

**Факультет радіофізики, електроніки та комп'ютерних систем
Київського національного університету імені Тараса Шевченка**

«ЗАТВЕРДЖЕНО»

Декан

_____ Анісімов І.О.

« » _____ 2018 року

Освітня програма «Інженерія комп'ютерних систем і мереж»

**Робоча програма навчальної дисципліни
Програмування для UNIX-систем**

галузь знань 12 Інформаційні технології
спеціальність 123 Комп'ютерна інженерія
форма навчання денна
мова навчання українська

Навчальний рік
Семестр

2018/2019
4

«УЗГОДЖЕНО»

Гарант ОП Інженерія комп'ютерних систем і мереж

Завідувач кафедри Комп'ютерної інженерії

_____ Баужа О.С.

_____ Погорілий С.Д.

«27» серпня 2018 року

«27» серпня 2018 року

Схвалено НМК факультету радіофізики, електроніки та комп'ютерних систем
протокол № від « » _____ 2018 року

Голова науково-методичної комісії _____ Нетреба А.В.

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

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

Викладач: Погорілий Сергій Дем'янович, д.т.н, професор, завідувач кафедри
Комп'ютерної інженерії

Робоча програма навчальної дисципліни
Програмування для UNIX-систем
(2 курс, 4 семестр)

Лекції	30 год.
Лабораторні роботи	14 год.
Модульна контрольна робота 1	2 год.
Модульна контрольна робота 2	2 год.
Форма заключного контролю	залік

1. Статус дисципліни: Навчальна дисципліна «Програмування для UNIX-систем» є дисципліною вільного вибору підготовки фахівців за освітнім рівнем «бакалавр» галузі знань «12 – Інформаційні технології» зі спеціальності «123 – Комп’ютерна інженерія» (спеціалізація – мережеві адміністратори (МА)).

2. Анотація навчальної дисципліни: Методологічний розвиток методів обчислень є першопричиною зародження і еволюції обчислювальної техніки та сучасного програмного забезпечення. Процес обчислення, як відомо, є алгоритмічною процедурою перетворення певної інформації з метою отримання конкретного результату або доведення неможливості такого обчислення. Обчислювальна техніка та інформаційні технології щільно оперують поняттям інформації, яка для них є водночас і вихідними даними, і результатами обчислень. Інформація накопичується і зберігається обчислювальною технікою, як правило, в двійковому цифровому вигляді, а отже знання операційної системи є ключем до професійного застосування комп’ютерних систем і мереж.

ОС UNIX є прикладом виключно вдалої реалізації мультипрограмної багато-користувальницької багатопоточної операційної системи (ОС). Свого часу вона проектувалася як інструментальна система для створення програмного забезпечення. Система UNIX має просту і прозору, але вельми потужну командну мову (інтерфейс командного рядка) та незалежну від пристроїв файлову систему. В зв’язку з тим, що при її створенні було використано мову програмування C, сама система та створені для неї програмні застосування виявилися мобільними (портабельними) стосовно інших апаратних платформ.

Першою метою при створенні цієї системи була мета збереження простоти і використання мінімальної кількості функцій. Всі реальні складності покладалися на користувальницькі програми. Другою метою була загальність. Одні і ті ж методи та механізми використовуються у багатьох випадках. Третя мета полягала у створенні операційної системи, в якій великі задачі можна було б розв’язувати комбінацією існуючих невеликих програм, а не створювати програми з початку.

ОС UNIX добре стандартизована і започаткувала утворення класу відкритих (POSIX-сумісних) систем. На цей день до цього класу належать такі операційні системи як AIX, HP-UX, SOLARIS, BSD, SCO UNIX, LINUX тощо.

3. Метою навчальної дисципліни «Програмування для UNIX-систем» є оволодіння студентами 2-го курсу факультету радіофізики, електроніки та комп'ютерних систем спеціальності «123 – Комп'ютерна інженерія» знаннями з основних концепцій створення та використання POSIX-сумісних операційних середовищ для відкритих комп'ютерних систем і формуванні світогляду студентів у галузі ОС, навичок роботи із ними у подальшому навчанні. Курс передбачає оволодіння студентами методикою корпоративних застосувань ОС у подальшій професійній діяльності після закінчення університетського курсу за спеціальністю «123 – Комп'ютерна інженерія».

