



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 в енергетиці |
| Статус дисципліни | Нормативна |
| Форма навчання | Очна (денна) |
| Рік підготовки, семестр | 2 курс, весняний семестр |
| Обсяг дисципліни | 5 кредити, 150 годин, з яких 72 години аудиторних (36 год лекції, 36 год практичні), (78 годин становить самостійна робота) |
| Семестровий контроль/ контрольні заходи | Іспит/модульна контрольна робота |
| Розклад занять | http://rozklad.kpi.ua/ |
| Мова викладання | Українська |
| Інформація про керівника курсу / викладачів | Лектор: тел.: 0932358725, старший викладач Сарибога Ганна Володимирівна, sarigana-eds@iit.kpi.ua, телеграм https://t.me/binarysvit Практичні : Сарибога Ганна Володимирівна |
| Розміщення курсу | https://classroom.google.com/c/NjU3Nzq2MjMwMjY3?cjc=cqceo_zt |

Програма навчальної дисципліни

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

Дисципліна «Архітектура системного програмного забезпечення» присвячена вивченню структури і принципів побудови операційних систем, архітектурі комп'ютера, створенню системного програмного забезпечення для операційних систем Linux/Unix, компіляторів мовами C, асемблер, Python.

Метою дисципліни є отримання студентами знань про будову та принципи функціонування сучасних операційних систем та формування практичних навичок роботи з ними, про інформаційні моделі даних, здатність створювати системне програмне забезпечення мовами низького рівня (C, асемблер) для реалізації задач автоматизації обробки даних та створення систем автоматизації процесів.

Предмет дисципліни – вивчення архітектури комп'ютера, системних програм, операційних систем, основних функцій, принципів роботи, засобів розробки системних програм. Застосовувати на практиці ефективні підходи щодо керування пам'яттю, потоками та процесами.

Завдання У результаті вивчення дисципліни у студентів повинні сформуватися наступні компетентності:

фахові (ФК):

- здатність розробляти архітектуру, модулі та компоненти програмних систем (ФК 3),
- володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних (ФК 7).

Після засвоєння навчальної дисципліни студенти мають продемонструвати наступні **програмні результати навчання (ПРН):**

- знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення (ПРН5);
- вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання (ПРН11);
- застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення (ПРН12).

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

Згідно з робочим навчальним планом навчальна дисципліна «Архітектура системного програмного забезпечення» викладається студентам другого року підготовки першого (бакалаврського) рівня вищої освіти ступеня “бакалавр” спеціальності «Інженерія програмного забезпечення» освітньої програми «Інженерія програмного забезпечення інтелектуальних кіберфізичних систем в енергетиці» у четвертому навчальному семестрі. Дисципліна «Об'єктно-орієнтований аналіз та конструювання програмних систем» забезпечується дисципліною «Об'єктно-орієнтований аналіз та конструювання програмних систем» та забезпечує вивчення таких навчальних дисциплін як: «Методологія розробки інтелектуальних комп'ютерних програм», «Безпека програмного забезпечення».

3. Зміст навчальної дисципліни

Розділ 1. Апаратні та програмні засоби комп'ютерної системи

Тема 1.1 Архітектура комп'ютера

Тема 1.2 Архітектура системи команд

Тема 1.3 Архітектура операційних систем

Розділ 2. Системне програмне забезпечення

Тема 2.1. Системне програмування

Тема 2.2. Інструменти системного програмування

Тема 2.3. Мовний процесор

Тема 2.4. Розробка компілятора мовами LEX & YACC та/або C/C++/Java

Тема 2.5. Створення пакетів для програмної установки

Тема 2.6. Програмування пристроїв мовою assembler

Розділ 3. Операційні системи

Тема 3.1. Програмування ядра ОС Linux/Unix

Тема 3.2. Керування пам'яттю та процесором

Тема 3.3. Засоби виявлення помилок

Тема 3.4. Планування процесів в ОС Linux/Windows

Тема 3.5. Операційні системи реального часу

Розділ 4. Мережні засоби операційних систем

Тема 4.1.. Загальні принципи мережевої підтримки.

Тема 4.2. Реалізація стека протоколів Інтернету.

Тема 4.3. Програмний інтерфейс сокетів Берклі.

Тема 4.4. Архітектура мережної підтримки Linux та Windows.

