпечного адміністрування ОС

Продовження табл. 3

1	2	3
Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення		
4.	1, 2, 4, 7	Знати як визначати роль операційної системи у підтримці процесів розробки та супроводу ПЗ, усвідомлювати, як ОС забезпечує середовище виконання для інструментів розробки (компілятори, збирачі, налагоджувачі тощо), вміти обґрунтувати вибір ОС (Linux, Windows, BSD тощо) залежно від вимог проєкту (інструменти, платформа, доступ до ресурсів, безпека, масштабованість), освоювати роботу з системним інструментарієм ОС, оцінювати ресурси системи (навантаження CPU, пам'яті, I/O), застосовувати середовища віртуалізації та контейнеризації (VirtualBox, VMware, WSL), демонструвати готовність до самостійного освоєння нових інструментів, формувати навички самостійного освоєння нових. Засобів, таких як хмарні технології та ін.

3. Методичні рекомендації до лекцій

Опанування лекційного матеріалу є найважливішим етапом навчання. Від того, як студент прослухав матеріал, зрозумів ідеї, які доносить до нього лектор, запам'ятав почуте залежить уся подальша робота з вивчення дисципліни, зокрема проведення самостійної роботи, практичних занять і підготовка до вивчення нових лекційних модулів.

В цьому процесі виняткову роль відіграє конспектування матеріалу лекцій. Важливо зафіксувати принаймні основне, що є в лекції – це визначення, логічні побудови, формулювання основних положень, висновки. В сучасних умовах ведення викладачами електронних ресурсів навчальних матеріалів, у тому числі і конспектів лекцій, які доступні студентству, дослівний запис лекцій втрачає необхідності. Але ведення окремого зошиту для навчальної дисципліни, фіксуючи основне з лекцій, передбачивши поля для подальших доповнень виписками з літературних джерел, власними міркуваннями, новими уявленнями добутими на основі наступних лекцій залишається вельми актуальним і корисним. При цьому варто зазначити, що ведення конспекту перетворює прослуховування лекції у цікаву роботу, а не відбування часу, адже це дисциплінує і сприяє зосередженості на лекції.

Ще одним чинником, який значною мірою впливає на сприйняття матеріалу лекцій є необізнаність і, як наслідок, невизначеність студента про загальний план дисципліни і взаємопов'язаність розділів і тем предмету що розглядається. Важливо з самого початку скласти чітке уявлення щодо зазначених питань.

Нижче на рис. 4 показано структурування матеріалу лекцій з точки зору ієрархії взаємозв'язків, а у табл. 4 наведено основні

елементи лекційних матеріалів, на які необхідно звернути особливої уваги при їх прослуховування та вивченні.

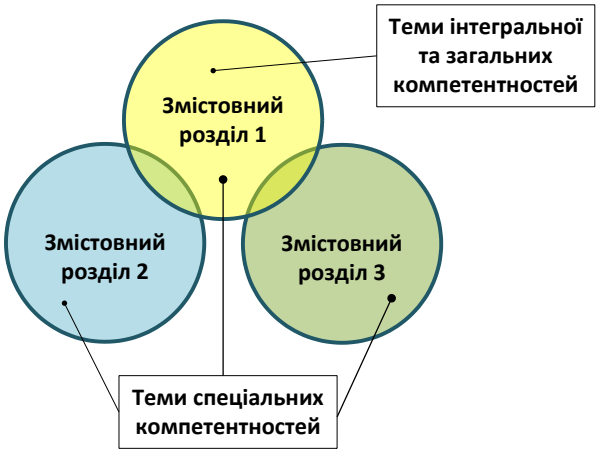

Рис. 4. Ієрархії взаємозв'язків матеріалів лекцій

Таблиця 4

Основні елементи лекцій з навчальної дисципліни

Теми лекції	Основні елементи	Літера-турні джерела
1	2	3
Лекція 1. Вступ до дисципліни. Функції та архітектура ОС		
1.	1.1. Функції, класифікації і складові ОС. Основні та додаткові функції. Сервіси, що надають типові ОС. 1.2. Структура операційних систем. Загальна структура. Монолітна, ієрархічна структура, структура мікроядра. 1.3. Системні виклики. Режим користувача та системний режим. Інтерфейс прикладного програмування 1.4. Поняття операційного і програмного середовища. 1.5. Архітектура операційних систем. Модулі і сорвіси. Функції ядра. Мікроядерна архітектура. Види ядр. Архітектура windows, unix і linux, android.	[2 - 12]

