Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Кафедра автоматизації проектування енергетичних процесів і систем

АРХІТЕКТУРА СИСТЕМНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	**Перший (бакалаврський)**
Галузь знань	12 Інформаційні технології
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем і веб-технологій
Статус дисципліни	Нормативна
Форма навчання	очна(денна)
Рік підготовки, семестр	2 курс весняний семестр
Обсяг дисципліни	На засвоєння дисципліни передбачено 150 год / 5 кредитів ЄКТС, з них - 36 год. лекції, 36 год. практичні, 78 год. самостійна робота
Семестровий контроль/ контрольні заходи	Екзамен
Розклад занять	Науково-педагогічний працівник
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: д.т.н., доцент, Левченко Лариса Олексіївна, levchenko_larisa@lll.kpi.ua, тел. 097-068-11-42 Практичні: д.т.н., доцент, Левченко Лариса Олексіївна, levchenko_larisa@lll.kpi.ua, тел. 097-068-11-42
Розміщення курсу	Кампус

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчання та результати навчання

В цій дисципліні детально вивчається архітектура системного програмного забезпечення операційно системи Linux. На сьогоднішній день затребувані фахівці, які впроваджують новітні технології, в основі яких лежить саме Linux. Також розглядається і архітектура операційної системи Window 10, ця система є найбільш вживаною серед користувачів. Отримані знання дозволяють виконувати роботи з адміністрування операційної системи, застосування технології контейнерів для розгортання, поширення та функціонування програмного забезпечення, створення мікросервісів, а також як результат будуть сприяти кар'єрному зростанню, самореалізації, застосуванню отриманих знань для вирішення практичних завдань.

Метою дисципліни є опанування студентами теоретичних знань архітектури системного програмного забезпечення, побудови, функціонування, використання засобів операційних систем, технології контейнерів для реалізації упаковки, розгортання та функціонування програмного забезпечення.

Предмет дисципліни - вивчення принципів побудови, архітектури, основних функцій, режимів роботи, засобів операційних систем (ОС), розроблення мікросервісів на основі технології Docker.

Завдання. В результаті вивчення дисципліни у студентів повинні сформуватися наступні компетентності:

загальні:

- здатність до абстрактного мислення, аналізу та синтезу (ЗК 1),
- здатність спілкуватися іноземною мовою як усно, так і письмово (ЗК 4),
- здатність діяти соціально відповідально та свідомо (ЗК 10).

фахові:

- здатність розробляти архітектури, модулі та компоненти програмних систем (ФК 3),
- здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки (в тому числі кібербезпеки) (ФК 6),
- здатність накопичувати, обробляти та систематизувати професійні знання щодо створення тестування і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя (ФК 10),
- здатність обґрунтовано обирати та освоювати інструментарій з розробки тестування та супроводження програмного забезпечення (ФК 13),
- здатність проектувати кібер-фізичні системи, володіти скриптовими та декларативними мовами програмування (ФК 16),
- здатність розробляти програмні системи з мікросервісною архітектурою, конструювати мобільні додатки, крос- та мульти-платформне програмування, зокрема, для кібер-фізичних систем (ФК 20).

Після засвоєння навчальної дисципліни студенти мають продемонструвати такі програмні результати навчання:

- дотримуватися морально-етичних та культурних норм, принципів академічної доброчесності та кодексу професійної етики, примножувати досягнення суспільства (ПРН 1),
- знати професійні стандарти та інші нормативно- правові документи в галузі інженерії програмного забезпечення (ПРН 10),
- використовувати методологію створення програмного забезпечення згідно поставленої задачі. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення. Мотивовано обирати мови програмування та технології розробки для розв’язання завдань створення і супроводження програмного забезпечення (ПРН 16),
- володіти навичками командної розробки, погодження, оформлення і випуску всіх видів програмної документації (ПРН 17).

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Пререквізити дисципліни. Знання, отримані при вивченні дисциплін: «Об’єктно-орієнтований аналіз та конструювання програмних систем», «Бази даних», «Технології конструювання програмного забезпечення».

