

## **Вопросы для проведения экзамена по дисциплине « Современные средства программирования »**

### **Часть I. Операционные системы и среды**

#### **Основные понятия**

1. Понятие операционной среды
2. Понятия вычислительного процесса и ресурса  
    Диаграмма состояний процесса  
    Реализация понятия последовательного процесса в ОС
3. Процессы и треды(нити)
4. Прерывания
5. Основные виды ресурсов
6. Классификация операционных систем

#### **Управление задачами и памятью в операционных системах**

7. Планирование и диспетчеризация процессов и задач  
    Стратегии планирования  
    Дисциплины диспетчеризации  
    Качество диспетчеризации и гарантии обслуживания  
    Диспетчеризация задач с использованием динамических приоритетов
8. Память и отображения, виртуальное адресное пространство
9. Простое непрерывное распределение и распределение с перекрытием (оверлейные структуры)
10. Распределение статическими и динамическими разделами  
    Разделы с фиксированными границами  
    Разделы с подвижными границами
11. Сегментная, страничная и сегментно-страничная организация памяти  
    Сегментный способ организации виртуальной памяти  
    Страничный способ организации виртуальной памяти  
    Сегментно-страничный способ организации виртуальной памяти
12. Распределение оперативной памяти в современных ОС для ПК  
    Распределение оперативной памяти в MS-DOS  
    Распределение оперативной памяти в Microsoft Windows 95/98  
    Распределение оперативной памяти в Microsoft Windows NT
13. Реальный и защищенный режимы работы процессора
14. Системные регистры микропроцессоров
15. Адресация в 32-разрядных микропроцессорах при работе в защищенном режиме  
    Поддержка сегментного способа организации виртуальной памяти  
    Поддержка страничного способа организации виртуальной памяти  
    Режим виртуальных машин для исполнения приложений реального режима
16. Защита адресного пространства задач  
    Уровни привилегий для защиты адресного пространства задач  
    Механизм шлюзов для передачи управления на сегменты кода с другими уровнями привилегий
17. Система прерываний микропроцессоров  
    Работа системы прерываний в реальном режиме работы процессора  
    Работа системы прерываний в защищенном режиме работы процессора

#### **Управление вводом/выводом и файловые системы**

18. Основные понятия и концепции организации ввода/вывода в ОС
19. Режимы управления вводом/выводом
20. Закрепление устройств, общие устройства ввода/вывода
21. Основные системные таблицы ввода/вывода
22. Синхронный и асинхронный ввод/вывод
23. Кэширование операций ввода/вывода при работе с накопителями на магнитных дисках

24. Функции файловой системы ОС и иерархия данных
25. Структура магнитного диска (разбиение дисков на разделы)
26. Файловая система FAT
  - Таблица размещения файлов
  - Структура загрузочной записи DOS
  - Файловые системы VFAT и FAT32
27. Файловая система HPFS
28. Файловая система NTFS (New Technology File System)
  - Основные возможности файловой системы NTFS
  - Структура тома с файловой системой NTFS
  - Возможности файловой системы NTFS по ограничению доступа к файлам и каталогам
  - Основные отличия FAT и NTFS

### **Архитектура операционных систем и интерфейсы прикладного программирования**

29. Основные принципы построения операционных систем
  - Принцип модульности
  - Принцип функциональной избирательности
  - Принцип генерируемости ОС
  - Принцип функциональной избыточности
  - Принцип виртуализации
  - Принцип независимости программ от внешних устройств
  - Принцип совместимости
  - Принцип открытой и наращиваемой ОС
  - Принцип мобильности (переносимости)
  - Принцип обеспечения безопасности вычислений
30. Микроядерные операционные системы
31. Монолитные операционные системы
32. Требования, предъявляемые к ОС реального времени
  - Мультипрограммность и многозадачность
  - Приоритеты задач (поток)
  - Наследование приоритетов
  - Синхронизация процессов и задач
  - Предсказуемость
33. Принципы построения интерфейсов операционных систем
34. Интерфейс прикладного программирования
  - Реализация функций API на уровне ОС
  - Реализация функций API на уровне системы программирования
  - Реализация функций API с помощью внешних библиотек
35. Платформенно-независимый интерфейс POSIX
36. Пример программирования в различных API ОС (Здесь можно сделать следующим образом: поставить тривиальную задачу, например написать программу для открывания CD привода по нажатию кнопки или команде из консоли, и студент должен будет объяснить через что это делается. Сам код не обязательно)
  - Текст программы для Windows (WinAPI)
  - Текст программы для Linux (POSIX API)

