

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

ЗАТВЕРДЖУЮ

Декан математичного факультету

_____ Гоменюк С.І.
(підпис) (прізвище, ініціали)
“ ____ ” _____ 2019 р.

**СИСТЕМНЕ ПРОГРАМУВАННЯ ТА АРХІТЕКТУРА
КОМП'ЮТЕРІВ**

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

підготовки бакалаврів
спеціальності 126 – “Інформаційні системи та технології”
освітньо-професійної програми “Інформаційні системи та технології”

Укладач: д.т.н., професор, професор кафедри програмної інженерії Гоменюк С. І.

Обговорено та ухвалено
на засіданні кафедри програмної інженерії

Ухвалено науково-методичною радою
математичного факультету

Протокол № 1 від “20” серпня 2019 р.
Завідувач кафедри _____ Лісняк А. О.

Протокол № 1 від “30” серпня 2019 р.
Голова науково-методичної
ради математичного
факультету _____ Пшенична О. С.

2019 рік

1. Опис навчальної дисципліни

| Найменування показників | Галузь знань, напрямок підготовки/ спеціальність, рівень вищої освіти (освітньо- кваліфікаційний рівень) | Характеристика навчальної дисципліни | |
|--|--|---|--------------------------|
| | | денна форма навчання | заочна форма навчання |
| Кількість кредитів – 5 | Галузь знань 12 – «Інформаційні технології» | Нормативна | |
| | | Цикл професійної підготовки | |
| Розділів – 2 | Спеціальність: 126 – «Інформаційні системи та технології» | Рік підготовки: | |
| Загальна кількість годин – 150 | | 2-й | – |
| | | Лекції | |
| Тижневих годин для денної форми навчання: аудиторних – 5 самостійної роботи студента – 6 | Освітньо-професійна програма «Інформаційні системи та технології» | 28 год. | – |
| | | Лабораторні роботи | |
| | 42 год. | – | |
| | Самостійна робота | | |
| | 80 год. | – | |
| | Рівень вищої освіти: бакалаврський | Вид контролю: екзамен | |

2. Мета та завдання навчальної дисципліни

Метою курсу є надання знань студентам спеціальності «Інформаційні системи та технології» про основні прийоми, засоби та підходи системного програмування в середовищі сімейства Unix-подібних операційних систем (ОС) Linux.

Завдання курсу:

- ознайомити студентів з основними стандартами (API) системного програмування в операційній системі Linux – POSIX, SUS і LSB;
- навчити використовувати мову програмування C для розробки системного програмного забезпечення в ОС Linux;
- навчити застосовувати на практиці низькорівневі та високорівневі API роботи з файлами;
- навчити основним прийомам і підходам до управління процесами в ОС Linux;
- навчити розробляти багатопотокові програми тощо.

У результаті вивчення дисципліни студент повинен знати:

- 1) основні поняття про системне програмування;
- 2) основи мови системного програмування C;

3) основи API POSIX;

4) базові поняття програмування в ОС Linux (файли та файлові системи, процеси, користувачі й групи, права доступу, сигнали та міжпроцесна взаємодія, обробка помилок тощо);

5) базові низькорівневі та багаторівневі функції для роботи файлами в ОС Linux;

6) основні поняття та принципи організації багатозадачності в ОС Linux;

7) основи розробки багатопотокових програм;

8) основи організації сучасних комп'ютерів (поняття архітектури комп'ютера та їх типи, елементна база тощо);

уміти:

1) виконувати компіляцію програм в ОС Linux в тому числі і із застосуванням автоматичної побудови програм із застосуванням утиліти make;

2) розробляти програми на мові C для буферизованого вводу-виводу файлів;

3) виконувати обробку файлів із застосуванням низькорівневих функцій ОС Linux;

4) виконувати базові операції над каталогами;

5) розробляти програми з використанням динамічного виділення пам'яті;

6) виконувати базові операції по управлінню процесами в Linux;

7) розробляти багатопотокові програми.