Продовження табл. 4

1	2	3
	Лекція 2. Процеси і потоки	
2.	2.1. Основні відомості про процеси. Управління процесами. Мультипрограмність. Створення процесу. Стани процесу. Моделі процесу. Опис процесів. Класифікація процесів. Стратегії планування і диспетчеризація процесів. Алгоритми диспетчеризації 2.2 Взаємодія процесів. Проблема взаємного виключення і способи її вирішення. Критичний ресурс. Критична секція. Апаратні та програмні способи досягнення взаємного виключення. Змінні блокування. Алгоритм Деккера. Алгоритм Петерсона 2.3 Примітиви синхронізації. Семафори. М'ютекс. Монітор. 2.4 Взаємні блокування. Ресурс. Повторно використовувані ресурси. Ресурси що витрачаються. Монопольний та Розділюваний режими доступу до ресурсу. Моделювання взаємних блокувань. Алгоритм банкіра. 2.5 Класичні проблеми міжпроцесної взаємодії. Проблема виробника і споживача. Задача про «читачів» і «письменників». Задача про «філософів, що обідають». Задача про «сплячого перукаря». 2.6 Потоки. Елементи, що використовуються потоками. Багатопоточність. 2.7 Сигнали. Функції сигналів. Обробка сигналів. Сигнали реального часу	[3, 4, 6, 9, 12 - 14]
	Лекція 3. Управління пам'яттю в операційних системах	
3	3.1. Загальні відомості. Ієрархія пам'яті в комп1ютері. Логічний та фізичний адресний простір. Пристрій управління пам'яттю. Схема адресації і перетворення логічної адреси у фізичну. Відкачування і підкачка. 3.2. Технології розподілу пам'яті. Функції мультипрограмної ОС з управління пам'яттю. Алгоритми розподілу пам'яті. Динамічний розподіл. Сторінкова організація. Сегментна організація. 3.3. Віртуальна пам'ять Передумови організації віртуальної пам'яті. Свопінг. Структура таблиці сторінок. Сегментація. Комбінація сегментації і сторінкової організації. Алгоритми управління віртуальною пам'яттю	[2, 4, 6, 9]

Продовження табл. 4

1	2	3
	Лекція 4. Введення/виведення та файлова система	
1.	4.1 Система введення/виведення. Основні принципи програмного забезпечення введення/виведення. Способи здійснення введення/виведення. Програмні рівні введення/ виведення. Драйвери пристроїв. Незалежне від пристроїв програмне забезпечення введення/виведення. 4.2 Файлова система. Поняття файлу. Фізична організація файлової системи. Структура файлу. Будова диску. Файлові системи FAT, NTFS. Фізична організація і адресація файлу на диску. Сектори і кластери. 4.3 Логічна організація файлової системи. Типи файлів. Каталоги. Спеціальні файли. Ієрархічна структура файлової системи. Імена файлів. Монтування. Атрибути файлів	[2, 4, 6, 9]
	Лекція 5. Мультипрограмування	
5	5.1 Основні визначення і характеристики. Системи пакетної обробки; системи поділу часу; системи реального часу. Мультипроцесорна обробка. Симетрична та асиметрична архітектура. 5.2 Тупики. Основні напрямки політики запобігання тупиків. Алгоритм запобігання тупикових ситуацій. Алгоритм банкіра. Алгоритм Лємпорта. Проблема розподілу ресурсів та запобігання тупиків. Знаходження тупика за допомогою редукції графа повторно використовуваних ресурсів. Мережа Петрі. 5.3 Мультипрограмування на основі переривань. Типи переривань. Диспетчеризація і пріоритезація переривань в ОС. Процедури обробки переривань і поточний процес. Системні виклики.	[9, 10, 12, 15]

Продовження табл. 4

1	2	3
	Лекція 6. Паралельна робота	
6	6.1 Організація паралельної роботи пристроїв і процесора. Супервізор введення-виведення. Узгодження швидкостей обміну і кешування даних. Розподіл пристроїв і даних між процесами. Підтримка синхронних і асинхронних операцій введення–виведення. 6.2 Багатопроцесорні системи. Мультипроцесорна обробка. Симетрична та асиметрична архітектура. Концепції UMA та NUMA мультипроцесорів. Мультиядерні мікропроцесори. Типи мультипроцесорних операційних систем Планування роботи мультипроцесорних систем. 6.3 Мультикомп'ютери. Апаратне забезпечення мультикомп'ютерів. Планування мультикомп'ютерів та балансування навантаження. Алгоритми розподілу процесорів	[17 - 19]
	Лекція 7. Віртуалізація та хмари	
2.	7.1. Засади віртуалізації. Віртуальні машини. Вимоги до віртуалізації. Гіпервізори першого та другого типу. 7.2 Технології ефективної віртуалізації. Гіпервізор першого типу. Ціна віртуалізації. 7.3 Віртуалізація пам'яті та введення/виведення. Віртуалізація пам'яті. Апаратна підтримка вкладених таблиць сторінок. Віртуалізація введення-виведення. Блоки управління пам'яттю під час введення-виведення. Віртуальні пристрої. 7.4 Розподілені системи. Особливості розподілених систем. ПЗ проміжного шару. CORBA. 7.5 Хмари як послуга. Основні характеристики хмар. Хмарна операційна система. 7.6 Міграція віртуальних машин. Жива міграція. Встановлення контрольних точок	[8, 10, 16]