### **Проектирование параллельных взаимодействующих вычислительных процессов**

37. Независимые и взаимодействующие вычислительные процессы
38. Средства синхронизации и связи при проектировании взаимодействующих вычислительных процессов
  - Использование блокировки памяти при синхронизации параллельных процессов
  - Синхронизация процессов посредством операции "ПРОВЕРКА И УСТАНОВКА"
  - Семафорные примитивы Дейкстры
  - Использование семафоров при проектировании взаимодействующих вычислительных процессов
39. Мониторы Хоара
40. Почтовые ящики
41. Конвейеры и очереди сообщений
  - Конвейеры (программные каналы)
  - Очереди сообщений
42. Примеры создания параллельных взаимодействующих вычислительных процессов

Пример создания многозадачного приложения с помощью системы программирования Borland Delphi  
Пример создания комплекса параллельных взаимодействующих программ, выступающих как самостоятельные вычислительные процессы

### **Проблема тупиков и методы борьбы с ними**

- 43. Понятие тупиковой ситуации при выполнении параллельных вычислительных процессов
- 44. Примеры тупиковых ситуаций и причины их возникновения
  - Пример тупика на ресурсах типа CR
  - Пример тупика на ресурсах типа CR и SR
  - Пример тупика на ресурсах типа SR
- 45. Формальные модели для изучения проблемы тупиковых ситуаций
  - Сети Петри
  - Вычислительные схемы
  - Модель пространства состояний системы
- 46. Методы борьбы с тупиками
  - Предотвращение тупиков
  - Обход тупиков
  - Обнаружение тупика

### **Современные операционные системы**

- 47. Семейство операционных систем UNIX
  - Общая характеристика семейства операционных систем UNIX, особенности архитектуры семейства ОС UNIX
  - Основные понятия системы UNIX
  - Функционирование системы UNIX
  - Файловая система
  - Межпроцессные коммуникации в UNIX
  - Операционная система Linux
- 48. Семейство операционных систем OS/2 Warp компании IBM
  - Особенности архитектуры и основные возможности OS/2 Warp
  - Особенности интерфейса OS/2 Warp
  - Серверная операционная система OS/2 Warp 4.5
- 49. Сетевая ОС реального времени QNX
  - Архитектура системы QNX
  - Основные механизмы QNX для организации распределенных вычислений

## **Часть II. Трансляторы, формальные языки и грамматики**

### **Формальные языки и грамматики**

- 1. Языки и цепочки символов. Способы задания языков
  - Цепочки символов. Операции над цепочками символов
  - Понятие языка. Формальное определение языка
  - Способы задания языков
  - Синтаксис и семантика языка
  - Особенности языков программирования
- 2. Определение грамматики. Форма Бэкуса--Наура
  - Понятие о грамматике языка
  - Формальное определение грамматики. Форма Бэкуса--Наура
  - Принцип рекурсии в правилах грамматики
  - Другие способы задания грамматик
- 3. Классификация языков и грамматик
  - Классификация грамматик. Четыре типа грамматик по Хомскому
  - Классификация языков
  - Примеры классификации языков и грамматик
- 4. Цепочки вывода. Сентенциальная форма

- Вывод. Цепочки вывода
- Сентенциальная форма грамматики. Язык, заданный грамматикой
- Левосторонний и правосторонний выводы
- Дерево вывода. Методы построения дерева вывода
- 5. Проблемы однозначности и эквивалентности грамматик
  - Однозначные и неоднозначные грамматики
  - Эквивалентность и преобразование грамматик
  - Правила, задающие неоднозначность в грамматиках
- 6. Распознаватели. Задача разбора
  - Общая схема распознавателя
  - Виды распознавателей
  - Классификация распознавателей по типам языков
  - Задача разбора (постановка задачи)

## Регулярные языки

- 7. Регулярные языки и грамматики
  - Левосторонние и правосторонние грамматики. Автоматные грамматики
  - Алгоритм преобразования регулярной грамматики к автоматному виду
  - Пример преобразования регулярной грамматики к автоматному виду
- 8. Конечные автоматы
  - Определение конечного автомата
  - Детерминированные и недетерминированные конечные автоматы
  - Преобразование конечного автомата к детерминированному виду
  - Минимизация конечных автоматов
- 9. Регулярные множества и регулярные выражения
  - Определение регулярного множества
  - Регулярные выражения. Свойства регулярных выражений
  - Уравнения с регулярными коэффициентами
- 10. Способы задания регулярных языков
  - Три способа задания регулярных языков
  - Связь регулярных выражений и регулярных грамматик
  - Связь регулярных выражений и конечных автоматов
  - Связь регулярных грамматик и конечных автоматов
  - Пример построения конечного автомата на основе заданной грамматики
- 11. Свойства регулярных языков
  - Свойства регулярных языков
  - Лемма о разрастании для регулярных языков

