

Міністерство освіти і науки України  
Харківський національний університет імені В.Н. Каразіна  
Кафедра штучного інтелекту та програмного забезпечення



**“ЗАТВЕРДЖУЮ”**

Проректор  
з науково-педагогічної роботи

Антон ПАНТЕЛЕЙМОНОВ

2020 р.

## РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### Операційні системи

рівень вищої освіти	перший (бакалаврський) рівень
галузь знань	15 Автоматизація та приладобудування
спеціальність	151 Автоматизація та комп'ютерно-інтегровані технології
освітня програма	Автоматизація та комп'ютерно-інтегровані технології
вид дисципліни	обов'язкова
факультет	комп'ютерних наук

2020 / 2021 навчальний рік

Програму рекомендовано до затвердження Вченою радою факультету комп'ютерних наук  
«31» серпня 2020 року, протокол № 12

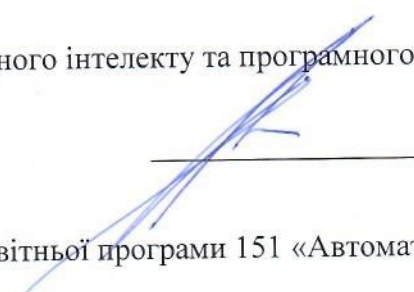
РОЗРОБНИК ПРОГРАМИ:

Старший викладач кафедри штучного інтелекту та програмного забезпечення **Горбань  
Анатолій Михайлович**

Доцент кафедри штучного інтелекту та програмного забезпечення **Споров Олександр  
Євгенійович**

Програму схвалено на засіданні кафедри штучного інтелекту та програмного забезпечення  
Протокол від «31» серпня 2020 року № 1

Завідувач кафедри штучного інтелекту та програмного забезпечення

  
Володимир КУКЛІН

Програму погоджено з гарантом освітньої програми 151 «Автоматизація та комп'ютерно-інтегровані технології»

Гарант освітньої програми 151 «Автоматизація та комп'ютерно-інтегровані технології»

  
Дмитро ЛАБЕНКО

Програму погоджено методичною комісією факультету комп'ютерних наук  
Протокол від «31» серпня 2020 року № 1

Голова методичної комісії факультету комп'ютерних наук

  
Анатолій БЕРДНІКОВ

## ВСТУП

Програма навчальної дисципліни “Операційні системи” складена відповідно до освітньо-професійної програми підготовки (бакалаврського) рівня вищої освіти, напрямів підготовки: 151 «Автоматизація та комп’ютерно-інтегровані технології».

### 1. Опис навчальної дисципліни

1.1. Основною метою є засвоєння студентами як фундаментальних принципів організації та функціонування ОС, так і прийомів практики системного програмування для широко розповсюджених ОС.

1.2. Формування у студентів знань про: організацію, структуру та функціонування ОС; програмні та програмно-апаратні засоби, а також, стратегії управління процесами та ресурсами.

Студенти мають оволодіти уміннями: оцінки архітектури ОС та системного програмування з використанням програмного інтерфейсу ядра (API) розповсюджених операційних систем сімейств Windows, UNIX, Linux.

Вивчення курсу "Операційні системи" базується на знаннях, отриманих при вивченні таких курсів як "Програмування та алгоритмічні мови", "Об'єктно-орієнтоване програмування", "Архітектура комп'ютерів" тощо.

В ході вивчення дисципліни у студента повинні формуватися наступні компетентності.

*Інтегральна компетентність.*

Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми, що характеризуються комплексністю та невизначеністю умов, під час професійної діяльності у галузі автоматизації або у процесі навчання, що передбачає застосування теорій та методів галузі.

*Загальні компетентності (КЗ).*

- здатність до пошуку, опрацювання та аналізу інформації з різних джерел. К05.