Основними завданнями (навчальними цілями) є:

- 1) надати основні відомості та концепції з курсу «Програмування для UNIX-систем», які складають важливу частину інженерної підготовки студента-бакалавра за спеціальністю «Комп'ютерна інженерія».
- 2) узагальнити відомі поняття курсів «Програмування», «Дискретна математика», «Програмні та апаратні платформи» тощо;
- 3) простежити взаємозв'язок та розширити світогляд студента для широкого кола об'єктів досліджень;
- 4) продемонструвати застосування теоретичних відомостей до розв'язання практичних задач;
- 5) застосування знань, умінь, навичок і комунікацій у професійній діяльності, розвиток логічного та аналітичного мислення студентів;
- 6) прищепити вміння розв'язувати прикладні задачі методами сценарного програмування, яке розглядається в курсі «Програмування для UNIX-систем».

4. Результати навчання. У результаті вивчення дисципліни «Програмування для UNIX-систем» студент отримає підготовку, необхідну для подальшого навчання за освітньою програмою «Інженерія комп'ютерних систем і мереж», самостійного вивчення необхідної наукової та технічної літератури, створення програмних сценаріїв, рішення типових задач, що виникають при розробці програмного забезпечення комп'ютерних систем.

5. Передумови для вивчення навчальної дисципліни:

До вивчення дисципліни «Програмування для UNIX-систем» необхідно пройти підготовку і скласти іспити/заліки з таких дисциплін:

- Дискретна математика
- Програмування
- Основи апаратного та програмного забезпечення ЕОМ.

6а. Засоби оцінювання результатів навчання:

- **Семестрове оцінювання:** контроль здійснюється у такий спосіб. Навчальна дисципліна «Програмування для UNIX-систем» має два змістові модулі: у

- змістовий модуль 1 (ЗМ1) входять теми 1-8, у змістовий модуль 2 (ЗМ2) входять теми 9-15. Після завершення вивчення першого модуля студенти пишуть контрольну роботу №1, після завершення другого – пишуть контрольну роботу №2.
- **Контрольні роботи:** містять по одному теоретичному питанню і по 2 задачі. Для визначення рівня досягнення результатів навчання завдання в білетах контрольної роботи перевіряють уміння розв'язувати конкретні задачі відповідних розділів курсу (взаємодія процесів, написання регулярних виразів та сценаріїв тощо). Максимальна кількість балів за кожну контрольну роботу складає **15**.
 - **Виконання лабораторного практикуму:** в кінці семестру студенти одержують бали за уміння і навички роботи із ОС UNIX, її складовими компонентами та написання і налагоджування сценаріїв мовами оболонки **bash** та **Perl**. Максимальна кількість балів за лабораторний практикум – **30**. У такий спосіб студент може одержати за семестр максимально **60** балів (15+15+30). Обов'язковим для допуску до заліку є мінімальна кількість балів – **30**.
 - **Підсумкове оцінювання (у формі заліку):** форма заліку – письмово-усна з обов'язковою роботою на комп'ютері. На залік студент отримує білет, який складається із 2 питань: 1 теоретичне і 1 практичне (створення сценарію, конвеєра, фільтра тощо). Кожне питання оцінюється від 0 до 13 балів, а задача, – від 0 до 14 балів. Всього за залік можна отримати від **0** до **40** балів. Умовою досягнення позитивної оцінки («зараховано») за дисципліну є отримання не менше ніж **60** балів, при цьому оцінка за залік не може бути меншою **20** балів.

бб. Умови допуску до підсумкового іспиту:

умовою допуску до іспиту є мінімальна кількість балів не менше, аніж *критично-розрахунковий мінімум 30 балів* за семестр. Студенти, які протягом семестру набрали сумарно меншу кількість балів, ніж критично-розрахунковий мінімум 30 балів, для одержання допуску до іспиту обов'язково повинні розв'язати додаткові задачі для отримання необхідної порогової кількості балів.

бв. Критерії оцінювання

Критерієм успішного проходження здобувачем освіти підсумкового оцінювання є досягнення ним мінімальних порогових рівнів умінь (рішення задач) і знань (опанування теоретичного матеріалу).

	Знання (мінімум)	Знання (максимум)	Уміння (мінімум)	Уміння (максимум)	Знання + уміння (мінімум)	Знання + уміння (максимум)
Результати семестру	15	30	15	30	30	60
Іспит	15	20	15	20	30	40
Підсумкова оцінка	30	50	30	50	min 60	max 100

бг. Загальні результати і шкала відповідності оцінок

Оцінка (за національною шкалою) /National grade	Рівень досягнень, % /Marks, %
Відмінно / Excellent	90-100%
Добре / Good	75-89%
Задовільно / Satisfactory	60-74%
Незадовільно / Failed	0 -59%

7.1. Програма навчальної дисципліни «Програмування для UNIX-систем»

