

**АННОТАЦИЯ К РАБОЧЕЙ ПРОГРАММЕ ДИСЦИПЛИНЫ
«Разработка компонентов системных программных продуктов»**

по основной профессиональной образовательной программе по направлению подготовки
09.03.01 «Информатика и вычислительная техника» (уровень бакалавриата)

Направленность (профиль): Программно-техническое обеспечение автоматизированных систем
Общий объем дисциплины – 12 з.е. (432 часов)

В результате освоения дисциплины у обучающихся должны быть сформированы компетенции с соответствующими индикаторами их достижения:

- ПК-4.1: Применяет языки программирования низкого уровня для написания кода компонентов системных программных продуктов и осуществляет его отладку;
- ПК-4.2: Демонстрирует знание принципов компиляции, интерпретации и создания исполняемого кода;
- ПК-4.3: Применяет языки высокого уровня для разработки компонентов системных программ;

Содержание дисциплины:

Дисциплина «Разработка компонентов системных программных продуктов» включает в себя следующие разделы:

Форма обучения очная. Семестр 6.

Объем дисциплины в семестре – 4 з.е. (144 часов)

Форма промежуточной аттестации – Зачет

1. Системное программное обеспечение для взаимодействия с периферийными устройствами.

Обработка прерываний в многопроцессорной вычислительной системе. Особенности низкоуровневого кода обработчика прерываний..

2. Низкоуровневое программирование в операционной среде. Программная обработка событий, связанных с мышью и клавиатурой..

3. Системные функции для работы с графическими устройствами. Программирование работы с графическими устройствами в приложениях..

4. Системные структуры данных и алгоритмы обмена данными с устройствами хранения данных. Программирование синхронного и асинхронного обмена данными с устройствами хранения данных...

5. Драйвер устройства как системный программный продукт (DLL).Схема обработки запроса на ввод-вывод.Многоуровневость драйверов.Основные функции, включаемые в драйвер. Объекты «файл», «драйвер» и «устройство». Понятие IRP. Обработка запросов ввода-вывода одноуровневым и многоуровневым драйверами.Программирование драйверов..

Форма обучения очная. Семестр 7.

Объем дисциплины в семестре – 5 з.е. (180 часов)

Форма промежуточной аттестации – Экзамен

1. Основные понятия теории формальных языков как основа построения компилятора. Концепция порождения и распознавания.. Алфавит. Формальные языки. Операции над языками. Порождающая грамматика. Понятие вывода. Классификация грамматик по Хомскому..

2. Контекстно-свободные языки и их порождение. Понятие грамматического разбора. Преобразование КС-грамматик.. Основные свойства контекстно-свободных языков. Замкнутость класса КС-языков. Грамматический разбор. Однозначные КС-грамматики. Приведение КС-грамматики к нормальной форме. Удаление бесполезных нетерминалов. Удаление эпсилон-правил..

3. Системное программное обеспечение для компиляции и интерпретации. Общая схема компилятора.. Трансляция, компиляция и интерпретация языковых конструкций.Этапы разработки компонентов системных программных продуктов, осуществляющих трансляцию и интерпретацию.

4. Лексический анализ. Лексемы языков программирования.. Построение грамматики модельного языка. Разработка на языке высокого уровня лексического анализатора как

компонента компилятора..

5. Синтаксический анализ.. Метод рекурсивного спуска. Построение, преобразование синтаксических диаграмм. Разметка ветвей диаграмм. Функции first, follow. Программирование синтаксических диаграмм на языке высокого уровня, используя современные инструментальные средства и технологии программирования..

Форма обучения очная. Семестр 8.

Объем дисциплины в семестре – 3 з.е. (108 часов)

Форма промежуточной аттестации – Экзамен

1. Семантический анализ. Нейтрализация ошибок.. Контекстные условия. Организация хранения семантической информации. Проектирование и особенности разработки на языке высокого уровня компонентов программного обеспечения для семантического анализа языковых конструкций. Принципы нейтрализации ошибок..

2. Системное программное обеспечение. Интерпретация.. Принципы работы интерпретатора. Интерпретация языковых конструкций различных типов. Разметка синтаксических диаграмм для интерпретации. Интерпретатор как компонент программных комплексов, осуществляющих лингвистический анализ..

3. Блок синтеза компилятора. Синтез языковых конструкций.. Понятие внутреннего кода компилятора. Способы представления дерева разбора. Понятие синтаксически управляемого перевода..

4. Оптимизация внутреннего кода компилятора. Преобразование в машинный код и загрузка.. Способы оптимизации внутреннего кода. Понятие графа управления программы. Оптимизация на линейных участках. Вынесение инвариантных триад из разветвлений и циклов. Формирование машинного кода. Загрузчик..

Разработал:
профессор
кафедры ИВТиИБ

Л.И. Сучкова

Проверил:
Декан ФИТ

А.С. Авдеев