*Спеціальні (фахові, предметні) компетентності (ФК)*

- Володіння методами та засобами програмування мовами високого та низького рівня, КЗП.07
- Знання принципів роботи операційних систем, КЗП.08

1.3. Кількість кредитів – 4

1.4. Загальна кількість годин – 120

1.5. Характеристика навчальної дисципліни	
Нормативна / за вибором	
денна форма навчання	Заочна (дистанційна) форма навчання
Рік підготовки	
3-й	
Семестр	
5-й	
Лекції	
32 год.	_____ год.
Практичні, семінарські заняття	
_____ год.	_____ год.
Лабораторні заняття	
32 год.	_____ год.
Самостійна робота	
56 год.	_____ год.
Індивідуальні завдання	
_____ год.	_____ год..

#### 1.6. Заплановані результати навчання

##### **знати:**

- базові знання призначення, будови, функціонування та властивостей операційних систем;
- засоби розробки та супроводу операційних систем,
- вимоги та стандарти сучасних операційних систем загального та спеціального призначення.

##### **вміти:**

- оцінити операційну систему та системне програмне забезпечення,
- працювати з інструментарієм системного програміста,
- розробляти та реалізовувати нескладне системне програмне забезпечення.

В результаті вивчення дисципліни у студента повинні формуватися наступні програмні результати навчання (ПРН).

- розуміти суть процесів, що відбуваються в об'єктах автоматизації (за галузями діяльності) та вміти проводити аналіз об'єктів автоматизації і обґрунтовувати вибір структури, алгоритмів та схем керування ними на основі результатів дослідження їх властивостей. ПР04.

## 2. Тематичний план навчальної дисципліни

### *Розділ 1. Процеси і потоки.*

#### *Тема 1. Основні поняття. Комп'ютерні системи та ОС.*

Визначення ролі та функцій ОС та системного програмного забезпечення. Знайомство з апаратними засобами обчислювальних систем, поняття архітектури.

#### *Тема 2. Процеси. Життєвий цикл процесу та управління ним.*

Поняття процесу, стани процесу, типи багатозадачності, управління з боку програми та ОС.

*Тема 3. Потоки та управління ними.*

Поняття потоку, типи багатопоточності, управління потоками, архітектури багатопроцесорних систем, організація та робота симетричних мультипроцесорних систем.

*Тема 4. Типи архітектури ядра.*

Архітектура монолітних, модульних, та мікроядер. Особливості архітектури ядер ОС Windows, UNIX, Linux.

*Розділ 2. Паралельна обробка інформації.*

*Тема 1. Принципи паралельних обчислень, взаємовиключення.*

Особливості паралельних обчислень. Необхідність взаємовиключень, вимоги до них та пов'язані з ними проблеми.

*Тема 2. Засоби синхронізації процесів: семафори, монітори і повідомлення.*

Реалізація взаємовиключень – програмний підхід, апаратна підтримка, спін-блокування, семафори та м'ютекси. Синхронізація за допомогою повідомлень.

*Тема 3. Задача умовної синхронізації процесів.*

Безумовна та умовна синхронізація – канонічні задачі умовної синхронізації – задача читачів – письменників та виробника - споживача.

*Тема 4. Взаємне блокування і голодування.*

Стратегії запобігання та відвернення взаємного блокування голодування процесів. Канонічна задача «обідаючих філософів».

*Тема 5. Засоби синхронізації ОС сімейства UNIX та Windows.*

Особливості реалізації засобів синхронізації в ОС сімейств UNIX, Windows та в мовах програмування.

*Розділ 3. Пам'ять і планування процесора*

*Тема 1. Управління пам'яттю та її організація.*

Вимоги до управління пам'яттю, типи організації – розділи, сторінкова та сегментна.

*Тема 2. Віртуальна пам'ять*

Програмна реалізація та апаратна підтримка віртуальної пам'яті. Стратегії управління віртуальною пам'яттю.