№ п/п	Назва теми	У тому числі			
		Лекції	Семінарські заняття	Лабораторні заняття	Самостійна робота
Змістовий модуль 1. Вступ до операційної системи UNIX					
1	Концепція відкритої системи стосовно операційних систем. Основні поняття операційних систем. Архітектура операційних систем. Концепції процесів та потоків і їх призначення	2	-	-	5
2	Відкриті системи та відкриті специфікації, призначення концепції відкритих систем. Ресурси операційних систем, підходи до їх класифікації. Сервіси ядра операційної системи 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. Функції API для створення процесів та потоків	2	-	-	5
4	Ресурси процесів та їх компоненти. Первинний потік, елементи потоків. Пріоритетна мультизадачність. 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 концепції системи UNIX. Синтаксис команди. Реєстрація користувача в системі та припинення сеансу зв'язку	2	-	2	5
6	Основні команди взаємодії користувачів в системі: mail, mailx, write, mesg, news . Команди date, echo, banner, passwd, lp, touch . Online-посібник з команд системи	2	-	2	5
7	Деревоподібна структура файлової системи UNIX, найбільш важливі каталоги: /bin, /usr, /usr/bin, /usr/lib, /usr/mail, /usr/man, /usr/local/bin, /usr/contrib/bin, /etc, /users, /dev . Абсолютний та відносний шляхи, навігація у файловій системі. Монтування та демонтування логічних пристроїв. Команди pwd, ls, cd, mkdir, rmdir	2	-	2	5
8	Файли та права доступу (повноваження) до них в операційній системі UNIX. Повноваження груп користувачів. Команди зміни повноважень.	2	-	2	5
Всього		16	-	8	40

Змістовий модуль 2. Створення сценаріїв у операційній системі UNIX					
9	Призначення оболонки shell та shell-змінні, встановлення значень та звернення до них. Области даних shell-змінних: локальна та оточення. Команди роботи з областями даних: set, unset, env, export . Передачі значень shell-змінних між локальною областю та оточенням	2	-	-	5
10	Аналіз різних інтерпретаторів команд shell в операційній системі UNIX, еволюція їх розвитку та сумісність. Ретроспективний аналіз інтерпретаторів команд shell: sh, csh, tcsh, ksh, bsh, bash, POSIX shell	2	-	-	5
11	Програмування мовою shell, сценарії та методи їх активізації. Спеціальні змінні shell # \$ @ та * . Спеціальні команди для обробки сценаріїв: shift, read, expr . Інтерактивні команди shell. Файли .profile	2	-	2	5
12	Галуження в сценаріях мовою shell. Коди завершення виконання команд. Команда test , перевірки файлів, рядків та чисельних виразів. Керуючі конструкції для розгалуження обчислень if та case . Керуючі конструкції для організації циклічних обчислень while, until, for , та foreach , приклади конструкцій. Команди shell-сценаріїв break, continue та exit	2	-	2	5
13	Застосування концепції сигналів та пасток (traps) як засобів міжпроцесної взаємодії при створенні сценаріїв. Команди kill та trap , екранування сигналів. Методика створення сценаріїв та методи їх налагоджування	2	-	-	5
14	Мова створення сценаріїв Perl . Призначення та концептуальні особливості мови. Типи даних (скаляри, масиви скалярів та асоційовані масиви скалярів (хеші)), вирази та операції. Списки та структури списків. Вирази та регулярні вирази, динамічні регулярні вирази, операції =~ та !~ . Основні керуючі конструкції мови Perl : циклічні for, foreach, while, until та допоміжні last, next, redo ; розгалуження if { }, unless { } . Зарезервовані змінні. Підпрограми, пакети, класи та модулі. Об'єктна орієнтація мови. Об'єкти, класи, методи.	2	-	2	5
15	Python як повнофункціональна мова сценаріїв UNIX. Застосування Python системними адміністраторами. Управління користувачами, дисковим простором, процесами, потоками, пристроями, журналюванням, резервним копіюванням тощо. Повнофункціональна мова сценаріїв (як Python) спрощує рішення цих задач. Приклади демонструють ці можливості.	2	-	2	5
Всього		14	-	8	35

Загальний обсяг **180 год.**, в тому числі:

Лекцій – **30 год.**

Лабораторні роботи – **14 год.**

Самостійна робота – **102 год.**

7.2 Самостійна робота студентів (СРС).

