

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І.І. МЕЧНИКОВА  
Кафедра математичного забезпечення комп'ютерних систем



**ЗАТВЕРДЖУЮ**

Проректор з науково-педагогічної роботи

“ 30.08 2024 ” р.

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

ОК23 «Системне програмування»

(назва навчальної дисципліни)

Рівень вищої освіти Перший (бакалаврський)

Галузь знань: 12 – Інформаційні технології

Спеціальність 123 – Комп'ютерна інженерія  
код і назва спеціальності (тей)

Освітньо-професійна/наукова програма: Комп'ютерна інженерія

Робоча програма навчальної дисципліни  
«Системне програмування». – Одеса: ОНУ, 2024. – 19 с.

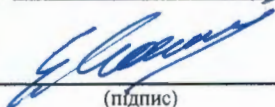
Розробники:

Трубіна Н.Ф., старший викладач кафедри МЗКС,  
Лісіцина І.М., старший викладач кафедри МЗКС

Робоча програма затверджена на засіданні кафедри математичного забезпечення  
комп'ютерних систем

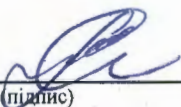
Протокол № 1 від. “ 28 ” 08 2024 р.

Завідувач кафедри

  
\_\_\_\_\_  
(підпис)

( Євгеній МАЛАХОВ )  
(Ім'я ПРІЗВИЩЕ)

Погоджено із гарантом ОПП «Комп'ютерна інженерія»

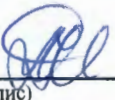
  
\_\_\_\_\_  
(підпис)

( Людмила ВОЛОЩУК )  
(Ім'я ПРІЗВИЩЕ)

Схвалено навчально-методичною комісією (НМК) з ІТ спеціальностей  
факультету МФІТ

Протокол № 1 від. “ 30 ” 08 2024 р.

Голова НМК

  
\_\_\_\_\_  
(підпис)

( Лариса МАРТИНОВИЧ )  
(Ім'я ПРІЗВИЩЕ)

Переглянуто та затверджено на засіданні кафедри

\_\_\_\_\_

Протокол № \_\_\_ від. “ \_\_\_ ” \_\_\_\_\_ 20\_\_ р.

Завідувач кафедри

\_\_\_\_\_  
(підпис)

( \_\_\_\_\_ )  
(Ім'я ПРІЗВИЩЕ)

Переглянуто та затверджено на засіданні кафедри

\_\_\_\_\_

Протокол № \_\_\_ від. “ \_\_\_ ” \_\_\_\_\_ 20\_\_ р.

Завідувач кафедри

\_\_\_\_\_  
(підпис)

( \_\_\_\_\_ )  
(Ім'я ПРІЗВИЩЕ)

## 1. Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Загальна кількість: кредитів – 5,5  годин – 165  змістових модулів – 3  (вид завдання)	Галузь знань <u>12 – Інформаційні технології</u> (шифр і назва)  Спеціальність <u>123 – Комп’ютерна інженерія</u> (шифр і назва)  Рівень вищої освіти: <u>перший</u> (бакалаврський)	Нормативна	
		<b>Рік підготовки:</b>	
		2-й	3-й
		<b>Семестр</b>	
		4-й	5-й
		<b>Лекції</b>	
		34 год.	6 год.
		<b>Практичні, семінарські</b>	
		год.	год.
		<b>Лабораторні</b>	
		50 год.	6 год.
		<b>Самостійна робота</b>	
		81 год.	153 год.
у т.ч. ІНДЗ*: - РГР – 15 год			
Форма підсумкового контролю: залік			

\* – за наявності

## 2. Мета та завдання навчальної дисципліни

Дисципліна “Системне програмування” орієнтована на вивчення основних принципів створення системних програм.

Мета курсу — отримання як теоретичних знань, так і практичних навиків в створенні та використанні компонент системного програмного забезпечення, що повинне забезпечити вміння оперувати з простими і складними структурами даних, користуватися стандартними системними засобами вводу та виводу, працювати з дисковими файлами, розробляти драйвери, управляти розподілом пам'яті, файлами, процесами, здійснювати захист інформації від несанкціонованого доступу.

Для підготовки курсу використана велика низка новітніх наукових та навчально-методичних робіт як зарубіжних так і вітчизняних фахівців у галузі системного програмування, характеристики апаратного та програмного забезпечення комп'ютерів від провідних виробників, тощо.