Постреквізити дисципліни. Отримані знання при вивченні дисципліни «Архітектура системного програмного забезпечення» формує базові знання для вивчення наступних дисциплін: «Основи розробки трансляторів», «Методології розробки інтелектуальних комп’ютерних програм», «Розробка програмного забезпечення мобільних пристроїв», які викладаються в наступних семестрах. Компетенції, отримані студентами в процесі вивчення цієї дисципліни, використовуються ними при виконанні дипломної роботи.

3. Зміст навчальної дисципліни

Розділ 1. Призначення, функції та архітектура операційної системи Linux.
Розділ 2. Управління Linux-пакетами
Розділ 3. Файлова система Linux.
Розділ 4. Управління процесами
Розділ 5. Управління пам’яттю
Розділ 6. Застосування технології контейнерів
Розділ 7. Операційні системи сімейства Windows

4. Навчальні матеріали та ресурси

Основна література

1. Немет Эви, Гарт Снайдер, Трент Хейн, Бэн Уэйли. Unix и Linux: руководство системного администратора. Москва : ООО “И.Д. Вильямс”. 2012. 1312 с.
2. Керриск М. Linux API. Исчерпывающее руководство. Санкт-Петербург : Питер, 2019. 1248 с.
3. Руссинович М., Соломон Д., Ионеску А., Йосифович П. Внутреннее устройство Windows. Санкт-Петербург : Питер. 2018. 944 с.
4. Моэт Э. Использование Docker. Москва : ДМК Пресс, 2017. 354 с.
5. Кочер П.С. Микросервисы и контейнеры Docker. Москва : ДМК Пресс. 2019. 240 с.
6. Милл И., Сейерс Э. Docker на практике. Москва : ДМК Пресс. 2020. 516 с.

Додаткова література

7. Уорд Б. Внутреннее устройство Linux. Санкт-Петербург : Питер. 2016. 384 с.
8. Негус К., Каэн Ф. Ubuntu и Debian Linux для продвинутых: более 1000 незаменимых команд. Санкт-Петербург : Питер. 2014. 384 с.
9. Волох С. Ubuntu Linux с нуля. Санкт-Петербург : БХВ-Питер. 2016. 400 с.
10. Бреснахэн К., Блум Р. Linux на практике. Санкт-Петербург : Питер.2017. 384 с.
11. Колисниченко Д. Linux от новичка к профессионалу. Санкт-Петербург : БХВ-Питер. 2016. 672 с.
12. Шотт У. Командная строка Linux. Полное руководство. Санкт-Петербург : Питер. 2016. 480 с.
13. Тейлор Д., Перри Б. Сценарии командной оболочки. Linux, OS X и Unix. СПб: Питер. 2017. 448 с.
14. Роббинс А. Bash. Карманный справочник системного администратора. СПб.: ООО "Альфа-книга". 2017. 152 с.
15. Сейерс Э. Х., Милл А. Docker на практике. Москва : ДМК. 2019. 516 с.
16. Джиджи С. Осваиваем Kubernetes. Оркестрация контейнерных архитектур. СПб.: Питер. 2019. 400 с.
17. Ричардсон К. Микросервисы. Паттерны разработки и рефакторинга. Санкт-Петербург : Питер. 2020. 544 с.
18. Хорсдал Кристиан Микросервисы на платформе .NET. Санкт-Петербург : Питер 2018. 352 с.
19. Танебаум Э., Бос Х. Современные операционные системы. Санкт-Петербург : Питер. 2019. 1120 с.
20. Дейтел Х., Дейтел П., Чофнес Д. Операционые системы. Основы и принципы. Москва : Бином. 2016. 1024 с.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Розділ 1. Призначення, функції та архітектура операційної системи Linux

Лекція 1. Концепції програмування в Linux. Завантаження і установка інструментарію для роботи з відкритим вихідним кодом

Історія виникнення Linux. Стандарт POSIX. Дистрибутиви Linux (Debian, Ubuntu, Red Hat, SUSE, Solaris, HP-UX и AIX) та їх характеристика. Огляд Linux, системного програмування. Ядро, конфігурування параметрів ядра, методи конфігурування ядра. Драйвери пристроїв. Файли і файлова система. Процеси. Користувачі і групи. Права доступу. Сигнали. Міжпроцесна взаємодія. Заголовки. Обробка помилок. Пошук інструментарію. Формати поширення. Архівні файли.

