

# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет радіофізики, електроніки та комп'ютерних систем

Кафедра комп'ютерної інженерії

«ЗАТВЕРДЖУЮ»

Заступник декана з навчальної роботи

\_\_\_\_\_ Наталія ГОРБОВЦОВА

« \_\_\_\_ » \_\_\_\_\_ 2023 року

## РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Програмування для UNIX-систем

-

для студентів

галузь знань	12 “Інформаційні технології”
спеціальність	123 “Комп'ютерна інженерія”
рівень вищої освіти	перший
освітня програма	“Інженерія комп'ютерних систем і мереж”
вид дисципліни	вибіркова за блоком "мережеві адміністратори"

Форма навчання	Денна
Навчальний рік	2023/2024
Семестр	4
Кількість кредитів ECTS	4
Мова викладання	українська
Форма заключного контролю	залік

**Викладач:**

**Сергій ПОГОРІЛИЙ**, доктор технічних наук, професор кафедри комп'ютерної інженерії

Пролонговано: на 20\_\_/20\_\_ н. р. \_\_\_\_\_ ( \_\_\_\_\_ ) « \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

на 20\_\_/20\_\_ н. р. \_\_\_\_\_ ( \_\_\_\_\_ ) « \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**Розробник:**

**Сергій ПОГОРІЛИЙ**, доктор технічних наук, професор кафедри комп'ютерної інженерії

**«ЗАТВЕРДЖЕНО»**

Завідувач кафедри комп'ютерної інженерії

\_\_\_\_\_ **Юрій БОЙКО**

Протокол № \_\_\_\_ від « \_\_\_\_ » \_\_\_\_\_ 2023 р.

Схвалено науково-методичною комісією факультету радіофізики, електроніки та комп'ютерних систем

Протокол № \_\_\_\_ від « \_\_\_\_ » \_\_\_\_\_ 2023 р.

Голова науково-методичної комісії

**Сергій РАДЧЕНКО**

« \_\_\_\_ » \_\_\_\_\_ 2023 року.

## ВСТУП

### 1. Мета дисципліни

Метою навчальної дисципліни «Програмування для UNIX-систем» є оволодіння студентами 2-го курсу знаннями з основних концепцій створення та використання POSIX-сумісних операційних середовищ для відкритих комп'ютерних систем і формуванні світогляду студентів у галузі операційних систем (ОС), навичок роботи із ними у подальшому навчанні. Курс передбачає оволодіння методикою корпоративних застосувань ОС у подальшій професійній діяльності.

### 2. Попередні вимоги до опанування або вибору навчальної дисципліни

До вивчення дисципліни «Програмування для UNIX-систем» студентам потрібно засвоїти знання з математики та суміжних дисциплін в курсах «Дискретна математика», «Програмування» та «Апаратне та програмне забезпечення комп'ютерних систем».

### 3. Анотація навчальної дисципліни

До спектру тем дисципліни «Програмування для UNIX-систем» включено: розділ з класифікації та архітектури ОС (концепція відкритої комп'ютерної системи (відповідно до стандарту POSIX 1004.3.0.5), таксономія та класифікація ОС (монолітні, багаторівневі, мікроядрові та мікросервісні архітектури), системи реального часу, гіпервізор); функціональні можливості новітніх ОС (програмні агенти, спеціальні операції, інтерфейс прикладного програмування (API), середовища розробки застосунків); основні групи команд та механізми ОС UNIX (наявність різних інтерпретаторів команд, файлова система, механізм повноважень, групи користувачів, регулярні вирази, редактори тексту, сі-подобна сценарна мова програмування (awk), інструментальні засоби створення застосунків); сценарне програмування мовою оболонки (інтерпретатори команд sh, ksh, bash, POSIX shell, використання мови Python, програмування в середовищі Internet) тощо.

### 4. Завдання навчальної дисципліни (навчальні цілі)

Надати основні відомості курсу «Програмування для UNIX-систем», які складають важливу частину загально-наукової і інженерної підготовки; навчити застосовувати базові відомості у професійній діяльності, розвивати у студентів теоретичне та інженерне мислення; навчити застосовувати знання та уміння в процесі проектування, та програмування в середовищах комп'ютерних систем і мереж.