**Предметом** дисципліни є системне програмне забезпечення комп'ютера та його периферійного обладнання.

**Завданнями** дисципліни є:

- отримання теоретичних знань і практичних навичок, достатніх для проектування і програмування системного програмного забезпечення сучасних комп'ютерів;
- отримання теоретичних знань і практичних навичок, необхідних для експлуатації операційних систем;
- вивчення і реалізація основних алгоритмів, покладених в основу операційних систем.

Процес вивчення дисципліни спрямований на формування елементів наступних **компетентностей** (згідно ОПП «Комп'ютерна інженерія» від 2022 р.):

1) загальних:

Z2. Здатність вчитися і оволодівати сучасними знаннями.

2) фахових:

P2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

P3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

**Програмні результати навчання:**

N1. Знати і розуміти наукові положення, що лежать в основі функціонування комп'ютерних засобів, систем та мереж.

N6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.

*N8. Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.*

*N13. Вміти ідентифікувати, класифікувати та описувати роботу комп'ютерних систем та їх компонентів.*

*N20. Усвідомлювати необхідність навчання впродовж усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення.*

**Очікувані результати навчання.** У результаті вивчення навчальної дисципліни студент повинен

**ЗНАТИ:**

- засоби та задачі системного програмування;
- програмний інтерфейс операційних систем;
- використання і програмування стандартних функцій мов програмування та інтерфейсу системних викликів;
- мови високого рівня в системному програмуванні;
- програмування типових елементів системних програм;
- організацію взаємодії між процесами;
- засоби створення системних служб.

**ВМІТИ:**

- складати, відлагоджувати і виконувати системну програму в середовищі операційної системи UNIX, проаналізувати результат її виконання;
- використовувати інтерфейс системних викликів та стандартну бібліотеку C для складання програм;
- вирішувати проблеми, пов'язані із взаємодією між процесами та захистом даних;
- організувати роботу програми у вигляді системної служби.

Курс складається з наступних частин: теоретичної (лекції), практичної (завдання, лабораторні роботи, тести для самоперевірки) та самостійної роботи (курсової роботи та завдань на лекціях).

Лабораторні роботи виконуються в середовищі операційної системи UNIX, яка застосовується сьогодні практично в усіх сферах інформаційних систем.

### 3. Зміст навчальної дисципліни

**Змістовний модуль 1.** Введення у системне програмування

**Тема 1.** Засоби та задачі системного програмування.

Література: [1, 13].

**Тема 2.** Класифікація системних програм

Література: [1].

**Змістовний модуль 2.** Мови системного програмування

**Тема 1.** Мова асемблера як засіб ефективного програмування.

Література: [1,4,14, 15].

**Тема 2.** Роль мови C у системному програмуванні

Література: [1,4,6, 12].

**Змістовний модуль 3.** Створення програм в середовищі UNIX. Багатомодульні системні програми.

**Тема 1.** Створення и виконання програми в середовищі UNIX. Формати виконуємих файлів. Створення та використання бібліотек

Література: [1,3,5,12].

**Тема 2.** Автоматизація пакетних завдань. Утиліта make.

Література: [1,2,4,9].

**Тема 3.** Взаємодія з командним рядком. Правила формування командного рядка та програмування утиліт. Функції розбору опцій програм.

Література: [1,2,6,7].

**Тема 4.** Змінні оточення. Коди завершення програм.

Література: [1,2,3,5].

**Тема 5.** Середовище програмування UNIX. Системні виклики та бібліотечні підпрограми. Обробка помилок.

Література: [1,3,4].

**Змістовний модуль 4.** Робота з файловою системою

**Тема 1.** Файлова підсистема. Робота с файлами. Примітиви доступу до файлів в системі UNIX.

Література: [1,2,5,7,16].

**Тема 2.** Файли в багатокористувачевому середовищі. Файли с декількома іменами. Отримання і відображення метаданих файлу.

Література: [1,2,4,5].

**Тема 3.** Каталоги. Реалізація каталогів

Література: [1,3,6,10].

**Тема 4.** Пристрої. Драйвери пристроїв. Файловий інтерфейс.

Література: [1,2,5,10].