Розділ 2. Управління Linux-пакетами

Лекція 2. Управління Linux-пакетами

Менеджери пакетів та їх характеристика. Що краще: бінарні пакети чи пакети з початковими кодами? Диспетчер пакетів RPM (Red Hat Package Manager), команда rpm. Огляд пакетів RPM. Управління пакетами .deb в системі Ubuntu (команда dpkg). Оновлення пакетів: Інструмент Apt: Advanced Package Tool. Пакети tarball. Менеджер пакетів Aptitude. Synaptic: передовий GUI-інтерфейс для APT. Огляд пакетів RPM (Red Hat Package Manager). Інструмент Yum (Yellowdog Updater Modified). Перевірка автентичності. Система контролю версій проектів з відкритим вихідним кодом - GIT.

Розділ 3. Файлова система Linux

Лекція 3. Особливості файлової системи

Організація файлової системи. Стандарт FHS (Filesystem Hierarchy Standard). Стандартні підкаталоги та їх вміст. Абсолютні і відносні шляхи до файлу. Типи файлів та їх характеристики: звичайні (-), каталоги (d), файли байт-орієнтованих символьних пристроїв (c), файли блочно-орієнтованих пристроїв (b), локальні сокети (sokets), іменовані канали (pipe), символічні посилання. Атрибути файлів. Біти режиму. Отримання інформації про файл – системний виклик stat().Перенаправлення введення-виведення. Файловий менеджер mc (Midnight Commander).

Лекція 4. Робота з файловою системою

Оболонка bash. Користувачі (UID) і групи (GID). Права доступу для файлів та для каталогів. Редагування прав доступу - команда chmod. Команда ls – перегляд атрибутів. Додаткові атрибути файлів (SUID, SGID). Робота з привілеями root – sudo. Зміна власника файлу і групи. Команда отримання відомостей про ім'я користувача (whoami). Команди для роботи з файлами (man, cat, touch, ln, cp, mv, head, tail, less) та каталогами (cd, ls, mkdir, rm, cp, tree). Система контролю версій програмного забезпечення Git.

Лекція 5. Фільтрація даних у файлі. Низькорівневе-введення виведення

Утиліта grep – потужний інструмент системного адміністратора. Приклади роботи. Базові регулярні вирази. Розширений глобальний вираз – egrep. Модель введення-виведення в системах UNIX. Системні виклики (open(), read(), write(), close(), lseek()).

Лекція 6.Сценарії в оболонці bash

Скрипт. Створення сценарію. Перетворення сценарію у виконуваний файл. Змінні середовища та користувача. Підстановка команд. Математичні операції. Управляючі конструкції if-then/if-then-else. Оператори циклу while/until/for. Функції. Виклик функції з аргументами. Глобальні та локальні змінні в функціях. Передача параметрів у командному рядку. Передача функції масива в якості аргумента. Приклади сценаріїв.

Розділ 4. Управління процесами

Лекція 7. Управління процесами, системні виклики

Програми, процеси і потоки. Виділення ідентифікатора процесу. Ідентифікатор дочірнього процесу Ієрархія процесів. pid_t. Отримання ідентифікаторів батьківського процесу (PID) і дочірнього процесу PPID. Привілейовані процеси. Запуск нового процесу: Сімейство викликів exec. Системні виклики fork(). Завершення процесу: Інші способи завершення. atexit(), on_exit(). SIGCHLD. Очікування завершених дочірніх процесів: Очікування певного процесу. Ще більше гнучкості при очікуванні. Стиль BSD: wait3() і wait4(). Запуск і очікування нового процесу. Зомбі.

Лекція 8. Управління процесами в оболонці Bash

Системні процеси. Демони. Міжпроцесна взаємодія (канали, сигнали, сокети). Фонові процеси. Таблиця процесів. Моніторинг процесів. Управління сигналами.