	Тема СРС	Джерело інформації
1.	Вивчення теорії процесів, потоків, обчислень	[1, 2] §1.1 – 1.4
2.	Розв'язання задач із теорії процесів, потоків, обчислень	[1, 2] §1.5
3.	Вивчення редакторів UNIX	[1, 2] §2.1 – 2.11
4.	Надбання навичок роботи із редакторами UNIX	[1, 2] §2.12
5.	Вивчення команд UNIX	[1, 2] §3.1 – 3.6
6.	Формування конвеєрів та фільтрів UNIX	[1, 2] §3.7
7.	Вивчення програмування мовою сценаріїв bash	[1, 2] §§4.1 – 4.6
8.	Написання сценаріїв мовою bash	[1, 2] § 4.7
9.	Вивчення програмування мовою Perl	[1, 2] §5.1 – 5.4
10.	Написання програм і сценаріїв UNIX мовою Python	[1, 2, 12] §5.5

Примітка: всі питання СРС включаються до питань на залік

7.3. Список джерел до тем самостійної роботи студентів:

[1] Погорілий С.Д. Програмне конструювання. Підручник за редакцією академіка АПН України Третяка О.В., видання 2-е. Київ : ВПЦ "Київський університет", Київ, 2007.

[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.

8. Рекомендована література:

[1] Б. Керниган Р. Пайк. UNIX универсальная среда программирования. М.: Финансы и статистика. 1992.

- [2] Погорілий С.Д. Програмне конструювання. Підручник за редакцією академіка АПН України Третяка О.В., видання 2-е. Київ : ВПЦ "Київський університет", Київ, 2007.
- [3] Погорілий С.Д., Калита Д.М. Комп'ютерні мережі. Апаратні засоби та протоколи передачі даних. За редакцією академіка АПН України Третяка О.В. Київ : ВПЦ "Київський університет", Київ, 2007.
- [4] Погорілий С.Д. "Автоматизація наукових досліджень. Основоположні математичні відомості. Програмне забезпечення" за редакцією академіка АПН України Третяка О. В. Київ: ВПЦ "Київський університет", 2002.
- [5] 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.
- [6] <http://www.citforum>
- [7] <http://www.infocity.com.ua>
- [8] Гордеев А.В., Молчанов А.Ю. Системное программное обеспечение. Санкт Петербург: Питер, 2001. – 734 с.
- [9] Блискавицкий А.А., Кабаев С.В. Операционные системы реального времени (обзор). Мир компьютерной автоматизации, 1995 N 1. – с. 31-38.
- [10] 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.
- [11] Хэвиленд К., Грэй Д., Салама Б. Системное программирование в UNIX. М.: Узд-во ДМК, 2000.
- [12] 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.

ДОДАТКИ:

9. Рекомендації щодо оцінки типів навчальної роботи

		Знання	Уміння	Комунікаційність	Автономність (ініціативність) і відповідальність
1	Лекції	100			
2	Лабораторні роботи	50	50		

10а. Результати навчання за дисципліною:

Код	Результат навчання	Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
1	студент повинен знати:			
1.1	Основоположні концепції і відомості із ОС UNIX; основні команди та програмні складові, в т.ч. awk. Знати можливості оболонки POSIX SHELL	лекційні заняття	залік, контрольні роботи, завдання для самостійної роботи	до 50
1.1 а	Основоположні відомості із програмування мовою оболонки bash	лекційні заняття	залік, контрольні роботи, завдання для самостійної роботи	
1.1 б	Основоположні відомості із програмування мовою Perl	лекційні заняття	залік, контрольні роботи, завдання для самостійної роботи	
2	студент повинен вміти:			
2.1	Використовувати основні команди ОС UNIX; створювати конвеєри, фільтри. Вільно працювати із текстовими редакторами	лабораторні роботи	контрольні роботи, завдання для самостійної роботи	до 50
2.1 а	Створювати сценарії мовою оболонки bash і мати навички їх налагоджування засобами ОС UNIX	лабораторні роботи	контрольні роботи, завдання для самостійної роботи	
2.1 б	Створювати сценарії мовми Perl і Python, мати навички їх налагоджування засобами ОС UNIX	лабораторні роботи	контрольні роботи, завдання для самостійної роботи	

10.6 Співвідношення результатів навчання дисципліни із програмними результатами навчання

	Знання (код 1)			Уміння (код 2)		
	1.1	1.1a	1.1б	2.1	2.1a	2.1б
Програмні результати навчання (назва)						
Знати основоположні концепції відкритих комп'ютерних систем, в т.ч. ОС UNIX; володіти основоположним набором команд ОС	+					
А) Основні методи і принципи створення сценаріїв мовою оболонки bash		+				
Б) Основні методи і принципи створення сценаріїв інтерпретованою мовою Perl			+			
Розв'язання задач при проектуванні та створенні програмних платформ				+		
А) Створювати сценарії мовою оболонки bash і мати навички їх налагоджування засобами ОС UNIX					+	
Б) Створювати сценарії мовою Perl і мати навички їх налагоджування засобами ОС UNIX						+

11. Розробник: Погорілий Сергій Дем'янович, доктор технічних наук, професор, завідувач кафедри комп'ютерної інженерії