**Змістовний модуль 5.** Основи управління процесами. Міжпроцесна взаємодія

- Тема 1.** Основи управління процесами. Типи процесів. Атрибути процесів. Системні виклики для отримання інформації про процес.  
Література: [1,3,7].
- Тема 2.** Створення процесів. Запуск нових програм.  
Література: [1,3,5,7].
- Тема 3.** Синхронізація процесів. Очікування зміни стану нащадків. Зомбі-процеси и передчасне завершення програми..  
Література: [1,2,10].
- Тема 4.** Управління процесами. Групи процесів і сеанси.  
Література: [1,3,5].
- Тема 5.** Сигнали. Надійні сигнали. Блокування сигналів. Посилка сигналів. Принципи написання обробників сигналів  
Література: [1,4,6].
- Тема 6.** Міжпроцесна взаємодія за допомогою програмних каналів .  
Література: [1,2,3,10].

#### **Змістовний модуль 6. Створення системних служб**

- Тема 1.** Клієнт серверні-системи. Паралельні сервери. Введення у потоки POSIX.  
Література: [1,2,6,11].
- Тема 2.** Процеси-демони. Правила створення демонів. Взаємодія з журналом помилок. Створення системних служб.  
Література: [1-3,6-8].

#### **Змістовний модуль 7. Оформлення та документування програм.**

- Тема 1.** Стандарти оформлення та документування програм.  
Література: [1, 17].

#### 4. Структура навчальної дисципліни

Назви змістових модулів і тем 1	Кількість годин									
	Денна форма					Заочна форма				
	Усього	у тому числі				Усього	у тому числі			
		л	п	лаб	ср		л	п	лаб	ср
2	3	4	5	6	7	8	9	10	11	
<b>Змістовий модуль 1. Введення у системне програмування</b>										
Тема 1.	4	1			3	4	0,5			3,5
Тема 2.	4	1		1	2	4				4
<b>Змістовий модуль 2. Мови системного програмування</b>										
Тема 1.	6	1		1	4	6				6
Тема 2.	6	1		1	4	6	0,5			5,5
<b>Змістовий модуль 3. Створення програм в середовищі UNIX. Багатомодульні системні програми.</b>										
Тема 1.	8	2		4	2	8	1		1	6
Тема 2.	8	2		4	2	8				8
Тема 3.	8	2		3	3	8				8
Тема 4.	8	2		3	3	8				8
Тема 5.	8	2		2	4	8	0,5		1	6,5
<b>Змістовий модуль 4. Робота з файловою системою</b>										
Тема 1.	8	2		3	3	8	1		1	6
Тема 2.	8	1		3	4	8				8
Тема 3.	8	2		3	3	8				8
Тема 4.	8	2		3	3	8	0,5			7,5
<b>Змістовий модуль 5. Основи управління процесами. Міжпроцесна взаємодія</b>										
Тема 1.	8	1		2	5	8	1		1	6
Тема 2.	7	2		2	3	7				7
Тема 3.	8	2		2	4	8				8
Тема 4.	8	1		3	4	8				8
Тема 5.	8	1		2	5	8			1	7
Тема 6.	8	2		2	4	8				7
<b>Змістовий модуль 6. Створення системних служб</b>										
Тема 1.	8	2		3	3	8	1			7
Тема 2.	10	1		3	6	10			1	9
<b>Змістовий модуль 7. Оформлення та документування програм</b>										
Тема 1.	8	1			7	8				8
<b>Всього годин</b>	<b>165</b>	<b>34</b>	<b>0</b>	<b>50</b>	<b>81</b>	<b>165</b>	<b>6</b>	<b>0</b>	<b>6</b>	<b>153</b>

Форма контролю: **КО** – контрольне опитування (поточне)  
**ІЗ** – індивідуальне завдання (домашнє)  
**КР** – контрольна робота  
**КМ** – контроль модуля за тестовою системою



## 5. Теми семінарських занять

Семінарські заняття не передбачені

## 6. Теми практичних занять

Практичні заняття не передбачені

## 7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин	
1	Створення програм в середовищі UNIX	6	
2	Автоматизація пакетних завдань	8	1
3	Зв'язок з операційною системою. Змінні середовища. Програмування системних утиліт	8	1
4	Використання системних викликів роботи з файлами	6	1
5	Програмування обходу файлового дерева операційної системи	6	1
6	Обробка переривань та сигналів	8	1
7	Створення процесу-демону	8	1
	<b>Разом</b>	<b>50</b>	<b>6</b>