4. Навчальні матеріали та ресурси

Основна література

1. Andrew S. Tanenbaum, Herbert Bos. *Modern Operating Systems (4th Edition)*
<http://index-of.es/Varios-2/ModernOperatingSystemsEdition.pdf>

2. *Assembly Language Tutorial*
https://www.tutorialspoint.com/assembly_programming/assembly_tutorial.pdf

3. Stallings, William. *Operating systems: internals and design principles / William Stallings.* – 7 th ed.

4. Prentice Hall, New Jersey, 2012, p.769. ISBN-13:978-0-13- 230998-1

5. Kusswurm Daniel. *Modern X86 Assembly Language Programming/ Daniel Kusswurm.* - Apress, 2019. — 604 p

6. William Stallings. *Operating Systems: Internals and Design Principles, 9th Edition.* – Pearson, 2018. ISBN-10: 0-13-467095-7 | ISBN-13: 978-0-13- 467095-9

Додаткова література

1. Шеховцов В.А. *Операційні системи/ В.А. Шеховцов.* – Київ: Видавнича група BHV, 2005. – 576 с., іл.

2. Gary Nutt. *Operating Systems (3rd Edition), ISBN 978-020-177344-6, Published by Pearson (2003).* 8. Abraham Silberschatz, Peter B. Galvin, Greg Gagne. *Operating System Concepts (10th edition), ISBN 978-111-945633-9, Published by Wiley (2018).*

Онлайн-ресурси

1. *Assembly Language Tutorial* -
https://www.tutorialspoint.com/assembly_programming/assembly_tutorial.pdf

2. *Windows Assembly Programming Tutorial* -
<https://doc.lagout.org/operating%20system%20/Windows/winasmmtut.pdf>

3. *Operating System Tutorial* – http://www.sncwgs.ac.in/wp-content/uploads/2015/11/operating_system_tutorial.pdf

Навчальний контент

1. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

| № з/п | Назва теми лекції та перелік основних питань (перелік дидактичних засобів, посилання на інформаційні джерела) |
|-------|--|
| 1 | 2 |
| 1 | Архітектура комп'ютера <u>Основні питання:</u> типи архітектури: принстонська (фон Неймана) і гарвардська, основні компоненти, завдання, принципи роботи. |
| 2 | Архітектура системи команд <u>Основні питання:</u> Еволюція архітектур системи команд. Перші акумуляторні архітектури. Архітектури з індексними регістрами. Архітектури з регістрами загального призначення CISC, RISC, VLIW. Класифікація архітектур системи команд. Стекові архітектури. Акумуляторні архітектури. Архітектури «регістр-пам'ять» і «регістр-регістр». Архітектури «пам'ять-пам'ять». Адресність команд. Класифікація команд |
| 3 | Архітектура операційних систем <u>Основні питання:</u> Типи даних, оператори, конкатенація, управляючі структури, обробка винятків, функції |
| 4 | Системне програмування <u>Основні питання:</u> особливості, приклади використання, інструменти розробки |
| 5 | Інструменти системного програмування <u>Основні питання:</u> асемблери, транслятори, компонувальники, препроцесори налагоджувач, текстові редактори, редактори початкових текстів, інтегроване середовище розробки, бібліотеки підпрограм |
| 6 | Мовний процесор <u>Основні питання:</u> Програми та програмне забезпечення. Системне програмування. Етапи проходження програми. |
| 7 | Розробка компілятора мовами LEX & YACC та/або C/C++/Java <u>Основні питання:</u> Етапи розробки компілятора LEX & YACC. Базовий аналіз. Прогнозні аналізатори. Таблиці символів. Виявлення помилок. Генерація коду. Оптимізація коду |
| 8 | Створення пакетів для програмної установки <u>Основні питання:</u> принципи створення нового додатку Windows Forms в Visual C++. методи компіляції, запуску та відлагоджування проекту Windows Forms. |
| 9 | Програмування пристроїв мовою assembler <u>Основні питання:</u> Мова Асемблера: означення, програмне забезпечення. Основні терміни і означення. Програмне забезпечення: Emu8086, TASM, MASM, WASM, Debug, Шістнадцятковий редактор, Текстовий редактор. Мова Асемблера: системи числення, перетворення чисел, типи даних |
| 10 | Ядро ОС Linux/Unix <u>Основні питання:</u> Концепція API. Стандарти системного програмування в Linux. Основні концепції програмування в Linux. Файли та файлова система. Процеси. Користувачі та групи. Права доступу. Сигнали та міжпроцесна взаємодія. Обробка помилок |
| 11 | Керування пам'яттю та процесором <u>Основні питання:</u> Призначення та класифікація процесорів. Структура процесора. Пристрій керування (Control Unit). Операційні пристрої. |