Лекція 9. Потоковість

Бінарні модулі, процеси і потоки. Багатопоточність: витрати багатопоточності. Альтернативи багатопоточності. Поточні моделі: Поточність на рівні користувача. Комбінована поточність. Співпрограми і фібери. Шаблони поточності: потік на з'єднання. Потік, керований подією. Конкурентність, паралелізм і гонки. Синхронізація: м'ютекси. Взаємні блокування. Р-потоки. Реалізація поточності в Linux. API для роботи з Р-потоками. Зв'язування Р-потоків. Створення потоків. Ідентифікатори потоків. Завершення потоків. Самозавершення. Завершення інших потоків. Приєднання і від'єднання потоків: Приєднання потоків. Від'єднання потоків. Приклад поточності. М'ютекси Р-потоків: Ініціалізація м'ютексів. Запирання м'ютексів. Відмикання м'ютексів. Приклад використання м'ютексів.

Розділ 5. Управління пам'яттю

Лекція 10. Управління пам'яттю

Адресний простір процесу: Сторінки та їх підкачка. Області пам'яті. Виділення динамічної пам'яті: Виділення масивів. Зміна розміру виділених областей. Звільнення динамічної пам'яті. Вирівнювання. Управління сегментом даних. Анонімні відображення в пам'яті: Створення анонімних відображень в пам'яті. Відображення /dev/zero. Розширене виділення пам'яті. Налагодження при операціях виділення пам'яті. Виділення пам'яті на основі стека: дублювання рядків в стеці. Масиви змінної довжини. Вибір механізму виділення пам'яті.

Управління пам'яттю: Установка байтів. Порівняння байтів. Переміщення байтів. Пошук байтів. Переклацування байтів. Блокування пам'яті: Блокування частини адресного простору. Блокування всього адресного простору. Розблокування пам'яті. Ліміти блокування. Чи знаходиться сторінка у фізичній пам'яті? Поступатися виділенням пам'яті.

Розділ 6. Застосування технології контейнерів

Лекція 11. Розробка та впровадження програмного забезпечення за допомогою технології контейнерів

Контейнери та їх призначення. Відмінності між віртуальними машинами та контейнерами. Docker і контейнери. Коротка історія Docker. Архітектура Docker. Базові технології Docker. Інструментальні засоби Docker. Установка Docker в ОС Linux.. Опції, основні команди управління, підкоманди Docker.

Лекція 12. Практична робота з Docker

Хостинг для Docker. Образи, реєстр образів. Команди для роботи з реєстром. Рівні образу контейнера. Команди управління контейнерами. Робота з образами. Приклади роботи з Docker. Базові образи Dockerfile, Docker Compose. Інструкції Dockerfile. Команди для роботи з Docker Compose. Встановлення зв'язку контейнерів із зовнішнім світом. Команда run. З'єднання між контейнерами.

Лекція 13. Життєвий цикл програмного забезпечення при використанні Docker.

Використання Docker в процесі розробки: Традиційне вітання світу. Життєвий цикл контейнера Автоматизація з використанням Compose. Порядок роботи Compose. Установка Docker Compose в Ubuntu 18.04. Створення простого веб-застосування: створення основної веб-сторінки. Приклад роботи Docker з БД Mongo.

Лекція 14. Мікросервіси і контейнери

Мікросервіси Оркестрація, кластеризація і управління: інструментальні засоби кластеризації і оркестрації. Swarm. Fleet. Kubernetes.

Розділ 7. Операційні системи сімейства Windows

Лекція 15. Операційні системи сімейства Windows

Історія Windows аж до Windows 10. Програмування в Windows: власний інтерфейс прикладного програмування NT. Інтерфейс прикладного програмування Win32. Реєстр Windows. Структура системи: структура операційної системи. Завантаження Windows. Реалізація диспетчера об'єктів. Підсистема середовища, DLL і служби користувацького режиму. Регістри.