## 8. Самостійна робота

№ з/п	Назва теми	Кількість годин	
		Денна	Заочна
1	2	3.	4.
1	Засоби та задачі системного програмування.. / підготовка до лекцій	2	3,5
2	Класифікація системних програм/ підготовка до лекцій та лабораторних занять	1	4
3	Мова асемблера як засіб ефективного програмування. / підготовка до лекцій та лабораторних занять	2	6
4	Роль мови С у системному програмуванні/ підготовка до лекцій та лабораторних занять	2	5,5
5	Створення и виконання програми в середовищі UNIX. Формати виконуємих файлів. Створення та використання бібліотек/ підготовка до лекцій та лабораторних занять	3	6

1	2	3.	4.
6	Автоматизація пакетних завдань. Утиліта make. / підготовка до лекцій та лабораторних занят/ підготовка до лекцій та лабораторних занять ь	2	8
7	Взаємодія з командним рядком. Правила формування командного рядка та програмування утиліт. Функції розбору опцій програм/ підготовка до лекцій та лабораторних занять	2	8
8	Змінні оточення. Коди завершення програм/ підготовка до лекцій та лабораторних занять/ підготовка до лекцій та лабораторних занять	3	8
9	Середовище програмування UNIX. Системні виклики та бібліотечні підпрограми. Обробка помилок.	4	6,5
10	Файлова підсистема. Робота с файлами. Примітиви доступу до файлів в системі UNIX	3	6
11	Файли в багатокористувачевому середовищі. Файли с декількома іменами. Отримання і відображення метаданих файлу/ підготовка до лекцій та лабораторних занять	3	8
12	Каталоги. Реалізація каталогів/ підготовка до лекцій та лабораторних занять	4	8
13	Пристрої. Драйвери пристроїв. Файловий інтерфейс/ підготовка до лекцій та лабораторних занять с	2	7,5
14	Основи управління процесами. Типи процесів. Атрибути процесів. Системні виклики для отримання інформації про процес/ підготовка до лекцій та лабораторних занять	3	6
15	Створення процесів. Запуск нових програм/ підготовка до лекцій та лабораторних занять	3	7
16	Синхронізація процесів. Очікування зміни стан/ підготовка до лекцій та лабораторних занять у нащадків. Зомбі-процеси и передчасне завершення програми..	34	8
17	Управління процесами. Групи процесів і сеанс/ підготовка до лекцій та лабораторних занять и.	2	8
18	Сигнали. Надійні сигнали. Блокування сигналів. Посилка сигналів. Принципи написання обробників сигналів/ підготовка до лекцій та лабораторних занять	5	7

19	Міжпроцесна взаємодія за допомогою програмних каналів/ підготовка до лекцій та лабораторних занять	4	8
<b>1</b>	<b>2</b>	<b>3.</b>	<b>4.</b>
20	Клієнт серверні-системи. Паралельні сервери. Введення у потоки POSIX/ підготовка до лекцій та лабораторних занять.	3	7
21	Процеси-демони. Правила створення демонів. Взаємодія з журналом помилок. Створення системних служб/ підготовка до лекцій та лабораторних занять	5	9
22	Стандарти оформлення та документування програм/ підготовка до лекцій та лабораторних занять	5	8
23	Розрахунково-графічна робота	15	15
	<b>Разом</b>	81	153

## 9. Методи навчання

Основна підготовка студентів здійснюється на лекційних та лабораторних заняттях.

Під час викладання курсу використовуються такі методи навчання:

- словесні (лекція, пояснення);
- наочні (презентація);
- практичні (лабораторні роботи);
- робота з літературними джерелами (самостійна робота студентів).

## 10. Форми контролю і методи оцінювання

Протягом семестру студенти виконують по 2 теоретичні контрольні роботи.

Перелік питань та приклади завдань містяться у п. 11.

Під час підсумкового контролю студент повинен відповісти на 2 теоретичні запитання з переліку, наведеному у п. 11.3, та вирішити завдання

### 10.1. Критерії оцінювання на підсумковому модульному контролі:

1. Відповідь повинна бути повною і короткою. Вона не повинна мати в собі матеріал, що не відноситься до сутті питання.
2. Чітко формулювати твердження, вправно застосовувати необхідні формули і знання основних питань програми.
3. Відповіді, що мають помилкові твердження оцінюються виходячи з близькості відповіді до правильної.