*Тема 3. Управління пам'яттю в ОС сімейства UNIX та Windows.*

Особливості реалізації стратегій управління пам'яттю ядра та адресного простору користувача в ОС сімейств UNIX та Windows.

*Тема 4. Планування роботи в системах з одним процесором.*

Огляд стратегій середньо та довгострокового планування процесора.

*Розділ 4. Багатопроцесорні і розподілені системи. Ввід-вивід і файли.*

*Тема 1. Багатопроцесорне планування і планування реального часу.*

Особливості багатопроцесорного планування та планування систем жорсткого режиму реального часу. Мережеві операційні системи.

*Тема 2. Розподілені операційні системи, кластери і управління розподіленими процесами.*

Організація управління розподіленими обчисленнями в кластерних системах та перспективи розподілених ОС.

*Тема 3. Управління вводом-виводом і дискове планування. Управління файлами.*

Структура систем вводу-виводу, основні стратегії планування та управління, системи масового зберігання інформації. Поняття файлової організації інформації та управління файловими системами.

## 3. Структура навчальної дисципліни

Назви розділів і тем	Кількість годин					
	денна форма					
	усього	у тому числі				
л		п	лаб.	інд.	с. р.	
1	2	3	4	5	6	7
<b>Розділ 1. Процеси і потоки</b>						
Тема 1. Основні поняття. Комп'ютерні системи та ОС	10	2		6		2
Тема 2. Процеси. Життєвий цикл процесу та управління ним	8	2		4		2
Тема 3. Потоки та управління ними	8	2		4		2
Тема 4. Типи архітектури ядра	10	2				8
<b>Разом за розділом 1</b>	<b>36</b>	<b>8</b>		<b>14</b>		<b>14</b>
<b>Розділ 2. Паралельна обробка інформації</b>						
Тема 1. Принципи паралельних обчислень, взаємовиключення	6	2				4
Тема 2. Засоби синхронізації процесів: семафори, монітори і повідомлення	8	2		4		2
Тема 3. Задача умовної синхронізації процесів	6	2				4
Тема 4. Взаємне блокування і голодування	6	2				4
Тема 5. Засоби синхронізації ОС сімейства UNIX та Windows	6	2		2		2
<b>Разом за розділом 2</b>	<b>32</b>	<b>10</b>		<b>6</b>		<b>16</b>
<b>Розділ 3. Пам'ять і планування процесора</b>						
Тема 1. Управління пам'яттю та її організація	8	2				6
Тема 2. Віртуальна пам'ять	6	2				4
Тема 3. Управління пам'яттю в ОС сімейства UNIX та Windows	8	2		4		2
Тема 4. Планування роботи в системах з одним процесором	6	2				4
<b>Разом за розділом 3</b>	<b>28</b>	<b>8</b>		<b>4</b>		<b>16</b>
<b>Розділ 4. Багатопроесорні і розподілені системи. Ввід-вивід і файли</b>						
Тема 1. Багатопроесорне планування і планування реального часу	6	2				4
Тема 2. Розподілені операційні системи, кластери і управління розподіленими процесами	6	2				4
Тема 3. Управління вводом-виводом і дискове планування. Управління файлами	12	2		8		2
<b>Разом за розділом 4</b>	<b>24</b>	<b>6</b>		<b>8</b>		<b>10</b>
<b>Усього годин</b>	<b>120</b>	<b>32</b>		<b>32</b>		<b>56</b>

