

## АНОТАЦІЯ ДИСЦИПЛІНИ

«АРХІТЕКТУРА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ UNIX ПОДІБНИХ СИСТЕМ»

*Ступінь вищої освіти – бакалавр*

*Спеціальність – 121 Інженерія програмного забезпечення*

*Галузь знань – 12 Інформаційні технології*

### **I. Мета та зміст навчальної дисципліни**

Дисципліна "Архітектура та програмна реалізація UNIX подібних систем" є дуже важливою в телекомунікаційній галузі, оскільки вона допомагає функціональній сумісності різних технологій: комп'ютерних систем, засобів зв'язку та передачі сигналів, програмного забезпечення та мультимедійних систем.

**Мета дисципліни** — вивчення та практичне засвоєння методів роботи з багатозадачними операційними системами, на базі яких функціонує велика кількість комп'ютерів в мережі Інтернет, а також набуття студентами практичних вмінь і навичок створення програмних систем "клієнт-сервер".

**Предметом** є механізми функціонування UNIX-подібних операційних систем та порядок розробки програмного забезпечення "клієнт-сервер".

**Основні завдання вивчення дисципліни:** механізми функціонування UNIX-подібних операційних систем та порядок розробки програмного забезпечення "клієнт-сервер".

### **II. Перелік знань і умінь, яких набуде студент після опанування даної дисципліни:**

#### **Програмні результати навчання:**

- парадигм програмування, сучасних мов програмування, основних структур даних і алгоритмів;
- принципи побудови, склад, структуру та функції сучасних UNIX-подібних операційних систем;
- основні системні виклики (функції) ОС UNIX, які застосовуються при роботі з файлами (каталогами) та процесами (потокami);
- засоби міжпроцесної взаємодії, які реалізовані в сучасних UNIX-подібних операційних системах;
- засоби синхронізації взаємодіючих процесів (потоків), що виконуються паралельно;
- принципи та протоколи взаємодії основних мережних служб мережі Інтернет.
- використовувати довідкову літературу, технічну документацію;
- встановлювати, налаштовувати та обслуговувати системне та прикладне програмне забезпечення в системах автоматизованого проектування;
- застосовувати основні системні виклики (функції) ОС UNIX, які використовують при роботі з файлами (каталогами) та процесами (потокami);
- розробляти локальне та мережне програмне забезпечення, яке функціонує за технологією "клієнт-сервер".

#### **Перелік компетентностей, яких набуде студент після опанування даної дисципліни:**

Процес вивчення дисципліни спрямований на формування у студентів наступних системних і професійних компетенцій:

- здатність застосовувати професійні знання й уміння на практиці;
- здатність застосовувати сучасні парадигми програмування під час програмної реалізації професійних задач;

– знання загальних принципів організації та функціонування операційних систем, здатність розробляти елементи системного програмного забезпечення систем автоматизованого проектування).

### **III. Зміст дисципліни, що пропонується для вивчення студентами за модулями та темами (делать сот раздел вариативно по желанию, если будет время)**

**Тема 1. UNIX – класична багатозадачна операційна система.** Історія створення ОС UNIX, огляд можливостей, опис архітектури ядра. Системні виклики та бібліотечні функції, обробка помилок. Довідкова підсистема. Джерела стандартів, граничні значення системних змінних, функції `sysconf`, `rathconf` і `frathconf`. Елементарні системні типи даних.

Порядок реєстрації користувача в системі, ідентифікація користувача. Оболонка (shell) – інтерактивний обробник команд користувача. Огляд можливостей оболонки.

**Тема 2. Файлова підсистема.** Керування файловим введенням-виведенням. Огляд бібліотечних функцій для роботи з файлами. Дескриптори файлів, спільний доступ до файлів, атомарність операцій введення-виведення.

Системні операції для роботи з файловою системою. Файли та каталоги. Внутрішня організація файлів. Робота з індексами. Типи файлів. Права доступу до файлів. Огляд побудови файлових систем UNIX-подібних ОС. Жорсткі та символічні посилання на файли. Часові характеристики файлів. Робота з каталогами.