4. Пропуски в обґрунтуванні тверджень враховуються і це призводить до зменшення кількості балів.
5. Малі недоліки, неточності при викладенні матеріалу, зменшують кількість балів.
6. Незнання і нерозуміння основної ідеї теоретичного питання або задачі призводить до зняття до 90 % балів.
7. Якщо відповідь на питання відсутня то виставляється нуль балів.

## **11. Перелік контрольних, екзаменаційних (залікових) питань**

### **11.1. Питання до модульної контрольної роботи №1**

#### **. Контрольні запитання**

1. Перелічить рівні програмного забезпечення?
2. Чим відрізняється системне програмне забезпечення від прикладного?
3. Наведіть приклади системних утиліт, що використовуються в Вашій версії системи UNIX?
4. Чому в середовищі UNIX для системного програмування переважно використовується мова C?
5. Яке призначення приділяється препроцесору мови?
6. Що таке умовна компіляція, яку виробляє препроцесор?
7. З якою метою застосовується умовна компіляція?
8. Назвіть оператори препроцесора.
9. Навіщо вони використовуються?
10. Які директиви препроцесора використовуються найчастіше у програмах, написаних мовою?
11. Що таке макровизначення препроцесора? Як воно реалізується?
12. Що таке макропідстановка?
13. Як записується директива умовної компіляції?
14. Як працює директива `#undef`? Якою директивою при компіляції підключаються файли?
15. У чому різниця між використанням кутових дужок (`<...>`) та лапок при вказівці імені файлу?
16. Чим погано макровизначення, яке обчислює квадрат свого параметра як `X*X`?
17. Як організувати одноразове включення заголовного файлу в текст, що компілюється?
18. Для чого використовується змінна `errno`? Які функції використовуються для полегшення виведення повідомлень про помилки?
19. Яким чином в командному рядку задаються аргументи програми? Яким чином функція `main()` дістає доступ до списку аргументів?
20. Що таке змінна оточення? В якому форматі вони зберігаються? Яким чином програма дістає доступ до змінних оточення?
21. Що таке обробник виходу? Як його зареєструвати?

22. Що таке системний виклик? Чим він відрізняється від функцій стандартної бібліотеки? Як отримати інформацію про системний виклик?
23. В яких ситуаціях виникають помилки системних викликів? Яким чином системний виклик інформує про помилки?
24. Які способи вказівки імен файлів підтримуються в ОС UNIX?
25. Що таке файловий дескриптор? Для чого вони використовуються?
26. Які файлові дескриптори вважаються стандартними?
27. Чим відрізняється системний виклик *open()* від системного виклику *creat()*?
28. Чи має смисл комбінація прапорів *O\_WRONLY* | *O\_EXCL* | *O\_TRUNC* при відкритті файлу? Чому?
29. Чи можна в програмі кілька разів відкрити один і той же файл?
30. Що станеться, якщо маску створення файлів задати рівною 777 (у вісімковій системі)? Перевірте результати за допомогою команди *umask*.
31. Як змінюється поточна позиція в результаті запису в файл (читання з файлу)?
32. Що служить ознакою кінця файлу при читанні?
33. Чи можна збільшити довжину файлу, не проводячи запис в нього? Які переваги, на вашу думку, дає використання системних викликів *pread()* і *pwrite()*?
34. Як змінити маску, що накладається на набір прав доступу при створенні файлу?
35. Яка функція використовується для відкриття каталогу?
36. За допомогою якої структури можна отримати запис каталогу?
37. Як змінити поточний каталог?
38. Як отримати ім'я поточного каталогу?
39. Що таке закладка?
40. Які складнощі виникають при виконанні операцій над ієрархією каталогів?
41. Для чого потрібні каталоги?
42. Що таке каталог? Які дані зберігаються в запису каталогу?
43. Що таке порожній каталог?
44. Як трактуються права доступу до каталогу?
45. Як створити каталог? Які обмеження на створення каталогу є у системі?
46. Чому для видалення каталогу застосовується власний системний виклик?
47. Для кожного процесу визначений також кореневий каталог. Кореневий каталог процесу може бути змінений за допомогою функції *chroot()*. Знайдіть опис цієї функції в своєму довідковому керівництві. У яких випадках вона може бути корисна?