| 1 | 2 |
|----|---|
| 12 | Засоби виявлення помилок. <u>Основні питання:</u> створення дамів, трасування, повідомлень про помилки та інших методів налагодження та виявлення помилок. |
| 13 | Планування процесів в ОС Linux/Windows <u>Основні питання:</u> Загальні принципи планування. Види планування. Стратегії планування. Витісняльна і невитісняльна багатозадачність. Алгоритми планування. |
| 14 | Операційні системи реального часу <u>Основні питання:</u> Операційні системи реального часу RTOS |
| 15 | Загальні принципи мережевої підтримки. <u>Основні питання:</u> Мережні операційні системи, адміністрування та управління мережею Мережні операційні системи Мережна операційна система (ОС) - це пакет програм, що забезпечує реалізацію та управління мережею |
| 16 | Реалізація стека протоколів Інтернету <u>Основні питання:</u> архітектура, реалізація, особливості стека TCP / IP |
| 17 | Програмний інтерфейс сокетів Берклі <u>Основні питання:</u> Інтерфейс сокетів Берклі, реалізація |
| 18 | Архітектура мережної підтримки Linux та Windows. Програмний інтерфейс Windows Sockets. <u>Основні питання:</u> Особливості керування розподіленемі Архітектура мережної підтримки Linux та Windows. |

Практичні заняття

Основні завдання циклу практичних занять полягають у набутті студентами практичних навичок у розробці програмного забезпечення на базі об'єктно-орієнтованого підходу та аналізу ІТ-продукту загалом.

| № з/п | Назва теми заняття |
|-------|---|
| 1 | Створити програму LEX для видалення рядків коментарів |
| 2 | Створити програму YACC для розпізнавання дійсних ідентифікаторів, операторів і ключових слів у заданому текстовому файлі (програма C) |
| 3 | Спроекувати, розробити та впровадити програму на C/C++ |
| 4 | Спроекувати, розробити та впровадити програму на Java для моделювання роботи алгоритмів планування найкоротшого часу |
| 5 | Спроекувати, розробити та реалізувати програму Java для реалізації алгоритму Банкіра. |
| 6 | Спроекувати, розробити та реалізувати програму C/C++ для реалізації алгоритму Банкіра |
| 7 | Спроекувати, розробити та реалізувати програму C/C++ для реалізації алгоритмів заміни сторінок LRU та FIFO |
| 8 | Програмування мовою assembler. Арифметичні операції |
| 9 | Програмування мовою assembler. Робота зі стеком |
| 10 | Програмування мовою assembler. Переривання |
| 11 | Програмування мовою assembler. Робота з пам'яттю |
| 12 | Програмування мовою assembler. Регістри |

| | |
|----|---|
| 1 | 2 |
| 13 | Спроекувати, розробити та реалізувати програму C/C++ для моделювання числового калькулятора |
| 14 | Спроекувати, розробити та реалізувати програму Java для моделювання числового калькулятора |
| 15 | Реалізація багатопоточності |
| 16 | Багатозадачність та багатопроцесорності |
| 17 | Створення простого драйвера за допомогою Windows Driver Frameworks (WDF) |
| 18 | Керування системним адмініструванням Windows за допомогою Python |

2. Самостійна робота студента

| № з/п | Назва теми, що виноситься на самостійне опрацювання |
|---|--|
| 1 | Засоби Windows Server для моніторингу операційної системи та програм на серверному комп'ютері. |
| 2 | Встановлення програмного забезпечення MASM64 SDK |
| 3 | Команда make в Linux |
| 4 | Linux Bash |
| 5 | Модуль ОС Python |
| 6 | Фільтри Linux |
| Питання, що виносяться на самостійне опрацювання студентів, контролюються шляхом їх включення до переліку питань модульних контрольних робіт. | |

Політика та контроль

1. Політика навчальної дисципліни (освітнього компонента)

Застосовуються стратегії активного і колективного навчання, які визначаються наступними методами і технологіями:

- 1) методи проблемного навчання (проблемний виклад, частково-пошуковий (евристична бесіда) і дослідницький метод);
- 2) особистісно-орієнтовані (розвиваючі) технології, засновані на активних формах і методах навчання («мозковий штурм», «аналіз ситуацій» дискусія, експрес-конференція);
- 3) інформаційно-комунікаційні технології, що забезпечують проблемно-дослідницький характер процесу навчання та активізацію самостійної роботи студентів (електронні презентації для лекційних занять, використання аудіо-, відео-підтримки навчальних занять).
- 4) лекційні та лабораторні заняття відносяться до аудиторних занять. Відвідування аудиторних занять є обов'язковим;
- 5) правила поведінки на заняттях: активність, підготовка коротких доповідей чи текстів, відключення телефонів, використання засобів зв'язку для пошуку інформації на гугл-диску викладача чи в інтернеті тощо;
- 6) правила захисту лабораторних робіт. На лабораторних заняттях студенти під керівництвом викладача вивчають методику експериментальних досліджень. На кожній лабораторній роботі студенти оформляють звіт у письмовому вигляді. До звіту заноситься перебіг досліду, його результати і даються пояснення отриманих результатів з урахуванням похибок експерименту.