Згідно з вимогами освітньо-професійної програми студенти повинні досягти таких **компетентностей**:

– здатність застосовувати вибрану програмну платформу для написання програмного коду;

– здатність розв'язувати складні спеціалізовані задачі та проблеми, що виникають при практичному застосуванні інформаційних технологій у професійній діяльності або в процесі навчання, які, насамперед, передбачають створення системного програмного забезпечення;

– здатність вести дослідницьку діяльність, включаючи аналіз проблем, вибір способу і методів дослідження, а також оцінку якості результатів;

– здатність використовувати мову програмування C для створення прикладного та системного програмного забезпечення;

– здатність аргументувати вибір методів розв'язання спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення;

– здатність вирішувати проблеми у галузі комп'ютерних та інформаційних технологій, визначати обмеження цих технологій.

3. Міждисциплінарні зв'язки

Для розуміння змісту курсу студент повинен мати знання основ наступних дисциплін:

1. Основи програмування.

2. Дискретна математика.
3. Алгоритми та структури даних.
4. Операційні системи.

У свою чергу, теоретичні відомості та практичні навички, що надаються у курсі “Системне програмування та архітектура комп’ютерів” використовуються при вивченні спецкурсів, що викладаються для спеціальності “Інформаційні системи та технології” згідно навчального плану, виконанні курсових, дипломних та магістерських робіт.

4. Програма навчальної дисципліни

Розділ 1. Основні поняття системного програмування. Мова програмування C. Стандарти POSIX, SUS та LSB. Низькорівневі та високорівневі API роботи з файлами.

Тема 1. Основи системного програмування.

Вступ. Історія створення обчислювальної техніки. Прикладне і системне програмне забезпечення. Мови програмування, що застосовуються для створення системного програмного забезпечення.

Тема 2. Основи мови системного програмування C.

Історія створення мови програмування C. Стандарти C. Типи даних, змінні, покажчики й константи в C. Структура програми на мові C. Передача параметрів в функції. Базові операції й оператори мови C. Реалізація операцій введення і виведення в C. Препроцесор. Стандартна бібліотека.

Тема 3. Основні концепції системного програмування в Linux.

Поняття API. Стандарти системного програмування в Linux POSIX, SUS та LSB. Концепції програмування в Linux. Файли й файлова система. Процеси. Користувачі та групи. Права доступу. Сигнали та міжпроцесна взаємодія. Обробка помилок.

Тема 4. Низькорівневий файловий ввід-вивід в Linux.

Базові низькорівневі операції роботи з файлами. Синхронізація введення-виведення. Пряме позиціонування в файлі. Усічення файлів.

Тема 5. Високорівневий ввід-вивід в Linux.

Поняття буферизації вводу-виводу. Стандартний ввід-вивід. Відкриття потоку даних за допомогою файлового дескриптора. Позиціонування в потоку даних. Скидання потоків даних. Управління буферизацією. Векторний ввід-вивід в Linux. Відображення файлів у пам’ять.

Розділ 2. Багатозадачність в Linux. Потоки та багатопотокові процеси. Управління пам’яттю. Сигнали. Архітектура комп’ютерів.

Тема 6. Багатозадачність в Linux.

Базові поняття багатозадачності в Linux. Програми, процеси та потоки. Ієрархія та атрибути процесів. Запуск нового процесу. Поняття зомбі. Демони. Організація

багатозадачності та поняття планування роботи процесів. Пріоритети процесів. Прив'язка процесу до процесору. Системи реального часу.

Тема 7. Багатопотоковість в Linux.

Поняття потоку. Однопотоківі та багатопотокові процеси. Переваги та недоліки багатопотокових програм. Моделі багатопотоковості. Конкурентність, паралелізм і гонки. P-потоки.

Тема 8. Управління пам'яттю.

Адресний простір процесу. Виділення динамічної пам'яті. Управління сегментом даних. Розширене виділення пам'яті. Управління пам'яттю. Блокування пам'яті.

Тема 9. Сигнали.

