

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет ім. Івана Франка
Факультет прикладної математики та інформатики
Кафедра програмування

ПРОГРАМА КУРСУ

“ Програмування під UNIX-подібними системами ”

Напрямок : інформатика
 Факультет : прикладної математики та інформатики
 Форма навчання : денна

Виписка з навчального плану

Семестр	Кількість кредитів	Загальний обсяг (год.)	Всього год. (год.)	у тому числі (год.):			Самостійна роб. (год)	Контрольні (модульні) роботи	Курсові роботи (проекти)	Залік	Іспит
				Лекції	Лабораторні	Практичні					
8	4		70	0	42	28	74			+	-

1. АНОТАЦІЯ

Стосовно сімейства UNIX-подібних операційних систем вивчаються архітектура та особливості функціонування системи, а також командний та програмний інтерфейси. Розглядається керування системою за допомогою скриптових програм, а також основи розробки прикладних програм мовами C++ та java. Як приклад розглядається операційна система linux.

2. ЗМІСТ ПРОГРАМИ

1. Загальна характеристика ОС UNIX. Історія утворення ОС UNIX. Поняття про архітектуру системи. Основні поняття ОС UNIX: ядро, програма, процес, файлова система. Генеалогія версій та основні стандарти. ОС Linux. Особливості функціонування UNIX; адміністрування системи, категорії користувачів. Поняття про командний та програмний інтерфейси.

2. Командний інтерфейс користувача ОС UNIX

2.1. Робота з файловою системою

Загальна характеристика файлової системи. Види файлів. Ієрархічна структура файлової системи. Користувачі операційної системи; стандартні користувачі та групи. Концепція володіння файлами; права доступу до файлів. Командний інтерфейс для роботи з каталогами і файлами. Особливості інтерфейсу на різних платформах. Утворення текстових файлів та їх редагування. Текстові редактори `vi`, `emacs`. Поняття про інтерфейс з пристроями; спеціальні файли блочних та символьних пристроїв. Ідентифікація пристроїв; структура назви спеціальних файлів.

2.2. Процеси

Концепція процесу як способу функціонування ОС UNIX. Процеси і програми. Типи процесів. Взаємодія процесів з ядром; поняття про системні виклики. Особливості виконання процесу в режимі задачі та режимі ядра. Основні атрибути процесу; відображення їх в структурах даних процесу; таблиця процесів, контекст процесу. Простір адрес процесу; поняття про віртуальну пам'ять і свопінг.

Принципи планування процесів; пріоритети. Можливі стани процесу. Утворення нових процесів, системні виклики `fork()`, `exec()` і `wait()`. Інтерактивний та фоновий режими виконання процесів. Основні команди керування процесами. Сигнальний механізм і керування процесами; огляд множини сигналів. Особливості інтерфейсу системи керування процесами на різних платформах.

2.3. Використання командних інтерпретаторів

Інтерпретатор команд: функціональне призначення та особливості виконання. Види команд. Запуск команд на виконання; формати команд; генерування списку назв файлів. Скрипти. Особливості інтерпретаторів `Bourne-shell` і `c-shell`. Переадресування стандартних потоків введення/виведення даних. Поняття про фільтри, приклади широкоживаних фільтрів. Основні принципи побудови скриптів. Синтаксис мови інтерпретатора `c-shell`. Система керування завданнями.

3. API файлової системи

3.1. Внутрішня будова файлової системи

Основні принципи організації файлової системи `s5fs`, суперблок, індексні дескриптори. Особливості `FFS`. Основи віртуальної файлової системи, таблиці ядра для супроводження `VFS`. Структури даних каталогів. Жорсткі і символічні зв'язки файлів.

3.2. Файлові API

Загальні файлові API. API файлових атрибутів. Робота із записами файла паролів. Спільне використання файлів; поняття про синхронізацію. API каталогів. Робота з файлами пристроїв і іменованими каналами. Інтерфейс символічних посилань.