7) індивідуальні завдання з дисципліни (реферати, розрахункові, графічні, тощо) видаються студентам в терміни, передбачені вищим навчальним закладом. Індивідуальні завдання виконуються студентом самостійно при консультуванні викладачем. Допускаються випадки виконання комплексної тематики кількома студентами.

8) правила призначення заохочувальних балів: своєчасне виконання та здача лабораторних, індивідуальних завдань, нестандартний підхід до вирішення певного завдання;

9) правила призначення штрафних балів: несвоєчасне виконання лабораторних та індивідуальних завдань, а також користування допоміжними засобами (наприклад, мобільний телефон, конспект лекцій) під час виконання контрольної роботи.

10) політика дедлайнів та перескладань: невчасно виконані та здані лабораторні роботи оцінюються нижчою оцінкою (-10-20% від загальної підсумкової оцінки).

11) політика щодо академічної доброчесності: письмові роботи можуть перевірятися на наявність плагіату і допускаються до захисту із коректними текстовими запозиченнями не більше 40%. Списування під час контрольних робіт та екзаменів заборонені.

12) інші вимоги, що не суперечать законодавству України та нормативним документам Університету:

– політика щодо відвідування: відвідування занять є обов'язковим компонентом оцінювання, за яке нараховуються бали. За об'єктивних причин (підтверджених документально) дозволяється перескладання пропущених тем курсу.

– політика щодо виконання завдань: позитивно оцінюється відповідальність, старанність, креативність, фундаментальність.

2. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

1. Оцінка з дисципліни виставляється за багато бальною системою, з подальшим перерахуванням у 4-бальну.

2. Максимальна кількість балів з дисципліни дорівнює 100.

3. Нарахування балів по окремих видах робіт:

Рейтинг студента з кредитного модуля складається з балів, що він отримав за:

- виконання практичних робіт;
- написання контрольної роботи (МКР).

3. Система рейтингових (вагових) балів та критерії оцінювання

1. Виконання практичних робіт

Оцінюються 18 робіт, передбачених робочою програмою.

Максимальний ваговий бал ЛР = 54.

Сума вагових балів практичних робіт:

| № з/п | Назва теми заняття | Максимальний ваговий бал |
|-------|---|--------------------------|
| 1 | 2 | 3 |
| 1 | Створити програму LEX для видалення рядків коментарів | 3 |
| 2 | Створити програму YACC для розпізнавання дійсних ідентифікаторів, операторів і ключових слів у заданому текстовому файлі (програма С) | 3 |

| 1 | 2 | 3 |
|----|--|----|
| 3 | Спроекувати, розробити та впровадити програму на C/C++ | 3 |
| 4 | Спроекувати, розробити та впровадити програму на Java для моделювання роботи алгоритмів планування найкоротшого часу | 3 |
| 5 | Спроекувати, розробити та реалізувати програму Java для реалізації алгоритму Банкіра. | 3 |
| 6 | Спроекувати, розробити та реалізувати програму C/C++ для реалізації алгоритму Банкіра | 3 |
| 7 | Спроекувати, розробити та реалізувати програму C/C++ для реалізації алгоритмів заміни сторінок LRU та FIFO | 3 |
| 8 | Програмування мовою assembler. Арифметичні операції | 3 |
| 9 | Програмування мовою assembler. Робота зі стеком | 3 |
| 10 | Програмування мовою assembler. Переривання | 3 |
| 11 | Програмування мовою assembler. Робота з пам'яттю | 3 |
| 12 | Програмування мовою assembler. Регістри | 3 |
| 13 | Спроекувати, розробити та реалізувати програму C/C++ для моделювання числового калькулятора | 3 |
| 14 | Спроекувати, розробити та реалізувати програму Java для моделювання числового калькулятора | 3 |
| 15 | Реалізація багатопоточності | 3 |
| 16 | Багатозадачність та багатопроцесорності | 3 |
| 17 | Створення простого драйвера за допомогою Windows Driver Frameworks (WDF) | 3 |
| 18 | Керування системним адмініструванням Windows за допомогою Python | 3 |
| | Разом | 54 |

Оцінювання лабораторних робіт:

- якщо робота виконана невчасно знімається 10-30% від максимальної кількості балів (кількість процентів залежить від терміну запізнення);
- якщо робота виконана не самостійно та простежується не індивідуальне виконання то знімається 50% від максимальної кількості балів;
- якщо в програмі не витримані основні правила створення програмних продуктів (модульність, дружній інтерфейс, наявність коментарів та т.п.) знімається 5%.