#### 4. Теми практичних (лабораторних) занять

№ з/п	Назва теми	Кількість годин
<b>Розділ 1. Процеси і потоки</b>		
1	Основи роботи в Linux. Налаштування оточення.	2
2	Основи використання мови C – компілятор gcc та інструментарій.	2
3	Робота з оточенням середовища. Утиліта make.	2
4	Процеси – моніторинг, функція system(), завершення процесів.	2
5	Створення процесу – ф-ція fork(), група ф-цій exec(), wait().	2
6	Потоки Linux. Створення та завершення потоку. Обмін даними.	2
7	Управління потоками, атрибути потоків, відміна, потокові дані.	2
<b>Розділ 2. Паралельна обробка інформації</b>		
8	Синхронізація потоків – м'ютекси, блокування читання та запису.	2
9	Синхронізація потоків – семафори, умовні змінні.	2
10	Робота з сигналами Linux.	2
<b>Розділ 3. Пам'ять і планування процесора</b>		
11	Міжпроцесна взаємодія з використанням спільної пам'яті та файлів.	4
<b>Розділ 4. Багатопроцесорні і розподілені системи. Ввід-вивід і файли</b>		
12	Використання механізму каналів для міжпроцесної взаємодії.	2
13	Низькорівневе введення-виведення в Linux.	2
14	Основи роботи з файловими системами Linux.	2
15	Безпека, права доступу, тимчасові файли.	2
<b>Разом:</b>		<b>32</b>

#### 5. Завдання для самостійної роботи

№ з/п	Види, зміст самостійної роботи	Кількість годин
<b>Розділ 1. Процеси і потоки</b>		
1	Знайомство з основними положеннями дисципліни та основами організації і функціонування багатозадачних ОС - за джерелами з списку рекомендованої літератури	12
<b>Розділ 2. Паралельна обробка інформації</b>		
2	Відпрацювання методики реалізації умовної синхронізації в багатопоточних задачах та задачах з кількома потоками. Мова програмування – довільна.	14
<b>Розділ 3. Пам'ять і планування процесора</b>		
3	Поглиблене вивчення теми за книгою [5], розділ 4.	15
<b>Розділ 4. Багатопроцесорні і розподілені системи. Ввід-вивід і файли</b>		
4	Поглиблене вивчення теми за книгою [5] розділ 3, розділ 5.	10
5	Підготовка до контрольної роботи	5
<b>Разом</b>		<b>56</b>

#### 6. Індивідуальні завдання

1 контрольна робота

#### 7. Методи навчання

На досягнення освітніх цілей спрямовані такі методи навчання студентів:

– *практичні* (використовують для пізнання дійсності, формування навичок і вмій, поглиблення знань. Під час їх застосування використовуються такі прийоми:

планування виконання завдання, постановка завдання, оперативне стимулювання, контроль і регулювання, аналіз результатів, визначення причин недоліків);

– *пояснювальне-ілюстративний* (використовують для викладання й засвоєння нового навчального матеріалу, фактів, підходів, оцінок, висновків тощо);

– *репродуктивний* (для застосування студентами вивченого на основі зразка або правила, алгоритму, що відповідає інструкціям, правилам, в аналогічних до представленого зразка ситуаціях);

Як правило лекційні та практичні заняття проводяться аудиторне. В умовах дії карантину заняття проводяться відповідно до Наказу ректора Харківського національного університету імені В.Н. Каразіна (аудиторне або дистанційно за допомогою платформ Google Meet або Zoom).

## 8. Методи контролю

На протязі усього терміну викладання означеної дисципліни проводиться поточний контроль засвоєння лекційного матеріалу (контроль знань) та контроль здобуття практичних навиків (контроль вмінь). Підсумковий семестровий контроль також дозволяє контролювати як одержані знання так і набуті вміння:

- Після закінчення викладання кожного розділу курсу контроль знань проводиться у вигляді тестування по матеріалам розділа.
- Після викладання курсу здійснюється підсумковий контроль.