### Приклад завдання модульної контрольної роботи №1.

1. Який з каталогів за умовчанням зберігає конфігураційні файли в UNIX?
  1. /usr

2. /conf
3. /var
4. /etc

2. Перетворити числове представлення прав доступу в символічне:: 4000

3. Файлова система є частиною:

1. дискових систем;
2. драйверів дисків;
3. ОС;
4. користувацьких програм.

4. Що таке системний виклик? Чим він відрізняється від функцій стандартної бібліотеки?

5. Що таке обробник виходу? Як його зареєструвати?

6. Що таке каталог? Що містить порожній каталог? Як перейти в батьківський каталог?

7. Які категорії прав доступу до файлів розрізняють в ОС UNIX?5. Що таке файловий дескриптор? Для чого вони використовуються?

8. Написати функцію, що визначає, чи існує заданий файл? Аргументом є колійне ім'я файлу. Функція повертає значення 0, якщо файл існує, і -1 у протилежному випадку.

9. Чому для видалення каталогу застосовується власний системний виклик?

## 11.2. Питання до модульної контрольної роботи №2

### Контрольні запитання

1. Що таке процес?
2. Які атрибути процесу є найважливішими?
3. Які атрибути дочірнього процесу відрізняються від атрибутів батьківського процесу?
4. В чому різниця між реальним та ефективним ідентифікаторами користувача?
5. Яким чином може бути породжений новий процес? Яка структура нового процесу?
6. Як процес може довідатися, є він дочірнім чи батьківським?
7. Якщо процес-предок відкриває файл, а потім породжує процес-нащадок, а той, у свою чергу, змінює положення покажчика читання-запису файлу, то чи зміниться положення покажчика позиції файлу процесу-батька?
8. У чому різниця між різними формами системних викликів типу *exec()*?
9. Які процеси прийнято називати процесами-зомбі?
10. Чим відрізняється реальний ідентифікатор користувача від ефективного?
11. Які обмеження є у звичайного користувача на зміну ідентифікаторів користувача і групи?
12. Для чого використовуються сеанси і групи процесів?

13. Який процес не може покинути свою групу процесів групи?
14. Який процес є лідером сеансу?
15. Що відбувається при завершенні лідеру сеансу?
16. Що таке відносний пріоритет? Як його змінити?
17. Напишіть програму, яка виводить якомога більше відомостей про поточний процес: *PID*, *PPID*, відкриті файли, поточний каталог, значення відносного пріоритету і так далі. Як можна узнати, які файли відкриті? Якщо декілька дескрипторів файлів посилаються на один і той же файл, вкажіть це.
18. Що станеться, якщо процес-нащадок завершиться раніше, ніж процес-предок здійснить системний виклик *wait()*?
19. В чому перевага системного виклику *waitpid()* перед системним викликом *wait()*?
20. Як система повідомляє батьківський процес про завершення дочірнього процесу?
21. З якою апаратною концепцією тісно пов'язане поняття сигналу?
22. Для чого використовуються сигнали в ОС UNIX? Які види сигналів існують в ОС UNIX?
23. Які види реакції на сигнал за умовчанням передбачені в ОС UNIX? Як виглядає обробник сигналу? Які стандартні обробники сигналів передбачені системою?
24. Чи можна вважати сигнали ОС UNIX універсальним механізмом взаємодії між процесами?
25. Як ви думаєте, чому структура *siginfo* поміщає в поле *a* реальний, а не ефективний ідентифікатор користувача?
26. Що таке блокування сигналу?
27. Як послати сигнал групі процесів?
28. В чому різниця між системними викликами *abort()* і *kill()* з аргументом *SIGABRT*?
29. В чому різниця між системними викликами *abort()* і *raise()*?
30. Для чого використовується системний виклик *pause()*?
31. Опишіть порядок створення однобічного програмного каналу від батьківського процесу до дочірнього.
32. Відкриваючи іменованій канал для читання, процес припиняється доти, поки ще один процес не відкриє канал для запису. Як Ви думаєте, чому?
33. Що розуміється під гарантією атомарності операції запису? У яких випадках система не гарантує таку атомарність?
34. Який максимальний розмір програмного каналу і чому?
35. Перелічить всі демони у вашій системі і вкажіть їх функціональне призначення.
36. Як запобігти повторному запуску демона?
37. Як демони повідомляють систему про помилки?
38. Де зазвичай зберігаються конфігураційні файли демонів?

