

# **Темы к экзамену по дисциплине**

## **«Операционные системы и архитектура компьютера».**

**весенний семестр 2023-2024 гг.**

1. Модель событийного программирования. Концепция функций обратного вызова. Некоторые приемы установления ассоциаций между данными и методами их обработки.
2. Понятие процесса Unix System V. Создание процесса: функция fork. Подмена образа исполняемого процесса: семейство функции exec.
3. Межпроцессное взаимодействие. Проблема читателей и писателей. Проблемы грязного чтения и грязной записи; их вариации.
4. Синхронизация процессов с помощью очередей сообщений. API для взаимодействия с очередями сообщения Unix System V/Windows.
5. Синхронизация процессов с помощью семафоров. API для взаимодействия с семафорами Unix System V/Windows.
6. Синхронизация процессов с помощью разделяемой памяти. API для взаимодействия с разделяемой памятью Unix System V/Windows.
7. Алгоритм синхронного доступа к разделяемой памяти с помощью семафоров. Реализация клиент-серверного комплекса приложений посредством семафоров и разделяемой памяти на языке C.
8. Синхронизация процессов с помощью файлов, проецируемых в память. API для взаимодействия с файлами, проецируемыми в память Unix System V/Windows.
9. Алгоритмы управления динамической памятью. Метод первого подходящего. Метод лучшего подходящего. Метод худшего подходящего. Метод освобождения в рассортированном списке. Метод освобождения с дескрипторами границ. Метод система двойников. Метод, организующий список всех и красно-чёрное дерево свободных блоков.

10. Сбалансированные по высоте деревья поиска. Операции малого левого и малого правого поворота. Операции двойного малого левого и двойного малого правого поворота. Операции большого левого и большого правого поворота. В-дерево,  $B^+$ -дерево,  $B^*$ -дерево,  $B^{*+}$ -дерево: операции добавления, удаления, поиска элемента по ключу; поиска в диапазоне ключей.
11. Язык программирования C++. Понятия класса и объекта. Поля и методы классов. Конструкторы, деструкторы. Конструктор со списком инициализации. Ключевое слово `explicit`. Ключевое слово `const` (применительно к методам класса), особенности. Ключевое слово `this`.
12. Язык программирования C++. Инициализация и присваивание. Конструктор копирования. Оператор присваивания. Деструктор. Конструктор перемещения. Присваивание перемещением. Правило трёх. Правило пяти. Особенности реализации методов правила трёх/пяти.
13. Инкапсуляция в языке программирования C++. Модификаторы доступа `public`, `private`, `protected`, назначение.
14. Наследование в языке программирования C++. Наследование вида “быть”, наследование вида “иметь”. Виртуальные методы, чистые виртуальные методы. Понятие интерфейса. Проблемы множественного наследования, “алмаз смерти”. Виртуальный деструктор. Запечатанные классы. Ключевые слова `virtual`, `override`. Модификаторы `public`, `private`, `protected` при наследовании, назначение и особенности.
15. Полиморфизм в языке программирования C++. Таблица виртуальных методов. Понятие перегрузки и переопределения методов, различия.
16. Контексты определения идентификаторов в языке программирования C++. Ключевое слово `namespace`. Оператор расширения области видимости. Вложенные типы, назначение. Ключевое слово `static`.

17. Обработка ошибок времени выполнения в языке программирования C++. Понятие исключительной ситуации. Ключевое слово `noexcept`. Тип `std::exception`. Ключевые слова `try`, `catch`, `throw`. Раскрутка стека. Особенности, возникающие при реализации обработки исключительных ситуаций.
18. Выделение и освобождение динамической памяти в языке программирования C++. Операторы `new`, `new[]`, `delete`, `delete[]`, `::operator new`, `::operator delete`; особенности при использовании. Тип `std::bad_alloc`.
19. Операторы приведения типов в языке программирования C++. Пользовательские операторы явного и неявного приведения типов. Операторы `static_cast<type>(expr)`, `dynamic_cast<type>(expr)`, `const_cast<type>(expr)`, `reinterpret_cast<type>(expr)`, различие, особенности.
20. Перегрузка операторов в языке программирования C++. Перегрузка унарных и бинарных операторов: арифметических операторов, индексирующего, операторов сравнения, операторов отношения порядка. Особенности перегрузки операторов инкремента и декремента. Особенности перегрузки операторов вставки в поток и выгрузки из потока. Понятия дружественных функций и дружественных классов.
21. Родовые функции и родовые классы в языке программирования C++, назначение. Ключевые слова `template`, `typename`. Конкретизация параметров типа родовых функций и классов. Особенности кодогенерации на этапе компиляции при конкретизации параметров типа родовых функций и классов.
22. Библиотека стандартных шаблонов C++ (standard template library): контейнеры. Понятие контейнера. Контейнеры `std::array`, `std::vector`, `std::forward_list`, `std::list`, `std::deque`, `std::set`, `std::unordered_set`, `std::map`, `std::unordered_map`; адаптеры контейнеров `std::queue`, `std::stack`, `std::priority_queue`; назначение, особенности внутреннего устройства, примеры использования.
23. Библиотека стандартных шаблонов C++ (standard template library): итераторы. Понятие итератора. Назначение, особенности внутреннего устройства. Последовательные итераторы, двунаправленные итераторы, итераторы произвольного доступа.

24. Библиотека стандартных шаблонов C++ (standard template library): умные указатели. Типы `std::unique_ptr`, `std::weak_ptr`, `std::shared_ptr`; назначение, особенности внутреннего устройства, примеры использования.

25. Понятие транзакции. Набор требований ACID. Уровни изоляции транзакций.