## 9. Схема нарахування балів

Підсумковий семестровий контроль в формі заліку без виконання залікової роботи

Поточний контроль та самостійна робота						
Розділ 1	Розділ 2	Розділ 3	Розділ 4	Контрольна робота, передбачена навчальним планом	Індивідуальне завдання	Разом
15	15	15	15	40	--	100

## КРИТЕРІЇ ОЦІНКИ УСПІШНОСТІ ТА РЕЗУЛЬТАТІВ НАВЧАННЯ З ПРАКТИЧНОЇ ЧАСТИНИ КУРСУ «ОПЕРАЦІЙНІ СИСТЕМИ»

*Форми навчальної діяльності – максимальна кількість балів, що можна отримати протягом семестру*

1. Відвідування лабораторних занять – 4 бали
2. Виконання лабораторних робіт – 46 балів

### Критерії оцінювання відвідування занять

*Відвідування лабораторних занять*

За семестр студент має бути присутнім на 16 лабораторних заняттях. Присутність на кожному занятті оцінюється в 0.25 балів. В кінці семестру результат за цією формою навчальної діяльності підраховується за формулою:

$$N * 0.25,$$

де  $N$  – кількість лабораторних занять, що були відвідані студентом.



### Критерії оцінювання виконання лабораторних робіт

Протягом семестру студент має виконати 12 лабораторних робіт за темами:

**Тема 1. Базові засоби мови програмування Cі: покажчики, масиви, структури** – максимальна кількість балів: 3 бали; термін виконання – 2 тижні.

**Тема 2. Взаємодія с середовищем виконання (execution environment)** – максимальна кількість балів: 3 бали; термін виконання – 2 тижні.

**Тема 3. Основи роботи з процесами в Linux** – максимальна кількість балів: 3 бали; термін виконання – 2 тижні.

**Тема 4. Програмне створення процесів в Linux** – максимальна кількість балів: 4 бали; термін виконання – 2 тижні.

**Тема 5. Основні операції по створенню потоків** – максимальна кількість балів: 4 бали; термін виконання – 2 тижні.

**Тема 6. Програмне налаштування та керування завершенням потоків** – максимальна кількість балів: 4 бали; термін виконання – 2 тижні.

**Тема 7. Синхронізація потоків: м'ютекси, блокування читання / запису** – максимальна кількість балів: 4 бали; термін виконання – 2 тижні.

**Тема 8. Синхронізація потоків: семафори для потоків, умовні змінні** – максимальна кількість балів: 4 бали; термін виконання – 2 тижні.

**Тема 9. Процеси. Сигнали** – максимальна кількість балів: 4 бали; термін виконання – 2 тижні.

**Тема 10. Засоби міжпроцесної взаємодії: загальні області пам'яті** – максимальна кількість балів: 5 балів; термін виконання – 2 тижні.

**Тема 11. Засоби міжпроцесної взаємодії: семафори для процесів** – максимальна кількість балів: 5 балів; термін виконання – 2 тижні.

**Тема 12. Засоби міжпроцесної взаємодії: неіменовані програмні канали pipe, іменовані канали FIFO** – максимальна кількість балів: 3 бали; термін виконання – 2 тижні.

Перед виконанням лабораторної роботи студент вивчає вимоги та технічне завдання. Після цього відбувається обговорення та перевірка того, що завдання було зрозуміле коректно. Кожен студент виконує роботу самостійно. Дозволяється сумісне обговорення роботи, але програмний код кожен студент пише власноруч. Якщо програмний код різних робіт має суттєву ступінь схожості, то ці роботи дискваліфікуються та не оцінюються.

Захист лабораторної роботи складається з двох етапів.

Перший етап – представлення результатів: готового програмного застосування та програмних кодів у відповідності до завдання лабораторної роботи.

Критерії оцінювання результатів роботи:

- робота була виконана у відповідності з технічним завданням за вказаний час – студент отримує 100% від максимальної кількості балів;
- робота була виконана у відповідності з технічним завданням с запізненням до одного тижня – студент отримує 80 % від максимальної кількості балів;
- робота була виконана у відповідності з технічним завданням с запізненням більше, ніж на тиждень – студент отримує 50 % від максимальної кількості балів;
- технічне завдання виконано не повністю, а на  $x$  % – студент отримує  $x$  % від балів, що мали б бути зараховані у відповідності до строків виконання.