Стандартна бібліотека введення-виведення. Потоки та об'єкти типу FILE. Стандартні потоки введення-виведення. Відкриття та закриття потоку, читання з потоку та запис у потік, використання буферизації потоків для пришвидшення операцій введення-виведення. Тимчасові файли. Введення-виведення згідно з форматом. Отримання та встановлення системної дати і часу.

**Тема 3. Підсистема керування процесами.** Середовище виконання процесу. Запуск та завершення роботи процесу. Аргументи командного рядка. Функції виділення та звільнення пам'яті. Список змінних середовища. Отримання та встановлення обмежень на використання ресурсів в рамках процесу.

Керування процесами. Функції створення та завершення роботи процесу. Ідентифікатори процесу. Виклик інших програм. Синхронізація продовження виконання процесу з моментом завершення роботи свого нащадка. Спільне використання файлів. Зміна ідентифікаторів користувача та групи. Ідентифікація користувача. Часові характеристики процесу.

Взаємовідносини між процесами. Вхід до системи з терміналу та мережний вхід. Групи процесів, сесії, керуючий термінал.

Сигнали. Концепція сигналів. Надсилання та обробка сигналів процесами. Надійні та ненадійні сигнали. Перервані системні виклики. Набори сигналів.

**Тема 4. Взаємодія процесів на локальній машині.** Огляд механізмів взаємодії процесів в ОС UNIX. Іменовані та неіменовані канали. Процеси, потоки та загальний доступ до інформації. Живучість об'єктів IPC. Дія команд `fork`, `exec` і `exit` на об'єкти IPC. Неіменовані канали (pipe), функції `open` і `close`. Іменовані канали (FIFO). Деякі властивості каналів. Послідовні та паралельні сервери. Обмеження при використанні каналів.

Програмні потоки. Концепція потоків та їх ідентифікація. Створення потоку та завершення його роботи.

Керування програмними потоками. Засоби синхронізації дій потоків. Атрибути потоків. Синхронізація дій потоків за допомогою взаємних виключень, блокувань читання-запису та умовних змінних.

Засоби міжпроцесної взаємодії System V. Черги повідомлень. Ідентифікація об'єктів IPC за допомогою ключів. Права доступу до об'єктів IPC. Створення та відкриття каналів IPC. Програми `ipcs` та `ipcrm`. Обмеження ядра. Робота з чергами повідомлень: створення, відкриття, читання та запис повідомлень, керування. Мультиплексування повідомлень. Приклади програм "клієнт-сервер". Обмеження, які накладають на черги повідомлень.

Семафори System V – засіб синхронізації дій процесів. Робота з семафорами: створення, відкриття, оперування, керування. Обмеження, які накладають на семафори.

Поділювана пам'ять – найшвидший засіб міжпроцесної взаємодії. Робота з сегментами поділюваної пам'яті: створення, приєднання та від'єднання сегменту до адресного простору процесу, керування. Обмеження, які накладають на поділювану пам'ять.


**Тема 5. Взаємодія процесів у мережі.** Вступ до мережного програмування. Протоколи транспортного рівня TCP та UDP. Приклад простої програми “клієнт-сервер”. Модель OSI. Огляд протоколів сім'ї TCP/IP. Встановлення та розрив з'єднання TCP. Номери портів. Використання протоколів прикладними програмами в мережі Інтернет.

Структура адрес гнізд IPv4 та IPv6. Системні функції для роботи з гніздами TCP. Складові частини мережної адреси. Огляд функцій перетворення адрес. Паралельні сервери. Лічильник посилок дескриптора гнізда.

Процеси-демони. Демон syslogd. Суперсервер inetd (xinetd). Альтернативні варіанти побудови мережних серверів. Правила програмування демонів. Журналювання повідомлень. Запуск єдиного екземпляра демона. Угоди для демонів. Модель “клієнт-сервер”. Автоматизація процесу запуску мережних серверів. Послідовні та паралельні сервери. Огляд різних варіантів побудови паралельних серверів та оцінка їх продуктивності.

### **ЗАТВЕРДЖЕНО**

Гарант освітньо-професійної програми  
«Інженерія програмного забезпечення»  
першого(бакалаврського) рівня освіти:  
зав. каф. КТМ, д.т.н., проф.

  
Ніконов О.Я.