## Контекстно-свободные языки

- 12. Распознаватели КС-языков. Автоматы с магазинной памятью
  - Определение МП-автомата
  - Эквивалентность языков МП-автоматов и КС-грамматик
  - Детерминированные МП-автоматы
- 13. Свойства КС-языков
  - Свойства произвольных КС-языков
  - Свойства детерминированных КС-языков
  - Лемма о разрастании КС-языков
- 14. Преобразование КС-грамматик. Приведенные грамматики
  - Преобразование грамматик. Цель преобразования.
  - Приведенные грамматики
  - Удаление недостижимых символов
  - Удаление бесплодных символов
  - Устранение ? -правил
  - Устранение цепных правил
- 15. КС-грамматики в нормальной форме
  - Граматики в нормальной форме Хомского
  - Устранение левой рекурсии. Граматики в нормальной форме Грейбах
- 16. Распознаватели КС-языков с возвратом

- Принципы работы распознавателей с возвратом
- Нисходящий распознаватель с возвратом
- Распознаватель на основе алгоритма "сдвиг-свертка"
- 17. Табличные распознаватели для КС-языков
  - Общие принципы работы табличных распознавателей
  - Алгоритм Кока--Янгера--Касами
  - Алгоритм Эрли (основные принципы)
- 18. Принципы построения распознавателей КС-языков без возвратов

### **Классы КС-языков и грамматик**

- 19. Нисходящие распознаватели КС-языков без возвратов
  - Левосторонний разбор по методу рекурсивного спуска
  - Определение LL(k)-грамматики
  - Принципы построения распознавателей для LL(k)-грамматик
  - Алгоритм разбора для LL(1)-грамматик
- 20. Восходящие распознаватели КС-языков без возвратов
  - Определение LR(k)-грамматики
  - Принципы построения распознавателей для LR(k)-грамматик
  - Граматики предшествования (основные принципы)
  - Граматики простого предшествования
  - Граматики операторного предшествования
- 21. Соотношение классов КС-языков и КС-грамматик
  - Особенности восходящих и нисходящих распознавателей
  - Отношения между классами КС-грамматик
  - Отношения между классами КС-языков

### **Основные принципы построения трансляторов**

- 22. Трансляторы, компиляторы и интерпретаторы -- общая схема работы
  - Определение транслятора, компилятора, интерпретатора
  - Этапы трансляции. Общая схема работы транслятора
  - Понятие прохода. Многопроходные и однопроходные компиляторы
  - Интерпретаторы. Особенности построения интерпретаторов
  - Трансляторы с языка ассемблера ("ассемблеры")
- 23. Таблицы идентификаторов. Организация таблиц идентификаторов
  - Назначение и особенности построения таблиц идентификаторов
  - Простейшие методы построения таблиц идентификаторов
  - Построение таблиц идентификаторов по методу бинарного дерева
  - Хэш-функции и хэш-адресация
  - Комбинированные способы построения таблиц идентификаторов
- 24. Лексические анализаторы (сканеры). Принципы построения сканеров
  - Назначение лексического анализатора
  - Принципы построения лексических анализаторов
  - Построение лексических анализаторов
  - Автоматизация построения лексических анализаторов (программа LEX)
- 25. Синтаксические анализаторы. Синтаксически управляемый перевод
  - Основные принципы работы синтаксического анализатора
  - Дерево разбора. Преобразование дерева разбора в дерево операций
  - Автоматизация построения синтаксических анализаторов (программа YACC)

### **Генерация и оптимизация кода**

- 26. Семантический анализ и подготовка к генерации кода
  - Назначение семантического анализа
  - Этапы семантического анализа
  - Идентификация лексических единиц языков программирования
  - Распределение памяти. Принципы распределения памяти
  - Дисплей памяти процедуры (функции). Стековая организация дисплея памяти

- Память для типов данных (RTTI-информация)
- 27. Генерация кода. Методы генерации кода
  - Общие принципы генерации кода. Синтаксически управляемый перевод
  - Способы внутреннего представления программ
  - Обратная польская запись операций
  - Схемы СУ-перевода
- 28. Оптимизация кода. Основные методы оптимизации
  - Общие принципы оптимизации кода
  - Оптимизация линейных участков программы
  - Другие методы оптимизации программ
  - Машинно-зависимые методы оптимизации

### **Современные системы программирования**

- 29. Понятие и структура системы программирования
  - История возникновения систем программирования
  - Структура современной системы программирования
- 30. Принципы функционирования систем программирования
  - Функции текстовых редакторов в системах программирования
  - Компилятор как составная часть системы программирования
  - Компоновщик. Назначение и функции компоновщика
  - Загрузчики и отладчики. Функции загрузчика
  - Библиотеки подпрограмм как составная часть систем программирования
- 31. Дополнительные возможности систем программирования
  - Лексический анализ "на лету". Система подсказок и справок
  - Разработка программ в архитектуре "клиент--сервер"
  - Разработка программ в трехуровневой архитектуре. Серверы приложений
- 32. Примеры современных систем программирования
  - Системы программирования компании Borland/Inprise
  - Системы программирования фирмы Microsoft
  - Системы программирования под ОС Linux и UNIX
  - Разработка программного обеспечения для сети Интернет