Лекція 16. Процеси і потоки в Windows

Базові поняття: процеси і потоки в сучасних ОС. Багатопотоковість. Складові елементи процесів і потоків. Стани процесів і потоків. Керуючі блоки процесів і потоків. Образи процесів і потоків. Організація контексту перемикання.

Процеси і потоки в Windows: фундаментальні концепції. Виклики API для управління завданнями, процесами, потоками і волокнами. Реалізація процесів і потоків.

Лекція 17. Управління пам'яттю в Windows

Організація оперативної пам'яті. Віртуальна пам'ять. Сторінкова організація пам'яті. Сегментна організація пам'яті. Сегментно-сторінкова організація пам'яті. Управління пам'яттю: фундаментальні концепції. Системні виклики управління пам'яттю. Реалізація управління пам'яттю. Кешування в Windows.

Лекція 18 Введення-виведення в Windows

Введення-виведення в Windows: фундаментальні концепції. Виклики інтерфейсу прикладного програмування вводу-виводу. Реалізація введення-виведення. Файлова система Windows NT: фундаментальні концепції. Реалізація файлової системи NTFS. Управління електроживленням в Windows. Безпека в Windows 8 (10): фундаментальні концепції. Виклики інтерфейсу прикладного програмування безпеки. Реалізація безпеки. Полегшення умов безпеки.

6. Самостійна робота студента

Розділ 1. Призначення, функції та архітектура операційної системи Linux

Історія створення Linux, принципи, філософія, стандарти та нормативна база Linux

Розділ 2. Управління Linux-пакетами

Центр застосувань Gnome Software для роботи з Deb та Rpm пакетами. Робота з програмою APPGRID.

Розділ 3. Файлова система Linux

Операції, які не вписуються у загальну модель універсального введення-виведення *iocl().* Архівація даних. Редагування файлів.

Розділ 4. Управління процесами

Ідентифікація процесів. Моніторинг дочірніх процесів

Розділ 5. Управління пам'яттю

Операції з віртуальною пам'яттю. Рівень блочного розподілу пам'яті.

Розділ 6. Застосування технології контейнерів

Оркестрація, кластеризація і управління.

Розділ 7. Операційні системи сімейства Windows

Інтерфейс прикладного програмування Win 32

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Відвідування лекційних та лабораторних занять є обов'язковим за винятком поважних причин (хвороби, форс-мажорних обставин).

В разі пропущення занять з поважних причин викладач надає можливість студенту виконати усі або деякі лабораторні завдання (винятком є виконання деяких завдань у зв'язку із закінченням навчального процесу).

В разі пропущення занять без поважних причин, а також через порушення граничного терміну виконання завдання (deadline) студент може отримати 80% від максимальної оцінки відповідне завдання.

Протягом семестру студенти:

- виконують та захищають лабораторні роботи у відповідні терміни (на кожну лабораторну роботу відводиться два тижні для здачі),
- пишуть модульну контрольну роботу,
- повинні позитивно закрити дві атестації (в кінці березня та в середині травня),
- по закінченні навчального процесу складають екзамен.

8. Види контролю та рейтингова система оцінювання результатів навчання (РСО)

Система рейтингових (вагових) балів та критерії оцінювання

1) Робота на лекціях

На лекціях може бути проведено бліцопитування студентів. Такі опитування проводяться на довільних лекціях 5 разів протягом семестру, наприкінці лекції. Ваговий бал за вірну відповідь - 1. Максимальна кількість балів, що може отримати кожен студент за семестр - 5.

2) Лабораторні практикуми

Максимальна кількість балів за усі виконані комп'ютерні практикуми дорівнює 50 балів. Розподіл балів серед лабораторних практикумів наступний:

№ з/п	Назва лабораторного практикуму	Кількість балів
1	Установка Ubuntu Linux на віртуальній машині VirtualBox	3
2	Менеджери для роботи з пакетами програм Linux	4
3	Робота в оболонці Bash, середовище оточення	4
4	Робота з файловою системою ОС Linux	5
5	Створенні сценаріїв в оболонці Bash	5
6	Робота з процесами ОС Linux	4
7	Потоки, перенаправлення потоків	5
8	Установка Docker	5
9	9.1. Створення проекту для web-розробки, який складається з наборів контейнерів з використанням Docker Compose та Dockerfile. 9.2. Створення проекту, що складається з контейтерів Django+PostgreSQL, з використанням Docker Compose та Dockerfile	10
Всього:		45