## Типове завдання модульної контрольної роботи №2.

1. Процес - це ...
  - a. мережний інтерфейс контролеру блокових передач;
  - b. число, приписане операційною системою кожній задачі;
  - c. динамічна сутність програми, її код в процесі свого виконання;
  - d. система, що виконує повторювану операцію/
2. Серед функцій зміни програми системним викликом є
  - a. execlp
  - b. execl
  - c. execvp
  - d. execl
  - e. execv
  - f. execve
3. Системний виклик для очікування завершення нащадку з ідентифікатором pid виглядає наступним чином
  - a. waitpid(pid, NULL, WNOHANG);
  - b. waitpid(-pid, NULL, WNOHANG);
  - c. waitpid(pid, &status, WNOHANG);
  - d. waitpid(-pid, &status, WNOHANG);
  - e. waitpid(pid, &status, 0);
  - f. waitpid(-pid, &status, 0);
4. Як дізнатися атрибути процесу: ідентифікатор процесу, ідентифікатор батьківського процесу, ідентифікатор власника і групи?
5. Що успадковується при запуску нової програми в адресному просторі процесу?
6. Поясніть, у чому полягає відмінність між такими поняттями, як «процес» і «потік»?
7. Як запобігти повторному запуску демона?

### 11.3. Питання до підсумкового контролю

У підсумковий контроль включено контрольні питання до модульних контрольних робіт.

Змістовий модуль № 4

### 12. Розподіл балів, які отримують студенти

Змістовий модуль №1		Змістовий модуль № 2		Змістовий модуль № 3					Змістовий модуль № 4			
T1	T2	T1	T2	T1	T2	T3	T4	T5	T1	T2	T3	T4



2	2	3	2	3	2	3	2	3	3	3	3	3
Змістовий модуль № 5						Змістовий модуль №6		Змістовий модуль №7	РГ Р	Підсумковий контроль (іспит)		Сума балів
Т 1	Т 2	Т 3	Т 4	Т 5	Т 6	Т1	Т2	Т1				
3	3	3	3	3	2	3	3	3		10	30	100

Т1, Т2 ... – теми змістових модулів

### 12.1. Критерії оцінювання на підсумковому контролі:

1. Відповідь повинна бути повною і короткою. Вона не повинна мати в собі матеріал, що не відноситься до сутті питання.
2. Чітко формулювати твердження, вправно застосовувати необхідні формули і знання основних питань програми.
3. Відповіді, що мають помилкові твердження оцінюються виходячи з близькості відповіді до правильної.
4. Пропуски в обґрунтуванні тверджень враховуються і це призводить до зменшення кількості балів.
5. Малі недоліки, неточності при викладенні матеріалу, зменшують кількість балів.
6. Незнання і нерозуміння основної ідеї теоретичного питання або задачі призводить до зняття до 90 % балів.
7. Якщо відповідь на питання відсутня то виставляється нуль балів.

### 12.2. Шкала оцінювання: національна та ECTS

Загальна сума балів	Оцінка ECTS	Національна шкала	
90 — 100	A – «відмінно»	5 «відмінно»	« з а л і к »
85 — 89	B – «дуже добре»	4 «добре»	
75 — 84	C – «добре»		
70 — 74	D – «задовільно»	3 «задовільно»	
60 — 69	E – «допустимо»		
35 — 59	F – «незадовільно з можливістю повторного складання»	2 «незадовільно»	« н е з а л і »
0 — 34	FX – «незадовільно з обов'язковим повторним курсом»		

			К »
--	--	--	--------

### 13. Методичне забезпечення

1. Операционные системы: конспект лекций / Н. Ф. Трубина, О. И. Розновец ; Одес. нац. ун-т им. И. И. Мечникова, Ин-т математики, экономики и механики, Каф. мат. обеспечения компьютер. систем. - О. : Астропринт, 2010. - 188 с.
2. Трубіна Н.Ф., Лісіцина І.М. Методичні вказівки до виконання лабораторних робіт з дисципліни «Системне програмування» для студентів спеціальності 126 – «Інформаційні системи та технології». Електронне видання. – Одеса: ОНУ, 2021. – 84 с
3. Трубіна Н.Ф. Методичні вказівки до виконання курсової роботи з дисципліни «Системне програмування» для студентів спеціальності 126 – «Інформаційні системи та технології». Електронне видання. – Одеса: ОНУ, 2021. – 20 с