Продовження табл. 4

1	2	3
	Лекція 8. Безпека ОС	
8.	8.1 Загрози і зловмисники. Основні поняття в сфері кібербезпеки. 8.2 Засади безпеки ОС. Модель безпеки в ядрі ОС. Монітор безпеки. Надійні системи. 8.3 Керування доступом до ресурсів. Принцип мінімальних повноважень. 8.4 Автентифікація. Ідентифікація й автентифікація. Технології. Біометрія. 8.5 Методи злому. Вразливості в операційній системі. Переповнення буфера. 8.6 Зловредні програми. Класифікація зловредного ПЗ. 8.7 Методи і засоби захисту. Сигнатура. Модель технології пошуку шкідливого коду. Антивірусні засоби захисту. Windows Defender. Міжмережні екрани. Пісочниці.	[20 – 23]

4. Методичні рекомендації до самостійної роботи

Навчальна дисципліна «Операційні системи» – це жива екосистема знань, заснована на самоорганізації як відповіді на зовнішні зміни. Враховуючи положення, що вища освіта як відповідь на суспільні виклики має постійно актуалізуватись в руслі сучасних трендів (challenge-based learning), ця дисципліна також повинна безперервно вдосконалюватись та оновлюватись [25, 26].

Напрямок операційних систем містить прояви різних сфер науки та досліджень, інтегрованих через технологічні потреби. Тому при вивченні предмету до нього має використовуватись міждисциплінарний підхід як до універсального професійного курсу. Необхідно також заначити, що в переході від інноваційної науково-технічної системи навчання STEM (Science, Technology, Engineering, Mathematics) через STEAM (плюс Arts – мистецтво) до STREAM (плюс Reading/wRiting – читання/письмо) саме в самостійній роботі студента проявляються, у доповненні до інжинірингу лекційних занять, творче ставлення до навчання як мистецтва та навички критичного мислення, втілені в читанні і письмі (рис. 5).

Рис. 5. Роль самостійної роботи студента в «потоці» STREAM

В цьому сенсі неабиякого значення набуває хід самостійної роботи студентів, опрацювання матеріалів лекцій та джерел інформації в процесі підготовки до контрольних заходів за результатами вивчення лекційних модулів. Відповідні рекомендації наведено у табл. 5.

Таблиця 5

Рекомендації щодо самостійної роботи студента з навчальної дисципліни

№ лекції	Теми	Завдання, що виноситься на самостійне опрацювання	Рекомендації	Додат-кові джерела
1	2	3	4	5
1	1	Оглядовий реферат «Основні етапи та тенденції розвитку операційних систем»	Розглянути процес розвитку операційних систем в контексті вдосконалення апаратної бази комп'ютерів, збільшення їх продуктивності, зростання обсягів пам'яті, кількості процесорів, поширення мережевих і хмарних технологій	[2, 3, 4, 6, 7, 10]
2	2	Алгоритм Деккера, Алгоритм Петерсона, Алгоритм розв'язання задачі про «філософів, що обідають».	Програмний підхід для паралельних процесів, які виконуються як в однопроцесорній, так і в багатопроцесорній системі із загальною основною пам'яттю. Такий підхід припускає елементарні взаємовиключення на рівні доступу до пам'яті без застосування апаратних засобів	[2, с. 239 - 240]

Продовження табл. 5

1	2	3	4	5
3	3	Проаналізувати Адаптивний алгоритм заміщення кешу (Adaptive Replacement Cache - ARC)	Опанування особливостей алгоритму для вирішення задачі заміщення сторінок	[12]
4	4	Особливості файлових систем в UNIX та Linux	Прослідити відмінності між файловими системами в UNIX та Linux від Windows. Розглянути кілька типів сучасних файлових систем для Linux, кожна з яких має свої особливості..	[13, 14]
5	5	Проаналізувати Алгоритм банкіра та Алгоритм взаємного виключення Лємпорта	Опанування особливостей алгоритмів для вирішення задач уникнення взаємних блокувань при розподілі ресурсів у системі на основі конкуренції, а також взаємного виключення в розподіленій системі	[2, с. 291 - 298]
6	3.2	Вивчити матеріал щодо підходів до паралельного програмування на основі MPI	Ознайомлення з організацією інформаційної взаємодії між процесорами, множиною операцій та різними способами передачі даних, особливістю розробки паралельних програм з використанням MPI	[19, с. 94-110]