Критерії оцінювання:

Виконання лабораторного практикуму:

- виконаний своєчасно (протягом двох тижнів з моменту видачі), у повному обсязі – відповідний бал згідно номеру комп'ютерного практикуму;
- виконаний із запізненням – знімається 10 – 30% від максимальної кількості балів в залежності від терміну запізнення;

- виконаний не самостійно, із запізненням – знімається 50% від максимальної кількості балів;
- невиконани1 протягом відведеного часу – 0 балів.

3) Модульна контрольна робота

Максимальна кількість балів за модульну контрольну роботу дорівнює 10 балів.

Якість виконання роботи:

- усі відповіді вірні та повні –10 балів,
- у відповідях допущені несуттєві неточності – 8 балів,
- половина відповідей вірна – 5 балів,
- відповіді з суттєвими неточностями, але без критичних помилок – 2 бали,
- менше половини відповідей вірна – 0 балів.

Штрафні та заохочувальні бали за:

- активність на комп'ютерних практикумах + 2 бали
- виконання комп'ютерного практикуму з використанням власного оптимального алгоритму + 1 бали
- відсутність на занятті без поважної причини – 2 бали
- несвоєчасна здача комп'ютерного практикуму (пізніше ніж за тиждень) – 0,5 балів;

4) Складання іспиту

Максимальний ваговий бал $r_{ісп}$=40

На іспиті студент виконує письмову контрольну роботу, яка містить два теоретичних питання і одне практичне питання. Теоретичні питання оцінюються максимально по 10 балів, практичне – 20 балів.

Умови позитивної проміжної атестації

Для отримання „зараховано" з першої проміжної атестації студент матиме не менше ніж 11 балів (за умови, що за 8 тижнів згідно з календарним планом контрольних заходів „ідеальний" студент має отримати 3+4+4+5+5 = 21 бал).

Для отримання „зараховано" з другої проміжної атестації студент матиме не менше ніж 22 балів (за умови, що за 14 тижнів згідно з календарним планом контрольних заходів „ідеальний" студент має отримати 21 + 4 +5 +5 +10 =45 балів).

Умови допуску до *іспиту*

Необхідною умовою допуску до іспиту є зарахування усіх комп'ютерних практикумів та виконання модульної контрольної роботи, а також стартовий рейтинг (R_c) не менше 40 балів.

Розрахунок шкали (R) рейтингу:

Сума вагових балів контрольних заходів протягом семестру (шкала рейтингу) складає:

$$R = r_{лек} + r_{практ} + r_{мод} + r_{ісп} = 5+45 + 10 + 40= 100 \text{ балів.}$$

Максимальний стартовий рейтинг становить $R_c = r_{лек} + r_{практ} + r_{мод} =$ 60 балів.

Рейтинг іспиту дорівнює 40 балів. Мінімальний рейтинг допуску до іспиту становить 40 балів.

Таким чином, рейтингова шкала з кредитного модуля складає

R = 60+40=100 балів.

Для отримання студентом відповідних оцінок рейтингова оцінка студента переводиться згідно таблиці:

Бали	**Оцінка**
95 - 100	Відмінно
85 - 94	Дуже добре
75 - 84	Добре
65 - 74	Задовільно
60 - 64	Достатньо
Менше 60	Незадовільно
R < 40 є незараховані роботи комп'ютерного практикуму або не виконані інші умови допуску до екзамену	Не допущено

Робочу програму навчальної дисципліни (силабус):

Складено доцент, д.т.н., доцент, Левченко Лариса Олексіївна

Ухвалено кафедрою _____ (протокол № ___ від _____)

Погоджено Методичною комісією факультету[1] (протокол № __ від _____)

[1] Методичною радою університету – для загальноуніверситетських дисциплін.