### 14. Рекомендована література

#### 14.1. Основна література

1. Князева Н.О., Жуковецька С.Л., Трубіна Н.Ф. Системне програмування – Одеса: ВМВ, 2013 – 272с.
2. K. C. Wang Systems Programming in Unix/Linux – Springer International Publishing AG, part of Springer Nature, 2018 – 452 с
3. Рочкинд М. Программирование для UNIX. 2-е изд. Пере раб. И доп. – М. Издательско-торговый дом «Русская редакция»; СПб. БХВ – Санкт-Петербург, 2005.

#### 14.2. Допоміжна література

4. Лав Р. Linux. Системное программирование. Пер. с англ., 2-е изд. — СПб.: Питер, 2014. — 448 с.:
5. Stevens, W. Richard, Rago, Stephan A., Advanced Programming in the UNIX Environment, 3rd Edition, AddisonWesley, 2013.
6. М. Митчелл, Д.Оулдем, А.Саммюэл. Программирование для LINUX. Профессиональный подход. Пер. с англ., – М.: Издательский дом «Вильямс», 2002
7. К. Хэвиленд, Д. Грэй, Б. Салама. Системное программирование в UNIX. – М., ДМК Пресс, 2000
8. Mecklenburg Robert. Managing Projects with GNU Make, 3rd Edition – O'Reilly Media, 2004, Pages: 302
9. Michael Kerrisk. The Linux Programming Interface. A Linux and UNIX System Programming Handbook [. – No Starch Press, October 2010, Pages: 1552
10. Соколова Н.О., Вовк С.М., Єгоров А.О., Синхронізація потоків в операційних системах: навч. посіб – Дніпропетровськ: "Ліра", 2015. – 96с.

11. An Introduction to GCC: For the GNU Compilers GCC and G++ 4244 2nd Edition
12. Системне програмування : підручник для студентів напряму "Комп'ютерна інженерія" ВНЗів / Олександр Миколайович Рисований; В.о. Нац. техн. ун-т "Харківський політехнічний інститут". – Харків : НТУ "ХПІ", 2010.– 912 с.
13. Галісеєв Г.В. Системне програмування. – Київ: Університет «Україна», 2019. – 113 с.
14. Sri Manikanta Palakollu. Pragmatic Example Applications in Linux and Unix-Based Operating/ Pragmatic Example Applications in Linux and Unix-Based Operating Systems – Systems Apress,-Berkeley, CA 27,October 2020, Pages: 1552
15. Гоменюк С. І., Чопоров С. В., Лісняк А. О., Кудін О. В., Гребенюк С. М. Системне програмування: розробка багатопотокових програм в операційній системі Linux: навчальний посібник для здобувачів ступеня вищої освіти бакалавра спеціальності “Інформаційні системи та технології” освітньо-професійної програми “Інформаційні системи та технології”. Запоріжжя: Запорізький національний університет, 2021. 120 с.

### **15. Інформаційні ресурси**

1. UNIX Application and System Programming – Режим доступу: [https://www.cs.hunter.cuny.edu/~sweiss/course\\_materials/unix\\_lecture\\_notes.php](https://www.cs.hunter.cuny.edu/~sweiss/course_materials/unix_lecture_notes.php)
2. –UNIX / LINUX Tutorial – Режим доступу: <https://www.tutorialspoint.com/unix/index.htm>
3. Using the GNU Compiler Collection (GCC) – Режим доступу: <http://www.physics.ohio-state.edu/doco/gnu/gcc/gcc.html>
4. Book Online: Introduction to Systems Programming: a Hands-on Approach by Gustavo A. Junipero Rodriguez-Rivera and Justin Ennen – Режим доступу: <https://www.cs.purdue.edu/homes/grr/SystemsProgrammingBook>
5. GCC online documentation – Режим доступу: <https://gcc.gnu.org/onlinedocs/>