Продовження табл. 5

1	2	3	4	5
7	3.3	Дослідження в галузі віртуалізації та хмар	Провести дослідження літератури та Інтернет-джерел та написати реферат, орієнтуючись на питанння переваг та недоліків використання хмарних обчислень для виконання програм користувача, зацікавленість хмарного дата-центру у віртуалізації. Навести приклади IAAS, PAAS та SAAS.	[24]
8	3.4	Вивчення додаткового матеріалу «Захист в операційних системах»	Розглянути презентаційний матеріал з розділів «Загрози безпеки ОС», «Побудова захищених ОС», «Адміністративні заходи захисту ОС», «Компоненти КЗЗ Windows», «Захист машинних носіїв».	[21]

5. Засоби діагностики, критерії та процедури оцінювання лекційних модулів

Лекційні заняття оцінюються шляхом виміру якості виконання конкретизованих завдань.

Конкретизовані завдання формуються на основі узагальнених шляхом конкретизації вихідних даних та способу демонстрації результатів навчання. Узагальнені засоби діагностики розробляються на базі програмних результатів навчання за стандартами вищої освіти та подаються в робочій програмі.

Поточний контроль результатів навчання зручно здійснювати за допомогою тестів, а також бального оцінювання практичних робіт (рис. 6). За результатами поточних контролів з усіх видів навчальних занять максимальне оцінювання досягає 90 балів.

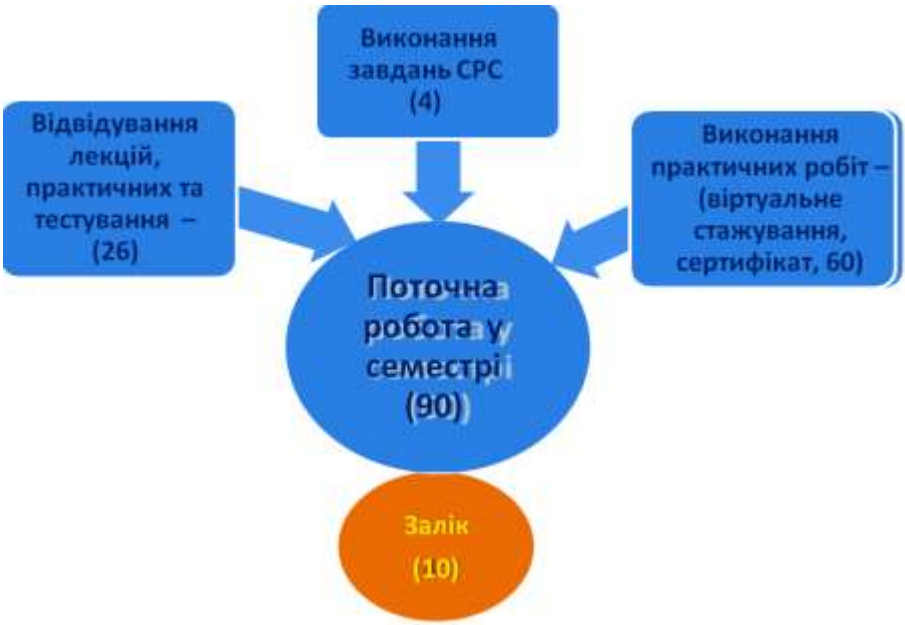

Рис. 6. Оцінювання поточної роботи студента у семестрі

Тестування засвоєння лекційного матеріалу відбувається в автоматизованому режимі на платформі Distance Learning - International European University (рис. 7). Кожен тест містить питання, пов'язані з ключовими дисциплінарними результатами навчання за даною лекцією. Кожен тест максимально оцінюється у 2 бали.