Дисципліна спрямована на формування програмних компетентностей [6]:

- ЗК1. Здатність до абстрактного мислення, аналізу і синтезу.
- ЗК 2. Здатність вчитися і оволодівати сучасними знаннями.
- ФК1. Здатність застосовувати законодавчу та нормативно-правову базу, а також державні та міжнародні вимоги, практики і стандарти з метою здійснення професійної діяльності в галузі комп'ютерної інженерії.
- ФК2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.
- ФК3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.
- ФК11. Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів.

## 5. Результати навчання за дисципліною

Результат навчання (1 - знати; 2 - вміти; 3 - комунікація; 4 - автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
<b>1</b>	<b>знати:</b>	лекційні заняття	письмові модульні контрольні роботи, лабораторні роботи та залік	до 45
1.1	Таксономію архітектур новітніх комп'ютерних систем, концепцію програмних агентів, їх види та мову ACL	лекція	залік	10
1.2	Концепцію процесів та потоків в ОС UNIX, методи їх створення та правила взаємодії	лекція	залік	5
1.3	Основні групи команд ОС UNIX, фонову обробку команд, методи реалізації інтерфейсу API, файлової системи, її монтування та розмонтування зовнішніх пристроїв (каталог dev)	лекція	залік	5
1.4	Концепції програмних фільтрів та конвеєрів, методи їх створення та правила роботи	лекція	залік	10
1.5	Порівняльний аналіз інтерпретаторів команд ОС UNIX та основні положення стандарту POSIX shell. Методи обробки опцій командного рядка	лекція	залік	5
1.6	Основні конструкції команд створення сценаріїв мовою оболонки, правила їх виконання. Володіти методами налагоджування сценаріїв	лекція	залік	5
1.7	Методи використання мови програмування Python як оболонки ОС UNIX. Методики створення застосунків мовою Python для мережі Internet	лекція	залік	5
<b>2</b>	<b>вміти:</b>	Лабораторні роботи	письмові модульні контрольні роботи (МКР) та лабораторні роботи	до 45
2.1	Використовувати основні групи команд та здійснювати навігацію в файловій системі	Лабораторні роботи	МКР	15
2.2	Формувати в середовищі UNIX конвеєри команд, фільтри та фонові процеси	Лабораторні роботи	МКР	10
2.3	Створювати та налагоджувати сценарії мовою bash оболонки (в тому числі зі змінною кількістю аргументів та обробкою опцій командного рядка)	Лабораторні роботи	МКР	10
2.4	Використовувати засоби мови Python як оболонки ОС UNIX. Створювати застосунки мовою Python для мережі Internet	Лабораторні роботи	МКР	10
<b>3</b>	<b>комунікація:</b>	Лекційні заняття та лабораторні роботи	письмові МКР та залік	до 5
3.1	Здатність грамотно будувати наукову комунікацію як в усній так і письмовій формах, підбирати правильну термінологію	Лабораторні роботи	МКР	2
3.2	Здатність до командної роботи у великих науково-дослідницьких проектах	Лабораторні роботи	МКР	3
<b>4</b>	<b>автономність та відповідальність:</b>	Лекційні та лабораторні роботи	письмові МКР, лабораторні роботи	до 5
4.1	Здатність до самостійного пошуку наукової літератури або інших джерел інформації для розв'язання поставленої перед студентом науково-дослідницької задачі	Самостійна робота студента	Лабораторні роботи та самостійна робота студента	5

## 6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	Код													
	1.1	1.2	1.3	1.4	1.5	1.6	1.7	2.1	2.2	2.3	2.4	3.1	3.2	4.1
<b>Програмні результати навчання (назва)</b>														
ПРН3. Знати новітні технології в галузі комп'ютерної інженерії.	+	+	+	+	+	+	+							
ПРН6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.								+	+	+	+			
ПРН10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати типове для спеціальності обладнання.					+	+	+	+	+	+	+			
ПРН18. Використовувати інформаційні технології та для ефективного спілкування на професійному та соціальному рівнях.												+	+	
ПРН19. Здатність адаптуватись до нових ситуацій, обґрунтовувати, приймати та реалізовувати у межах компетенції рішення.														+

## 7. Схема формування оцінки

### 7.1. Форми оцінювання

Рівень досягнення всіх запланованих результатів навчання визначається за результатами роботи на лабораторних заняттях (в тому числі за виконання домашніх завдань), написання письмових контрольних робіт, та письмового заліку. Внесок результатів навчання у підсумкову оцінку, за умови їх опанування на належному рівні:

- результати навчання 1.1 – 1.7 [знання] – до 45 %;
- результат навчання 2.1 – 2.4 [вміння] – до 45%;
- результат навчання 3.1-3.2 [комунікація] – до 5%;
- результат навчання 4.1 [автономність та відповідальність] – до 5%;

Форми оцінювання:

- **семестрове оцінювання:** Навчальний семестр має два змістовні модулі. Після завершення лекції №4 проводиться перша письмова модульна контрольна робота, а після лекції №8 – друга письмова модульна контрольна робота. Обов'язковим для допуску до заліку є: виконання лабораторного практикуму у повному обсязі, написання обох контрольних робіт із сумарною кількістю балів не менше 35. Замість модульної контрольної роботи або колоквиуму можуть бути зараховані самостійно виконані студентом завдання за тематикою лекційних занять.
- **підсумкове оцінювання (у формі заліку):** форма заліку – письмова. Залік складається з 2 питань і однієї задачі. Питання оцінюються по 13 балів, задача – 14 балів. Всього за залік можна отримати від 0 до 40 балів. Умовою досягнення позитивної оцінки за дисципліну є отримання не менш ніж 60 балів, оцінка за залік не може бути меншою за 25 балів.
- **умови допуску до підсумкового заліку:** умовою допуску до заліку є отримання студентом сумарно не менше, ніж *критично-розрахунковий мінімум* за семестр, 35 балів. студенти, які протягом семестру сумарно набрали меншу кількість балів, ніж критично-розрахунковий мінімум у **35 балів**, для одержання допуску до підсумкового заліку обов'язково повинні виконати додаткове завдання за тематикою лабораторних занять.

У випадку відсутності студента з поважних причин відпрацювання та перездачі модульної контрольної роботи допуск до іспиту здійснюються у відповідності до “Положення про організацію освітнього процесу у Київському національному університеті”.

### 7.2. Організація оцінювання

Оцінювання за формами контролю:

Семестрова робота	Кількість балів	
	Min. – 10	Max. – 20
Модульна контрольна робота 1	10	15
Модульна контрольна робота 2	10	15
Лабораторні роботи	15	30

Орієнтований графік оцінювання:

Форма оцінювання	Орієнтовний період для здійснення відповідної форми оцінювання
Модульна контрольна робота 1	лютий
Модульна контрольна робота 2	квітень
Лабораторні роботи	лютого по квітень
Добір балів/додаткова контрольна робота	травень
Залік	травень

Розрахунок балів, які отримують за умови успішної здачі заліку:

Значення	Протягом семестру	Залік	Підсумкова оцінка
Мінімум	35	25	60
Максимум	60	40	100

### 7.3. Шкала відповідності оцінок

Оцінка (за національною шкалою) / National grade	Рівень досягнень, % / Marks, %
<b>Зараховано</b> / Excellent	90 – 100%
<b>Зараховано</b> / Good	75 – 89%
<b>Зараховано</b> / Satisfactory	60 – 74%
<b>Не зараховано</b> / Fail	0 – 59%

### 8. Структура навчальної дисципліни. Тематичний план лекційних занять

№ з/п	Назва теми	Кількість годин		
		Лекції	Лабораторні роботи	Самостійна робота студента
1	Концепція відкритої системи стосовно ОС. Основоволожні поняття та архітектура ОС. Концепції процесів та потоків і їх призначення	2	-	5
2	Відкриті системи та відкриті специфікації, призначення концепції відкритих систем. Ресурси ОС, підходи до їх класифікації. Сервіси ядра на етапі виконання. Графічні оболонки операційних систем	2	-	5
3	Аналіз виконання процесів і потоків для програмних платформ. Аналіз виконання процесів і потоків для програмних платформ UNIX і інших POSIX-сумісних систем. Функції API для створення процесів та потоків.	2	-	5
4	Функції для формування та роботи із процесами та потоками. Аналіз засобів синхронізації для платформи Win32. Об'єкти синхронізації: подія (event), м'ютекс (mutex) та семафор (semaphore). Функції синхронізації	2	-	5
5	Основні команди взаємодії користувачів в системі: <b>mail, mailx, write, mesg, news</b> . Команди <b>date, echo, banner, passwd, lp, touch</b> . Online-посібник з команд системи	2	2	5
6	Деревоподібна структура файлової системи UNIX, найбільш важливі каталоги: <b>/bin, /usr, /usr/bin, /usr/lib, /usr/mail, /usr/man, /usr/local/bin, /usr/contrib/bin, /etc, /users, /dev</b> . Абсолютний та відносний шляхи, навігація у файлової системі. Монтування та демонтування логічних пристроїв. Команди <b>pwd, ls, cd, mkdir, rmdir</b>	2	2	5
7	Файли та права доступу (повноваження) до них в операційній системі UNIX. Повноваження груп користувачів. Команди зміни повноважень.	2	2	5
8	Призначення оболонки shell та shell-змінні, встановлення значень та звернення до них. Області даних shell-змінних: локальна та оточення. Команди роботи з областями даних: <b>set, unset, env, export</b> . Передачі значень shell-змінних.	2	2	5