Концепція сигналів. Основи управління сигналами. Відправлення сигналів. Блокування сигналів. Розширене управління сигналами.

Тема 10. Архітектура комп'ютера.

Рівні комп'ютерної організації. Поняття та історія розвитку архітектури комп'ютерів. Типи та сімейства комп'ютерів. Процесор. Пам'ять. Пристрої вводу та виводу.

5. Структура навчальної дисципліни

| Назви розділів і тем | Кількість годин | | | | | | | | | | |
|--|-----------------|--------------|----------|-----------|-----------|----------|--------------|--------------|-----|------|-----|
| | денна форма | | | | | | заочна форма | | | | |
| | усього | у тому числі | | | | | усього | у тому числі | | | |
| л | | п | лаб | с.р. | інд | л | | п | лаб | с.р. | інд |
| Розділ 1. Основні поняття системного програмування. Мова програмування C. Стандарти POSIX, SUS та LSB. Низькорівневі та високорівневі API роботи з файлами. | | | | | | | | | | | |
| Тема 1. Основи системного програмування | 2 | 2 | 0 | 0 | 0 | 0 | | | | | |
| Тема 2. Основи мови системного програмування C | 29 | 6 | 0 | 8 | 15 | 0 | | | | | |
| Тема 3. Основні концепції системного програмування в Linux. | 9 | 2 | 0 | 2 | 5 | 0 | | | | | |
| Тема 4. Низькорівневий файловий ввід-вивід в Linux. | 21 | 2 | 0 | 4 | 15 | 0 | | | | | |
| Тема 5. Високорівневий ввід-вивід в Linux. | 23 | 2 | 0 | 6 | 15 | 0 | | | | | |
| Разом за розділом 1 | 84 | 14 | 0 | 20 | 50 | 0 | | | | | |
| Розділ 2. Багатозадачність в Linux. Потоків та багатопотокові процеси. Управління пам'яттю. Сигнали. Архітектура комп'ютерів. | | | | | | | | | | | |
| Тема 6. Багатозадачність в Linux. | 22 | 4 | 0 | 8 | 10 | 0 | | | | | |
| Тема 7. Багатопотоковість в Linux | 18 | 2 | 0 | 6 | 10 | 0 | | | | | |
| Тема 8. Управління пам'яттю. | 16 | 2 | 0 | 4 | 10 | 0 | | | | | |
| Тема 9. Сигнали. | 6 | 2 | 0 | 4 | 0 | 0 | | | | | |
| Тема 10. Архітектура комп'ютера. | 4 | 4 | 0 | 0 | 0 | 0 | | | | | |
| Разом за розділом 2 | 66 | 14 | 0 | 22 | 30 | 0 | | | | | |
| Усього годин | 150 | 28 | 0 | 42 | 80 | 0 | | | | | |