Другий етап здачі лабораторної роботи – відповідь на контрольні питання. Кількість питань визначає викладач за результатами представлених результатів, але не менше ніж 3. Кожне контрольне питання оцінюється таким чином:

- повна розгорнута відповідь з прикладами та додатковим матеріалом, що був опрацьований на самостійній роботі – кількість балів, що отримані на першому

етапі збільшується на 5 % балів (але загальний результат не може перевищувати запланований максимальний бал за роботу) ;

- повна, але не розгорнута відповідь – кількість балів не змінюється.
- неповна відповідь, або відповідь, що містить незначні та некритичні помилки чи суперечності – кількість балів, що було отримано на попередньому етапі зменшується на 5 %;

відповідь, що містить критичні помилки, або відсутність відповіді – кількість балів, що було отримано на попередньому етапі зменшується на 10 %.

### Шкала оцінювання

Сума балів за всі види навчальної діяльності протягом семестру	Оцінка	
	для чотирирівневої шкали оцінювання	для дворівневої шкали оцінювання
90 – 100	відмінно	зараховано
70-89	добре	
50-69	задовільно	
1-49	незадовільно	не зараховано

## 10. Рекомендована література

### Основна література

1. Столлингс В. Операционные системы. 4-е изд. - М: «Вильямс», 2002 - 848 с.
2. Таненбаум Э. Современные операционные системы: 3-е издание - СПб.: «Питер», 2004.
3. Беляков М.И., Рабовер Ю.И., Фридман А.Л. Мобильная операционная система: - М.: «РиС», 1991.- 208 с.
4. Куклин В. М. ред. Введение в методы программных решений: учебное пособие: - Х.: ХНУ им. В.Н. Каразина, 2011

### Допоміжна література

1. Таненбаум Э., Вудхалл А. Операционные системы. Разработка и реализация. 3 изд. — СПб.: Питер, 2007..
2. Э. Таненбаум, М. ван Стеен. Распределенные системы. Принципы и парадигмы. СПб.: Питер, 2003.
3. Э. Таненбаум. Компьютерные сети. 4-е издание. СПб.: Питер, 2003.
4. Побегайло А. Системное программирование в Windows: - СПб.: «БХВ-Петербург», 2006.
5. Лав Р. Linux. Системное программирование: – СПб.: «Питер», 2008.

## 11. Посилання на інформаційні ресурси в Інтернеті, відео-лекції, інше методичне забезпечення

1. Подборка русскоязычной документации по UNIX-подобным операционным системам: [Електроний ресурс]// [АО «Флант»](http://www.nixp.ru/links/103.html), М.,2001—2017. URL: <http://www.nixp.ru/links/103.html>.
2. Стандарты ОС UNIX: [Електроний ресурс]// [Сервер Информационных Технологий](http://docstore.mik.ua/unix2/glava_39.htm), М. URL: [http://docstore.mik.ua/unix2/glava\\_39.htm](http://docstore.mik.ua/unix2/glava_39.htm).
3. Linux: [Електроний ресурс]// IBM developer Works, IBM Corporation, 1994-2017. URL: <http://ibm.com/developerworks/ru/linux>

4. Официальный сайт Windows: [Электронный ресурс]// Microsoft, 2017. URL: <http://windows.microsoft.com/ru-ru/windows/home>.
5. Операционные системы: [Электронный ресурс]// Internet-web.Ru, 2009. URL: <http://www.internet-web.ru/tema2.html>.
6. Официальный сайт QNX Software Systems: [Электронный ресурс]//. QNX Software Systems Limited, a subsidiary of BlackBerry, 2017. URL: <http://www.qnx.com/>
7. Единая спецификация «Юникс»: [Электронный ресурс]//. The IEEE and The Open Group, 2001-2016. URL: <http://pubs.opengroup.org/onlinepubs/9699919799/>