9	Аналіз різних інтерпретаторів команд shell в операційній системі UNIX, еволюція їх розвитку та сумісність. Ретроспективний аналіз інтерпретаторів команд shell: <b>sh, csh, tcsh, ksh, bsh, bash, POSIX shell</b>	2	-	5
10	Програмування мовою shell, сценарії та методи їх активізації. Спеціальні змінні shell <b># \$ @ та * .</b> Спеціальні команди для обробки сценаріїв: <b>shift, read, expr</b> . Інтерактивні команди shell. Файли <b>.profile</b>	2	-	5
11	Галуження в сценаріях мовою shell. Коди завершення виконання команд. Команда <b>test</b> , перевірки файлів, рядків та чисельних виразів. Керуючі конструкції для розгалуження обчислень <b>if</b> та <b>case</b> . Керуючі конструкції для організації циклічних обчислень <b>while, until, for</b> , та <b>foreach</b> , приклади конструкцій. Команди shell-сценаріїв <b>break, continue</b> та <b>exit</b>	2	2	5
12	Застосування концепції сигналів та пасток (traps) як засобів міжпроцесної взаємодії при створенні сценаріїв. Команди <b>kill</b> та <b>trap</b> , екранування сигналів. Методика створення сценаріїв та методи їх налагоджування	2	-	5
13	Приклади системного програмування для ОС UNIX мовою C++ з використанням функцій API	2	-	5
14	Методи використання мови програмування Python як оболонки ОС UNIX. Звернення до системних ресурсів ОС UNIX за допомогою модулів Python	2	2	5
15	Методики створення застосунків мовою Python для мережі Internet. Звернення до системних ресурсів ОС UNIX за допомогою модулів Python, приклад створення клієнт-серверної взаємодії	2	2	6
<b>Всього</b>		<b>30</b>	<b>14</b>	<b>76</b>

Загальний обсяг **120** год., в тому числі:  
 Лекції **30** год.  
 Лабораторні роботи **14** год.  
 Самостійна робота **76** год.

## 9. Рекомендована література

### Основні джерела:

- [1] Погорілий С.Д. Програмне конструювання. Підручник за редакцією академіка АПН України Третяка О.В., видання 2-е. Київ : ВПЦ "Київський університет", Київ, 2007, 440 с.
- [2] С.Д. Погорілий, В.А. Мар'яновський. Програмування для UNIX-систем. Навчальний посібник до лабораторних робіт. Видавнича лабораторія радіофізичного факультету Київського національного університету імені Тараса Шевченка, 2012, 88 с.
- [3] Погорілий С.Д. "Автоматизація наукових досліджень. Основоположні математичні відомості. Програмне забезпечення" за редакцією академіка АПН України Третяка О. В. Київ: ВПЦ "Київський університет", 2002.



[4] Погорілий С.Д. “Автоматизація наукових досліджень. Основоположні математичні відомості. Програмне забезпечення. Задачі та лабораторні практикуми” за редакцією академіка АПН України Третяка О. В. Київ : ВПЦ “Київський університет”, 2002.

[5] А. В. Анісімов, А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий. Програмування числових методів мовою PYTHON. За редакцією чл.-кор. НАН України А. В. Анісімова. Підручник із грифом МОН України. ВПЦ «Київський університет», 2015 р. 640 с.

**Додаткові джерела:**

[6] Стандарт вищої освіти України. Перший (бакалаврський) рівень. Галузь знань 12 – Інформаційні технології. Спеціальність 123 – Комп’ютерна інженерія. Видання офіційне. МОН України, Київ, 2018 р.