6. Теми лекційних занять

| № теми | Назва | Кількість годин |
|--------------|--|-----------------|
| 1 | Основи системного програмування. Вступ. Історія створення обчислювальної техніки. Прикладне і системне програмне забезпечення. Мови програмування, що застосовуються для створення системного програмного забезпечення. | 2 |
| 2 | Основи мови системного програмування C. Історія створення мови програмування C. Стандарти C. Типи даних, змінні, покажчики й константи в C. Структура програми на мові C. Передача параметрів в функції. Базові операції й оператори мови C. Реалізація операцій введення і виведення в C. Препроцесор. Стандартна бібліотека. | 6 |
| 3 | Основні концепції системного програмування в Linux. Поняття API. Стандарти системного програмування в Linux POSIX, SUS та LSB. Концепції програмування в Linux. Файли й файлова система. Процеси. Користувачі та групи. Права доступу. Сигнали та міжпроцесна взаємодія. Обробка помилок. | 2 |
| 4 | Низькорівневий файловий ввід-вивід в Linux. Базові низькорівневі операції роботи з файлами. Синхронізація введення-виведення. Пряме позиціонування в файлі. Усічення файлів. | 2 |
| 5 | Високорівневий ввід-вивід в Linux. Поняття буферизації вводу-виводу. Стандартний ввід-вивід. Відкриття потоку даних за допомогою файлового дескриптора. Позиціонування в потоку даних. Скидання потоків даних. Управління буферизацією. Векторний ввід-вивід в Linux. Відображення файлів у пам'ять. | 2 |
| 6 | Багатозадачність в Linux. Базові поняття багатозадачності в Linux. Програми, процеси та потоки. Ієрархія та атрибути процесів. Запуск нового процесу. Поняття зомбі. Демони. Організація багатозадачності та поняття планування роботи процесів. Пріоритети процесів. Прив'язка процесу до процесору. Системи реального часу. | 4 |
| 7 | Багатопотоковість в Linux. Поняття потоку. Однопотоківі та багатопотокові процеси. Переваги та недоліки багатопотокових програм. Моделі багатопотоковості. Конкурентність, паралелізм і гонки. P-потоки. | 2 |
| 8 | Управління пам'яттю. Адресний простір процесу. Виділення динамічної пам'яті. Управління сегментом даних. Розширене виділення пам'яті. Управління пам'яттю. Блокування пам'яті. | 2 |
| 9 | Сигнали. Концепція сигналів. Основи управління сигналами. Відправлення сигналів. Блокування сигналів. Розширене управління сигналами. | 2 |
| 10 | Архітектура комп'ютера. Рівні комп'ютерної організації. Поняття та історія розвитку архітектури комп'ютерів. Типи та сімейства комп'ютерів. Процесор. Пам'ять. Пристрої вводу та виводу. | 4 |
| Разом | | 28 |

7. Теми лабораторних занять

| № теми | Назва | Кількість годин |
|--------------|---|-----------------|
| 2 | Створення програм на мові С в операційній системі Linux. Редагування вихідного коду програм. Компіляція та запуск програмного забезпечення в Linux. Автоматичне збирання програм. | 4 |
| 2 | Робота з файлами в С засобами стандартної бібліотеки. Базові функції для роботи з файлами в С. Файловий ввід-вивід. Допоміжні функції для роботи з файлами. | 4 |
| 3 | Робота з каталогами. Типи файлів в Linux. Базові функції роботи з каталогами. | 4 |
| 4 | Низькорівневий файловий ввід-вивід в Linux. Концепція низькорівневого введення-виведення в Linux. Базові та допоміжні функції низькорівневого вводу-виводу | 4 |
| 5 | Буферизований ввід-вивід в Linux. Буферизоване зчитування з потоку даних (запис у потік даних). Управління буферизацією. | 2 |
| 5 | Розширені можливості вводу-виводу в Linux. Векторний ввід-вивід в Linux. | 1 |
| 5 | Відображення файлів у пам'ять. Базові поняття відображення файлів у пам'ять. Синхронізація файлів | 1 |
| 6 | Основні поняття багатозадачності в Linux. Базові функції створення нового процесу в Linux. | 8 |
| 7 | Створення багатопотокових додатків в Linux. Загальне поняття потоку виконання. Базові функції для роботи з потоками в Linux. | 6 |
| 8 | Управління пам'яттю. Виділення динамічної пам'яті та керування нею. Блокування пам'яті. | 4 |
| 9 | Основи міжпроцесної взаємодії в Linux. Загальне поняття міжпроцесної взаємодії. Способи організації міжпроцесної взаємодії. Використання сигналів для взаємодії між процесами. | 2 |
| 9 | Міжпроцесна взаємодія з використанням спільної пам'яті. Організація взаємодії між процесами із застосуванням загальної пам'яті. | 2 |
| Разом | | 42 |

8. Самостійна робота

| № теми | Назва | Кількість годин |
|--------------|-----------------------------------|-----------------|
| 2 | Стандарти мови програмування С | 20 |
| 3 | Управління файлами й каталогами | 30 |
| 3 | Запуск процесів в Linux | 10 |
| 5 | Створення багатопотокових програм | 10 |
| 6 | Робота з пам'яттю | 10 |
| Разом | | 80 |

9. Види контролю і система накопичення балів

Методи контролю

Кожна лабораторна робота складається з теоретичної і практичної частини. Теоретична частина захищається студентом шляхом бесіди з викладачем. Практична частина перевіряється викладачем на предмет якості написання програмного коду та відповідності результатів роботи програми заявленим задачам.