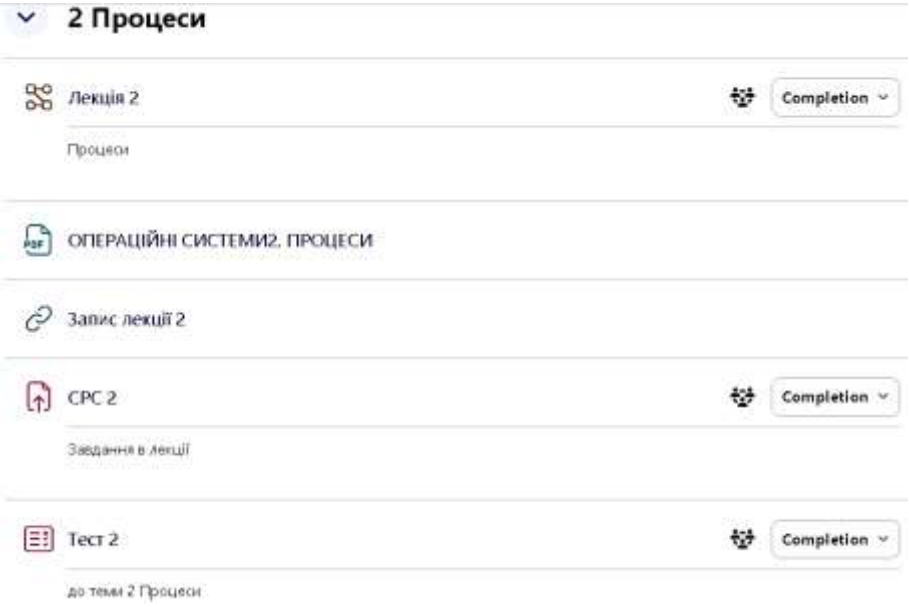

Рис. 7. Навчальні матеріали з дисципліни на платформі Distance Learning - International European University

Підсумковий контроль знань студентів проводиться у формі заліку.

Оцінювання відповіді під час заліку здійснюється в межах 0 – 10 балів та відбувається з використанням відповідних Критеріїв оцінювання (табл. 6). Залікова оцінка додається до оцінки поточної

роботи, що представляє загальну підсумкову оцінку в балах за національною шкалою та за шкалою ECTS.

Таблиця 6

Оцінювання відповіді студента під час заліку на окреме питання з навчальної дисципліни

№ з/п	Оцінка	Кіль-кість балів	Критерії оцінювання відповіді
1	2	3	4
1	«відмінно»	8 - 10	розгорнутий, вичерпний виклад змісту вирішення заданої у питанні проблеми; повний перелік необхідних для розкриття змісту питання термінів та понять; порівняльний аналіз різних теорій, концепцій, підходів та самостійно зроблені логічні висновки й узагальнення; демонстрація здатності висловлення та аргументування власного ставлення до альтернативних поглядів на дане питання
2	«добре»	4 - 7	відносно відповіді на найвищий бал не зроблено розкриття хоча б одного з пунктів, вказаних вище (якщо він явно потрібний для вичерпного розкриття питання) або, якщо: при розкритті змісту питання в цілому правильно за зазначеними вимогами зроблені окремі помилки
3	«задовільно»	1-3	відносно відповіді на найвищий бал не зроблено розкриття пунктів, зазначених у вимогах до нього; висновки, зроблені під час відповіді, не відповідають правильним чи загально визначеним при відсутності доказів супротивного аргументами, зазначеними у відповіді; характер відповіді дає підставу стверджувати, що студент не зовсім правильно розуміє зміст питання чи не знає правильної відповіді і тому не відповів на нього по суті, допустивши грубі помилки у змісті відповіді

6. Глосарій

Абсолютна адреса (absolute address, physical address) − адреса у вигляді цифрового коду, що однозначно ідентифікує реально існуючу в обчислювальній системі байт, комірку пам'яті або пристрій..

Агрегація − об'єкти одного класу, входять в об'єкти іншого.

Адаптер − пристрій зв'язку комп'ютера з периферійними пристро-ями.

Адреса (address) – номер байта в оперативній або іншій пам'яті комп'ютера.

Адресний простір (Address space) – діапазон адрес, доступні для комп'ютерної програми.

Адресний регістр (address register) – регістр процесора, що містить повну адресу ОПЗ або частину адреси, використовувану в разі визначення виконавчої адреси даних чи наступної виконуваної команди

Активна програма (active program) − програма, якій передано управління центральним процесором.

Артефакт (Artifacts) – будь-який продукт діяльності фахівців при розробці ПЗ.

Архітектура комп'ютера − логічна організація, структура й ресурси комп'ютера, які може використовувати програміст. Визначає принципи дії, інформаційні зв'язки й взаємне з'єднання основних логічних вузлів комп'ютера.

Архітектура системи (Systems arhitecture) – визначення системи в термінах обчислюваних складових (підсистем) і інтерфейсів між ними, яке відображає правила декомпозиції проблеми на складові.