4. API системи керування процесами

4.1. Засоби ядра підтримки процесів

Особливості програмного інтерфейсу UNIX. Структура програми. Формати виконуваних файлів. Механізм утворення та завершення процесів; призупинення процесу. Множина системних викликів `exec()`. Особливості виклику інтерпретатора команд функцією `system()`. Утворення комунікаційних каналів; виклик `pipe()`. Механізм переадресування введення/виведення. API атрибутів процесу; зміна атрибутів процесу. Системні виклики стосовно групи і сеансу.

4.2. Сигнальний механізм

Способи генерування сигналів. Реакція процесу на сигнали. Маска сигналів. Виклик обробника сигналу. Особливості сигнального механізму System V і BSD. Спеціальні структури даних для отримання детальної інформації про сигнал. Особливості міжпроцесної взаємодії з використанням системного виклику kill(). API таймерів.

5. API міжпроцесного зв'язку (IPC)

5.1.Інтерфейси повідомлень.

Засоби підтримки повідомлень ядром; структури даних. Передача/прийом повідомлень. Доступ до черги повідомлень, маніпулювання параметрами черги повідомлень.

5.2.Семафори як засіб синхронізації.

Засоби підтримки семафорів ядром; структури даних. Відкриття та утворення набору семафорів. Запит і зміна значення семафора. Запит і зміна керуючих параметрів набору семафорів.

5.3.Розділювана пам'ять.

Утворення та відкриття розділюваної пам'яті. Під'єднання розділюваної пам'яті; читання та запис даних. Від'єднання та знищення розділюваної пам'яті. Запит і зміна керуючих параметрів розділюваної пам'яті.

6. Адміністрування ОС UNIX

6.1.Завантаження системи.

Роль процесу init в системі. Автоматичне та ручне завантаження. Огляд командних файлів завантаження системи. Особливості завантаження в різних системах. Команди перезавантаження і зупинки операційної системи.

6.2.Адміністрування користувачів.

Реєстрація користувачів. Формування бюджету користувача; встановлення початкового паролю і початкового каталогу. Редагування файла /etc/group. Від'єднання та вилучення реєстраційних імен.

6.3.Обслуговування файлової системи.

Монтування та демонтування файлової системи. Керування правами доступу до файлової системи. Перевірка і відновлення файлових систем. Резервне копіювання файлових систем: команди dump і restore. Команди архівування файлів.

6.4.Обслуговування пристроїв.

Драйвери пристроїв, способи іменування. Номери пристроїв, таблиці переходів. Додавання драйверів пристроїв у різних системах, завантажувані модулі. Конфігурування терміналів. Процедура інсталювання нового диску.

6.5.Конфігурування ядра.

Формування файла конфігурації. Побудова ядра, інсталювання ядра. Особливості конфігурування ядра в різних системах. Використання системи syslog для реєстрування системних подій. Засоби syslog, файл конфігурації.

7. Бібліотека QT

Основні частини бібліотеки. Сигнал-Слот, Події. Доступ до баз даних. Ввід/вивід та мережева взаємодія. Діалоги. Створення користувацького інтерфейсу. Потоки. Синхронізація. Класи для роботи з XML.

8. Java

8.1. Програмування GUI за допомогою пакету AWT

Стандартні компоненти та контейнери. Менеджери розміщення. Створення вікна програми за допомогою Frame. Діалогові вікна. Стандартні діалоги.

8.2. Використання Servlets

Створення сервлету. Контейнер сервлетів. Клієнти/Використання.

ОСНОВНА ЛІТЕРАТУРА

1. У. Ричард Стивенс, Стивен А. Раго. UNIX. Профессиональное программирование, 2-ое издание. Символ-Плюс. 2007.
2. Жасмин Бланшет, Марк Саммерфилд. Qt 4. Программирование GUI на C++. КУДИЦ-Пресс. 2007.
3. QT Documentation. <http://doc.qt.nokia.com/>
4. Эккель Б. Философия Java. Библиотека программиста. – СПб.: Питер, 2009. – 640 с.

Автори програми:

Літинський С.В., асистент кафедри програмування

Музичук А.О., канд.фіз.-мат.наук, доцент кафедри програмування