Розподіл балів, які отримують студенти

| Розділ | Вид контролю | Кількість балів |
|--------------------------|-------------------------|-----------------|
| Розділ 1 | Лабораторна робота № 1 | 6 |
| | Лабораторна робота № 2 | 6 |
| | Лабораторна робота № 3 | 6 |
| | Лабораторна робота № 4 | 6 |
| | Лабораторна робота № 5 | 6 |
| Разом | | 30 |
| Розділ 2 | Лабораторна робота № 6 | 6 |
| | Лабораторна робота № 7 | 6 |
| | Лабораторна робота № 8 | 6 |
| | Лабораторна робота № 9 | 6 |
| | Лабораторна робота № 10 | 6 |
| Разом | | 60 |
| Екзамен | | 40 |
| Всього за семестр | | 100 |

Критерії оцінювання кожного з проведених видів контролю:

1. Захист лабораторної роботи відбувається після виконання завдання та завантаження відповідного звіту до системи Moodle. Один завантажений звіт без захисту лабораторної роботи може бути оцінений не вище, ніж 50% від максимального можливого значення балів. Під час захисту студент має відповісти на питання викладача та пояснити деякі етапи виконання завдання.

2. Екзамен проводиться у формі тестування рівня теоретичної підготовки з усіх розділів курсу та має 40 питань, які оцінюються по 1 балу.

3. Підсумкова оцінка складається з суми балів, отриманих студентом за виконання лабораторних робіт, та оцінки, отриманої на екзамені.

Шкала оцінювання: національна та ECTS

| За шкалою ECTS | За шкалою університету | За національною шкалою | |
|----------------|--|------------------------|---------------|
| | | Екзамен | Залік |
| A | 90 – 100 (відмінно) | 5 (відмінно) | Зараховано |
| B | 85 – 89 (дуже добре) | 4 (добре) | |
| C | 75 – 84 (добре) | | |
| D | 70 – 74 (задовільно) | 3 (задовільно) | |
| E | 60 – 69 (достатньо) | | |
| FX | 35 – 59 (незадовільно – з можливістю повторного складання) | 2 (незадовільно) | Не зараховано |
| F | 1 – 34 (незадовільно – з обов'язковим повторним курсом) | | |

10. Рекомендована література

Основна

1. Лав Р. Linux. Системное программирование. 2-е изд. – СПб. : Питер, 2014. – 448 с.: ил.
2. Керниган Б., Ритчи Д. Язык программирования С. – Москва : Вильямс, 2015. – 304 с.
3. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. – СПб.: Питер, 2013. – 816 с.: ил.
4. Прата С. Язык программирования С. Лекции и упражнения, 6-е изд. : Пер. с англ. – Москва : Вильямс, 2015. – 928 с. : ил.
5. Керриск М. Linux API. Исчерпывающее руководство. – СПб. : Питер, 2018. – 1248 с.: ил.

Додаткова:

6. Иванов Н. Н. Программирование в Linux. Самоучитель. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2012. – 400 с.: ил.
7. Чан Т. Системное программирование на C++ для UNIX: Пер. с англ. – Киев : Издательская группа ВНУ, 1997. – 592 с.
8. Системное программирование в современных операционных системах: Учебное пособие / Бондаренко М. Ф., Липанов А. В., Путятин Е. П., Синельникова Т. Ф. – Харьков: ООО «Компания СМИТ», 2005. – 432 с.
9. Колисниченко Д. Н. Разработка Linux-приложений. – СПб. : БХВ-Петербург, 2012. – 432 с.
10. Стахнов А. А. Linux: 3-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2009. – 1056 с: ил.

Погоджено _____
навчальний відділ
“ ____ ” _____ 2019 р.