Атрибут (Attribute) − ознака або властивість, що характеризують об'єкт.

Багатопоточність (Multithreading) – виконання декількох потоків в одному процесі

Базова система введення/виведення (БСВВ, BIOS) – набір програм, які працюють безпосередньо з апаратними засобами для того, щоб забезпечувати передачу інформації. Використовується для завантаження ОС.

Безвідмовність (Faultless) – атрибут надійності, який визначає відсутність або частоту відмов через помилки в програмному забезпеченні.

Блок – ділянка дискової пам'яті фіксованого розміру (найчастіше, 512 байтів), що є одиницею фізичного обміну даних.

Блокування (disable) – метод контролю функцій системи шляхом заборони певних видів діяльності.

Буфер – спеціальна ділянка пам'яті програми, яка надається кожній файловій змінній при її зв'язуванні.

Варіабельність (Variability) – здатність продукту (системи) до розширення, зміни, пристосування або конфігурації з метою використання в певному контексті і забезпечення його еволюції.

Введення даних (data input) – процес запису даних у пам'ять комп'ютера за допомогою пристрою введення.

Взаємне блокування (Mutual blocking) – блокування групи процесів, що очікують події, яка може ніколи не наступити, оскільки дану подію може викликати тільки інший процес з тієї ж групи. Взаємоблокування зазвичай виникають при доступі процесів до ресурсів.

Виведення даних (data output, output) − процес передачі даних з основної пам'яті комп'ютера на екран дисплея, папір або інший подібний носій, а також на зовнішні запам'ятовувальні пристрої.

Відновлюваність (Restored) – атрибут надійності, який вказує на можливість до перезапуску програми для повторного виконання і відновлення даних після відмов.

Віртуальна адреса (virtual address) − 1. Адреса у віртуальній машині. 2. Те жсаме, що логічна адреса

Віртуальна пам'ять (virtual storage, virtual memory) − 1. Механізм управління пам'яттю обчислювальної системи, що дозволяє програмі використовувати пам'ять, розмір якої більший розміру реальної оперативної пам'яті, наявної в комп'ютера.

Виконання програми (program run, computer run, run) − робота комп'ютера по заданій програмі.

Вхід у програму (program entry) − оператор або команда, яким передається керування при запуску програми або виклику підпрограми.

Дефект (Fault) – результат помилок розробника у вхідних або проектних специфікаціях, текстах кодів програм, експлутаційной документації і т.п

Дефрагментація − це оптимізація дискового простору, упорядкування кластерів, які належать одному файлу.

Динамічна пам'ять (dynamic storage) − спосіб розподілу оперативної пам'яті між завданнями.

Драйвери (driver) − програми, що розширюють можливості операційної системи по управлінню пристроями введення-виведення.

Завантаження – запис програми в оперативну пам'ять. Виконується за. спеціальною програмою – завантажником, що входить до складу ОС.

Завершення програми (completion, termination) − остання фаза обробки програми операційною системою, у ході якої звільняються запитані програмою ресурси, виводяться результати, очищається виділена програмі оперативна пам'ять.

Зовнішня пам'ять (backing storage, external memory) − пам'ять, до вмісту якої можна звернутися тільки за допомогою операцій введення-виведення.

Ідентифікатор (identifier) − рядок символів, призначений для позначення об'єкта програми або обчислювальної системи.

Ідентифікація об'єкта − це властивість, що дозволяє відрізнити об'єкт від інших об'єктів того ж або інших класів..

Канал (channel) – маршрут або лінія зв'язку, яким здійснюється обмін інформацією між двома пристроями.

Каталог, довідник (directory) – збірка назв файлів та інших довідників, що зберігається на диску; спосіб організації та групування файлів на диску, за якого користувач не перевантажується довгим переліком файлів.

Кеш (cache) – спеціальна підсистема пам'яті, в якій часто використовувані дані копіюються для швидкого доступу.

Кластер (cluster) – одиниця зберігання на диску, що містить фіксовану кількість секторів диска і використовується операційною системою під час зчитування або записування інформації; звичайно кластер містить від двох до 8 секторів, кожний з яких вміщує певну кількість байтів (символів).

Комунікативний процес (Communicative process) – забезпечення отримання, сприйняття і розуміння інформації, що є предметом обміну

Конфігурація (Configuration) – структура програмної системи з компонентів і набору варіантів модулів.

Модуль (Module) – незалежна функціональна частина програми, до якої можна звертатися як до самостійної одиниці через зовнішній інтерфейс.

Монітор (Monitor) – набір процедур, змінних й інших структур даних, об'єднаних в особливий модуль, що забезпечує функціональність, еквівалентну до функціональності семафорів, але при цьому ним набагато легше управляти.