2. Модульний контроль

На одному з лекційних занять проводиться модульна контрольна робота: Максимальний ваговий бал МКР = 6.

Оцінювання модульної контрольної роботи виконується наступним чином:

- якщо на всі питання дані повні та чітко аргументовані відповіді, контрольна виконана охайно, з дотримання основних правил, то виставляється 5- 6 балів;
- якщо методика виконання запропонованого завдання розроблена вірно, але допущені не принципові помилки у теоретичному описі або розрахунках, то виставляється 4 - 5 балів;

- від 3 до 5 балів нараховується, якщо методика виконання завдання розроблена в основному вірно, але допущені деякі з наступних помилок: помилки у представленні вихідних даних, не обґрунтовані теоретичні рішення, помилки у методиці розрахунків;
- нижче 3 балів нараховується, якщо завдання не виконане або допущені грубі помилки.

3. Екзамен

Екзамен відбувається у письмовій формі. Максимальна оцінка за екзамен складає $rEK = 40$ балів.

Умови позитивної проміжної атестації

Для отримання „зараховано” з першої проміжної атестації студент повинен мати не менше, ніж 12 балів (за умови, що за 8 тижнів згідно з календарним планом контрольних заходів студент повинен отримати 24 бали).

Для отримання „зараховано” з другої проміжної атестації студент повинен мати не менше, ніж 40 балів (за умови, що за 14 тижнів згідно з календарним планом контрольних заходів студент повинен отримати 60 балів).

Розрахунок шкали (R) рейтингу:

Сума вагових балів контрольних заходів протягом семестру складає:

$$R=54 +6+40 = 100 \text{ балів}$$

Таким чином, рейтингова шкала з кредитного модуля складає 100 балів.

Умови допуску до іспиту: зарахування всіх лабораторних робіт, а також стартовий рейтинг $r \geq 40$ балів.

Для отримання студентом відповідних оцінок (ECTS та традиційних) його рейтингова оцінка RD переводиться згідно таблиці:

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

| Кількість балів | Оцінка |
|---------------------------|--------------|
| 100-95 | Відмінно |
| 94-85 | Дуже добре |
| 84-75 | Добре |
| 74-65 | Задовільно |
| 64-60 | Достатньо |
| Менше 60 | Незадовільно |
| Не виконані умови допуску | Не допущено |

Додаткова інформація з дисципліни (освітнього компонента)

Перелік питань, які виносяться на семестровий контроль:

1. Визначення системного програмного забезпечення (System Software)
2. Типи системного програмного забезпечення
3. Визначення операційної системи
4. Функції операційної системи
5. Типи операційних систем
6. Архітектура операційних систем (монолітна, багаторівнева, мікроядерна)
7. Класифікації операційних систем
8. Визначення процесу

9. Стани процесів і їхні переходи
10. Основні операції над процесами
11. Обробка та типи переривань
12. Класифікація процесів
13. Алгоритми планування процесів
14. Організація та ієрархія пам'яті
15. Стратегії керування пам'яттю
16. Розподіли пам'яті
17. Механізм відображення віртуальних адрес в реальні
18. Типи організації віртуальної пам'яті
19. Стратегії керування віртуальною пам'яттю
20. Призначення та класифікація процесорів
21. Структура процесора
22. Пристрій керування (Control Unit)
23. Операційні пристрої
24. Етапи розробки програми
25. Засоби Windows Server для моніторингу операційної системи та програм на серверному комп'ютері.
26. Встановлення програмного забезпечення MASM64 SDK
27. Команда make в Linux
28. Linux Bash
29. Модуль ОС Python
30. Фільтри Linux
31. Чим Debian GNU/Linux відрізняється від Ubuntu? Назвіть два аспекти.
32. Ви плануєте встановити дистрибутив Linux у новому середовищі. Назвіть чотири речі, які слід врахувати, вибираючи дистрибутив.

Робочу програму навчальної дисципліни (силабус):

Складено *ст. викл. кафедри інженерії програмного забезпечення в енергетиці*

Ганна Сарибога

Ухвалено *кафедрою ІПЗЕ (протокол № 28 від 15.05.2023)*

Погоджено *НН ІАТЕ ім.Ігоря Сікорського (протокол № 9 від 26.05.2023)*