

**Аннотация к рабочей программе  
дисциплины «Системное программное обеспечение»  
по направлению 09.03.01 Информатика и вычислительная техника  
профиль «Автоматизированные системы обработки информации и управления»**

**Общая трудоемкость дисциплины** составляет 7 зачетных единиц (252 часов)

**Форма контроля:** зачет, экзамен.

**Предполагаемые семестры:** 4, 5.

**Цель** преподавания дисциплины – обучение студентов принципам построения основных компонентов системного программного обеспечения и практическим навыкам работы с некоторыми из них.

**Задачи** преподавания дисциплины: формирование систематизированного представления о концепциях и моделях, положенных в основу построения компонентов системного программного обеспечения; получение практической подготовки в области выбора и применения компонентов системного программного обеспечения для задач автоматизации обработки информации и управления.

Учебная дисциплина входит в профессиональный цикл дисциплин (дисциплина по выбору).

**Знания**, полученные по дисциплине Системное программное обеспечение, используются при изучении

**Краткое содержание дисциплины:**

Цели и задачи дисциплины. Место и роль дисциплины в общей структуре подготовки специалистов, связь с другими дисциплинами. Назначение и функции системного программного обеспечения (СПО). Роль и место СПО в архитектуре вычислительных систем. Требования, предъявляемые к СПО.

Пользовательский интерфейс операционной среды. Командный интерфейс. Оконный интерфейс (WIMP-интерфейс). Речевой интерфейс (SILK-интерфейс).

Управление задачами. Очередь готовых задач. Приоритеты задач: статические и динамические. Основные алгоритмы планирования задач: алгоритмы, основанные на квантовании (разделения времени), алгоритмы, основанные на приоритетах (невывесняющая и вытесняющая многозадачность), комбинированные алгоритмы.

Взаимосвязанные задачи. Способы их коммуникации: сообщения, почтовые ящики. Конкурирующие задачи. Проблемы синхронизации: тупики, инверсия приоритетов, гонки. Понятие критической секции. Способы синхронизации: семафоры, исключающие семафоры (мутексы), мониторы.

Управление памятью. Функции ОС по управлению памятью. Типы адресов: виртуальные и физические. Методы распределения памяти без использования дискового пространства. Мультипрограммирование с фиксированными разделами. Мультипрограммирование с переменными разделами. Фрагментация памяти. Перемещаемые разделы. Методы распределения памяти с использованием дискового пространства. Понятие виртуальной памяти. Страничное распределение. Сегментное распределение. Странично-сегментное распределение. Механизмы реализации виртуальной памяти. Свопинг. Стратегии управления виртуальной памятью. Иерархия запоминающих устройств. Принцип кэширования данных. Кэш-память. Многопроцессорный режим работы. Совместное использование памяти. Защита памяти.

Управление вводом–выводом. Организация устройств ввода – вывода. Программное обеспечение ввода – вывода: обработчики прерываний, драйверы устройств, независимый от устройств слой ОС, пользовательский слой. Спулинг.

Управление файлами. Имена файлов. Типы файлов: обычные файлы, специальные файлы, каталоги. Логическая организация файла. Физическая организация и адрес файла. Дескриптор файла. Управление доступом к файлам и каталогам. Защита от несанкционированного доступа. Многоуровневая модель файловой системы. Отображение файлов в адресное пространство выполняемого процесса. Архитектура современной файловой системы.

Структура ОС. Назначения и функции ее компонентов. Реализация прикладных сред. Особенности интерфейса. Алгоритмы планирования процессов и потоков. Реализация многозадачности. Управление памятью. Организация файловой системы.

Формальные системы и языки программирования. Примеры формальных систем. Использование формальных систем в языках программирования. Формальное описание языка. Формальные грамматики. Нотация Бэкуса – Наура. Формальные языки, их классификация. Пример описания синтаксиса языка.

Ассемблеры. Основные функции ассемблера. Форматы машинных команд. Способы адресации. Перемещение программ. Литералы. Средства определения имен. Выражения. Команды ассемблера. Управляющие секции и связывание программ. Варианты построения ассемблеров: однопросмотровые и многопросмотровые ассемблеры. Примеры реализации ассемблеров.

Линкеры. Основные функции линкера. Перемещение программ. Связывание программ. Автоматический поиск в библиотеках. Управление процессом загрузки. Основные варианты построения линкеров, примеры реализации.

Макропроцессоры. Особенности макроязыков. Макроопределения и макрорасширения. Макровыводы внутри макроопределений. Конкатенация макропараметров. Генерация уникальных меток. Условные макрорасширения. Ключевые макропараметры. Варианты построения макропроцессоров: макропроцессоры общего назначения; макропроцессоры, встроенные в компиляторы. Примеры реализации макропроцессоров.

Компиляторы. Основные функции компилятора: лексический анализ, синтаксический анализ, генерация кода. Особенности их реализации. Оптимизация кода. Варианты построения компиляторов. Интерпретаторы. Компиляторы на псевдокод. Пример использования: Java–машина.

Средства автоматизации разработки компиляторов (компиляторы компиляторов). Примеры реализации компиляторов и интерпретаторов.

Системы интерактивной отладки программ. Функции и возможности отладчиков. Пошаговый режим работы. Режим трассировки выполнения программ. Использование контрольных точек. Доступ к объектам программы. Примеры реализации отладчиков.

**В результате изучения дисциплины бакалавр должен обладать следующими профессиональными компетенциями:**

ОПК-1: владеет способностью устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем.

В результате изучения дисциплины бакалавр должен:

**Знать:**

1. машино-зависимые и машино-независимые свойства ассемблеров, загрузчиков, макропроцессоров, компиляторов и операционных систем;
2. функции Win32API;
3. алгоритмы выполнения типичных представителей системных программ ассемблеров, загрузчиков, макропроцессоров, компиляторов и операционных систем;
4. структуры данных, используемых типичными представителями системных программ: однопросмотровыми, двухпросмотровыми и многопросмотровыми ассемблерами, абсолютными, перемещающими, связывающими загрузчиками, редакторами связей, макропроцессорами, компиляторами и современными операционными системами;
5. требования к программам на языке ассемблера, разрабатываемым для операционных систем Windows 98 (Windows 2000), Linux и MS-DOS;
6. современные тенденции развития информатики и вычислительной техники, компьютерных технологий;
7. современные программные средства взаимодействия с ЭВМ;
8. особенности работы ЭВМ в различных режимах (реальный и защищенный режимы работы процессора).

**Уметь:**

1. отнести системную программу к соответствующему классу (по принятой классификации), выбрать нужную в конкретном случае системную программу;
2. спроектировать системную программу;
3. использовать объекты ядра операционной системы, процессы и потоки при разработке приложений;
4. использовать функции Win32API и алгоритм распределения процессорного времени между потоками, применяемый в операционной системе;
5. разрабатывать программы на языке ассемблера для операционных систем Windows 98 (Windows 2000), Linux и MS-DOS (приложения, системные программы, драйверы);
6. применять математические методы, физические законы и вычислительную технику для решения практических задач;
7. устанавливать, тестировать, испытывать и использовать программные средства вычислительных и информационных систем.

**Владеть:**

1. специальной терминологией и лексикой данной дисциплины как минимум на одном иностранном языке;
2. навыками самостоятельного овладения новыми знаниями в области развития теории и практики эксплуатации.