Мультипрограмування (Multiprogramming) – це такий спосіб організації обчислень, коли на однопроцесорній обчислювальній системі імітується одночасне виконання декількох програм.

М'ютекс (Mutex) – це спрощена версія семафора: змінна, яка може знаходитися в одному з двох станів – блокованому і неблокованому.

Надійність (Reliability) – набір атрибутів, які вказують на можливість програми перетворювати вхідні дані в вихідні результати за умов, які залежать від періоду часу (знос і старіння не враховуються) і наявності відмов в результаті внутрішніх дефектів або порушення умов його застосування

Об'єктно орієнтована модель (Object-oriented model) – уявлення програм системи як сукупності об'єктів, які взаємодіють між собою і мають певні властивості і поведінку.

Образ процесу (Process image) – містить програму (код), дані, стек і атрибути процесу. Структура даних разом з атрибутами процесу часто називають керуючим блоком процесу.

Паралельні процеси (Concurrent processes) – процеси, виконання яких може перекриватися в часі, наприклад, деякий процес починається раніше, ніж завершується інший.

Переносимість (Bearable) – група властивостей, яка забезпечує пристосованість до переносу з одного середовища функціонування в інші, а також зусилля для перенесення програм в нове середовище.

Переривання (Interrupt) – призупинка нормального виконання процесу, викликана зовнішньою по відношенню до процесу подією і здійснюється таким чином, що по її завершенні виконання процесу може бути продовжено.

Помилка (Mistake) – недоліки в операторах програми або в технологічному процесі розробки, що призводять до неправильної інтерпретації вхідної інформації, а отже, і до неправильного вирішення.

Потік (Thread) – незалежно запланований контекст виконання, що розділяє єдиний адресний простір з іншими потоками свого процесу.

Процес (Process) – окрема програма в момент її виконання разом з виділеними їй ресурсами комп'ютера.

Реєнтерабельна процедура (Re-enterable procedure) – підпрограма, вхід в яку може бути здійсненим до завершення її попереднього виконання.

Резерв, резервувати (backup) – копія програми, диска або даних, що створюється з метою архівації або захисту цінних файлів від пошкоджень.

Ресурс (Resource) – будь-який об'єкт, який може запитуватися, очікуватися та використовуватись процесом. Ресурс може складатися з будь-якої кількості ідентичних одиниць.

Розподіл часу (Time sharing) – механізм паралельного використання пристрою декількома користувачами або процесами.

Розподілена операційна система (Distributed operating system) – спільна операційна система, що сумісно використовується мережею комп'ютерів.

Сервісно орієнтоване програмування (Service oriented programming) - сукупність взаємодіючих сервісів, веб-сервісу і їх інтерфейсів.

Сертифікація продукту (Certification of product) - процес, який підтверджує відповідність ідентифікованої програмної продукції (процесу або послуг) конкретному стандарту або технічним умовам з оцінкою спеціальним знаком або свідоцтвом.

Специфікація (Specification) – уявлення правил, стандартів, критеріїв якості, обмежень користування об'єктами.

Спулінг (Spooling) – використання вторинної пам'яті в якості буферу для зниження затримок при передаванні даних між процесором і периферійними пристроями комп'ютера.

Супровід (Software support) – будь-які роботи по внесенню змін до ПС після того, як вона була передана користувачеві для експлуатації.

Тест (Test) – деяка програма, призначена для перевірки працездатності іншої програми і виявлення в ній помилкових ситуацій.

Тестування (Testing) – засіб семантичної налагодження (перевірки) програми, яка полягає в опрацюванні програмою послідовності різних контрольних наборів тестів, для яких відомий результат.

Ущільнення (compaction) – процес збирання і пакування виділених ділянок оперативної пам'яті у найменш можливу сферу для того, щоб створити якнайбільшу ділянку пам'яті.

Фізична адреса (Physical address) – реальна адреса області пам'яті, з якою працює пристрій управління пам'яттю (Memory Management Unit – MMU).

Функціональність (Functionality) – сукупність властивостей, які визначають можливість ПО виконувати перелік функцій в заданому середовищі і відповідають вимогам.

Ядро (Kernel) операційної системи – набір функцій, структур даних та окремих програмних модулів, які завантажуються в пам'ять комп'ютера при завантаженні операційної системи.

