

Вопросы к экзамену (осень 2025)

1. Промежуточные представления программы верхнего, среднего и нижнего уровней. Описание промежуточного представления среднего уровня. Генерация промежуточного представления среднего уровня.
2. Базовые блоки и граф потока управления. Алгоритм построения графа потока управления.
3. Локальная оптимизация. Ориентированный ациклический граф и алгоритм его построения. Оптимизирующие преобразования, выполняемые во время локальной оптимизации.
4. Восстановление базового блока по его ориентированному ациклическому графу.
5. Нумерация вершин графа потока управления. Как она выполняется (алгоритм) и для чего она нужна.
6. Поток данных. Точки программы, состояния программы, передаточная функция (определение). Передаточная функция базового блока.
7. Достигающие определения. Передаточная функция для анализа достигающих определений и ее свойства. Для чего нужно вычислять достигающие определения.
8. Система уравнений для вычисления достигающих определений и итерационный алгоритм ее решения. Пример применения алгоритма.
9. Живые переменные. Передаточная функция для анализа живых переменных и ее свойства.
10. Система уравнений для вычисления живых переменных и итерационный алгоритм ее решения. Пример применения алгоритма.
11. Множества *Input* и *Output* базового блока и их вычисление.
12. Доступные выражения и исключение избыточных вычислений. Передаточная функция для анализа доступных выражений и ее свойства.
13. Система уравнений для вычисления доступных выражений и итерационный алгоритм ее решения. Пример применения алгоритма.
14. Определение полурешетки с операцией «сбор». Полурешеточное отношение частичного порядка (\leq) и его свойства.
15. Верхний элемент полурешетки и его свойства. Наибольшая нижняя граница двух элементов полурешетки и ее свойства. Диаграмма полу решетки.

16. Структура потока данных. Примеры структур потока данных.
17. Замкнутость семейств передаточных функций, используемых при анализе достигающих определений, живых переменных и доступных выражений.
18. Монотонные структуры анализа потока данных. Два определения монотонности и их эквивалентность.
19. Дистрибутивные структуры анализа потока данных.
20. Доказательство дистрибутивности структуры достигающих определений.
21. Доказательство дистрибутивности структур живых переменных и доступных выражений.
22. Обобщенный итеративный алгоритм анализа потока данных.
23. Сходимость обобщенного итеративного алгоритма анализа потока данных к решению системы уравнений потока данных.
24. Максимальная фиксированная точка системы уравнений (определение). Доказательство сходимости решения системы уравнений для монотонной структуры потока данных к ее максимальной фиксированной точке.
25. Идеальное решение системы уравнений для монотонной структуры потока данных и его отношение к максимальной фиксированной точке.
26. Свойства решений уравнений потока данных для монотонных и дистрибутивных структур.
27. Решение сбором по всем путям системы уравнений для монотонной структуры потока данных и его отношение к максимальной фиксированной точке.
28. Решение сбором по всем путям системы уравнений для дистрибутивной структуры потока данных и его отношение к максимальной фиксированной точке.
29. Консервативность решения системы уравнений потока данных с помощью итеративного алгоритма для монотонных и дистрибутивных структур.
30. Доминаторы. Определение и алгоритм вычисления. Построение дерева доминаторов.
31. Постдоминаторы. Определение и алгоритм вычисления. Построение дерева постдоминаторов.

32. Свойства отношения доминирования.
33. Граница доминирования. Определение, свойства. Построение границ доминирования (алгоритм). Обратная граница доминирования.
34. Зависимость по управлению. Определение. Связь с обратной границей доминирования. Эквивалентность по управлению.
35. Простые оптимизации: сворачивание констант, алгебраические упрощения и перегруппировка, распространение копий.
36. Классификация ребер графа потока управления. Определение обратного ребра. Определение естественного цикла. Алгоритм построения естественного цикла по обратному ребру.
37. Выявление кода, инвариантного относительно цикла.
38. Алгоритм перемещения кода, инвариантного относительно цикла.
39. Индуктивные переменные. Выявление групп индуктивных переменных.
40. Использование индуктивных переменных для ускорения цикла путем удешевления операций.
41. Что такое бесполезный код. Алгоритм *Mark & Sweep* удаления бесполезного кода.
42. Алгоритм *Mark & Sweep* удаления недостижимого кода.
43. Оптимизация потока управления.
44. Форма статического единственного присваивания (SSA-форма). Определение, свойства. Определение ϕ -функции.
45. Базовый алгоритм построения максимальной SSA-формы.
46. Алгоритм построения частично усеченной SSA-формы. Постановка задачи.
47. Алгоритм построения частично усеченной SSA-формы. Размещение ϕ -функций.
48. Алгоритм построения частично усеченной SSA-формы. Переименование переменных.
49. Восстановление кода из SSA-формы. Замена ϕ -функций группами инструкций копирования. Исключение критических ребер.
50. Глобальное распространение констант. Передаточные функции структуры распространения констант. Монотонность структуры

распространения констант. Недистрибутивность структуры
распространения констант.

51. Алгоритм глобального распространения констант.
52. Нумерация значений в суперблоках и глобальная нумерация значений. Рекурсивный алгоритм глобальной нумерации значений.
53. Структурный анализ графа потока управления. Построение иерархии областей. Дерево управления. Несводимые графы потока управления.
54. Схема анализа потока данных на основе областей. Построение передаточных функций областей. Операции композиции, сбора и замыкания, и их свойства. Алгоритм анализа на основе областей.
55. Как реализуется вызов процедуры и каковы накладные расходы на вызов.
56. Открытая вставка процедуры в место ее вызова. Как она выполняется. Преимущества и недостатки. Какие вызовы следует заменять открытой вставкой. Когда следует выполнять открытую вставку.
57. Профилирование. Два метода: инструментирование и семплирование.
58. Машинно-ориентированная оптимизация. Характеристика объектного кода. Архитектура целевого процессора. Режимы адресации. Основные фазы генерации оптимизированного кода.
59. Выбор команд. Постановка задачи. Метод переписывания дерева. Схема трансляции дерева. Алгоритм генерации кода для размеченных деревьев выражений.
60. Вычисление выражений при недостаточном количестве регистров.
61. Генерация кода с использованием алгоритма динамического программирования.
62. Локальное распределение регистров. Deskрипторы регистров и переменных. Функция *getReg()* и ее реализация.
63. Глобальное распределение и назначение регистров. Интервалы жизни. Построение интервалов жизни, расщепление и слияние интервалов жизни. Граф конфликтов и его построение. Слив переменных.
64. Быстрый алгоритм распределения регистров – метод линейного сканирования.
65. Распределение регистров с помощью трехфазного алгоритма раскраски графа конфликтов.
66. Планирование кода. Постановка задачи. Ограничения. Аппаратные

возможности параллельного выполнения команд: конвейер обработки команд, команды и операции, выполнение операций «не по порядку», аппаратное переименование регистров.

67. Планирование кода. Модель процессора. Ресурсы процессора. Зависимости по данным. Зависимости по данным между итерациями цикла. Зависимости по управлению.

68. Планирование базовых блоков. Граф зависимостей по данным и его построение.

69. Планирование базовых блоков. Выбор приоритетного топологического порядка. Алгоритм планирования списка.

70. Стратегии глобального планирования. Пример глобального планирования. Перемещение кода вверх по пути управления. Перемещение кода вниз по пути управления.

71. Алгоритм глобального планирования кода на основе областей. Планирование с добавлением компенсирующего кода. Агрессивные алгоритмы перемещения кода.