Література

1. Салов В.О., Письменкова Т.О. Рекомендації до створення комплексу навчально-методичного забезпечення дисциплін: метод. посіб. для наук.-пед. прац. Нац. гірн. ун-т, наук. метод. центр. Д. : НГУ, 2017. 48 с.
2. Авраменко В. С., Авраменко А. С. Основи операційних систем. Навчальний посібник. Черкаси: ЧНУ імені Богдана Хмельницького, 2018. 524 с.
3. Габрусєв В. Ю. Лапінський В.В., Нестеренко О.В. Основи операційних систем: Ядро, процес, потік. Тернопіль: Богдан, 2007. 94 с.
4. Матвієнко М.П., Розен В.П., Закладний О.М. Архітектура комп'ютера. К.: Видавництво Ліра-К, 2013. 264 с.
5. Зайцев В. Г., Дробязко І. П. Операційні системи: навч. посіб. КПІ ім. Ігоря Сікорського. 2019. 240 с. [Електронний ресурс]
6. Операційні системи: навчальний посібник. [за ред. В. М. Рудницького] / І. М. Федотова-Півень, І. В. Миронець, О. Б. Півень, С. В. Сисоєнко, Т. В. Миронюк; Черкаський державний технологічний університет. – Харків : ТОВ «ДІСА ПЛЮС», 2019. 216 с.
7. Abraham Silberschatz, Peter B. Galvin, Greg Gagne. Operating System Concepts
8. Andrew S. Tanenbaum, Herbert Bos. Modern Operating Systems
9. William Stallings Operating Systems: Internals and Design Principles
10. Gary Nutt. Operating Systems: A Modern Perspective
11. Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau. Operating Systems: Three Easy Pieces" (OSTEP). Безкоштовна онлайн-книга: https://pages.cs.wisc.edu/~remzi/OSTEP/
12. Thomas Anderson and Michael Dahlin. Operating Systems: Principles and Practice.
13. Robert Love. Linux Kernel Development.

14. Haseman C. Android Essentials. Apress, 2008. 116 p.

15. Allen B. Downey. The Little Book of Semaphores. PDF: https://greenteapress.com/semaphores/LittleBookOfSemaphores.pdf

16. Max Hailperin. Operating Systems and Middleware: Supporting Controlled Interaction. https://gustavus.edu/mcs/max/os-book/

17. Семеренко В. П. Технології паралельних обчислень : навч. посібник. Вінниця : ВНТУ, 2018. 104 с.

18. Коцовський В. М. Теорія паралельних обчислень: навч. посібник. Ужгород: ПП «АУТДОР-Шарк», 2021. 188 с.

19. Минайленко Р.М. Паралельні та розподілені обчислення: навч. посіб. Кропивницький: Видавець Лисенко В. Ф., 2021. 153 с.

20. Вінницький В. В. Захист інформації у багатопроцесорних системах**.** Донецький національний університет імені Василя Стуса, Вінниця, 2021. 66 с.

21. Гайворонський М.В., Новіков О.М. Безпека інформаційно-комунікаційних систем: підручн. К.: Видавнича група BHV, 2009. 608с.

22. Шевченко В.Л., Нестеренко О.В., Нетесін І.Є., Шевченко А.В. Прогностичне моделювання комп'ютерних вірусних епідемій. Київ: УкрНЦ РІТ, 2019. 152 с.

23. Ковтунець В.В., Нестеренко О.В., Савенков О.І. Безпека систем підтримки прийняття рішень / Навч. посіб. К: Національна академія управління, 2016. 190 с.

24. В університетах є Creative Commons-ліцензовані конспекти та книги, наприклад:

Carnegie Mellon University's 15-410 OS Course

https://www.cs.cmu.edu/~410/

Georgia Tech OMSCS Advanced OS (CS 6210)

https://gt-cs6210.github.io/syllabus/

MIT OpenCourseWare — Operating System Engineering (6.828)

https://ocw.mit.edu/courses/6-828-operating-system-engineering-fall-2012/

25. Калашнікова С. Імплементація парадигми вдосконалення викладання в університетах України: інституційний та індивідуальний виміри. ITeachers Meet-Up #6. Інститут вищої освіти НАПН України, 2022. 27 с.

26. Нестеренко О. Інфологічне моделювання простору освіти з інженерії програмного забезпечення. Проблеми освіти, 2022, 2(97), 147-168.

Навчально-методичне видання

Нестеренко Олександр Васильович

ОПЕРАЦІЙНІ СИСТЕМИ

Методичні рекомендації
до опанування лекційних занять
з навчальної дисципліни
для здобувачів вищої освіти спеціальності
F2 «Інженерія програмного забезпечення»

Міжнародний європейський університет

Навчально-науковий інститут
«Європейська школа бізнесу»

Кафедра інформаційних технологій

Відповідальний за випуск
Ліщенко А. В.