

Индивидуальное задание №7 по курсу «Языки программирования и методы трансляции» Тема: «Простой компилятор»

Редакция 2022 г.

Варианты заданий

1. Компилятор языка «О» с операторами FOR и REPEAT
2. Компилятор языка «О» с оператором CASE.
3. Компилятор языка «О» с операторами LOOP и EXIT.
Следует учесть, что операторы LOOP могут быть вложенными, а оператор EXIT вне оператора LOOP не используется.
4. Компилятор языка «О» с линейными массивами.
5. Компилятор языка «О» с типом SHORTINT и необходимыми стандартными процедурами, а также с шестнадцатичными константами.
6. Компилятор языка «О» с типом CHAR и необходимыми стандартными процедурами.
7. Компилятор языка «О» с переменными логического типа но без логических операций.
8. *Компилятор языка «О» с логическими операциями (&, OR, ~), но без логических переменных.
9. *Компилятор языка «О» с константными выражениями общего вида.
Предусматривается возможность использования константных выражений общего вида (как в языке Оберон-2) везде, где разрешается константа. Вычисление константных выражений должно выполняться во время компиляции.
- ~~10. Реализация компилятора языка «О» на языке Оберон.—~~
- ~~11. Реализация компилятора языка «О» на языке Модула 2.—~~
- ~~12. Реализация компилятора языка «О» на языке Си.—~~
- ~~13. Реализация компилятора языка «О» на языке Ява.—~~
- ~~14. Реализация компилятора языка «О» на языке Си#.—~~
15. Реализация компилятора языка «О» на языке Ада.
16. Реализация компилятора языка «О» на языке Go.
17. Реализация компилятора языка «О» на языке Rust.
18. Реализация компилятора языка «О» на языке Lua.
19. Компилятор языка «О» с возможностью записи объектного кода в двоичный файл и файл псевдокода OVM. Автономная виртуальная машина.
Разрабатываются две программы, одна – компилятор (os.exe), в результате работы которого всегда создается двоичный файл команд (расширение .ovm) виртуальной машины, а по требованию (параметр командной строки) и файл на языке ассемблера виртуальной машины (расширение .oas). Вторая программа – виртуальная машина (ovm.exe), которая, принимая в качестве параметра файл кода, выполняет программу.

20. *Оптимизирующий компилятор языка «О».
Обеспечивается оптимизация генерируемого компилятором кода за счет устранения лишних команд загрузки, переходов, улучшения схемы генерации циклов и т.п.
21. ~~Оптимизация компилятора языка «О» по быстродействию.
Целью является максимально возможное ускорение работы самого компилятора за счет использования более совершенных алгоритмов работы с таблицами, оптимизации сканера и драйвера исходного текста. Для анализа быстродействия компилятора и выявления критических мест рекомендуется использовать программу профилировщик.~~
22. Компилятор языка «О» на основе табличного синтаксического анализатора.
23. Компилятор языка «О» в код оптимизированной виртуальной машины.
Конструируется виртуальная машина, обеспечивающая возможно большую компактность и быстродействие объектной программы, получаемой компиляцией с языка «О».
24. *Компилятор языка «О» в код виртуального RISC-компьютера.
Используется виртуальная машина, описанная в книге N.Wirth. Theory and Techniques of Compiler Construction. Addison-Wesley, 1996. (<https://people.inf.ethz.ch/wirth/CompilerConstruction/>). Описание виртуальной RISC-машины содержится в главе 9. Текст модуля виртуальной машины в файле RISC.MOD.
25. Конвертор¹ языка «О» в язык Паскаль.
Используется стандартный Паскаль. Применение диалектизмов не допускается.
26. Конвертор языка «О» в язык Си.
27. Конвертор языка «О» в язык Ява.
28. Конвертор языка «О» в язык Ада.
29. Конвертор языка «О» в язык Си#.
30. Конвертор языка «О» в язык Python.
31. Конвертор языка «О» в язык Go.
32. *Конвертор языка «О» в язык ассемблера x86.
33. *Компилятор языка «О» в код x86.
34. Компилятор языка «О» с улучшенной диагностикой ошибок.
При возникновении ошибки отмечается фрагмент программы, к которому относится сообщение (лексема, выражение).
35. Компилятор языка «О» с расширенной диагностикой ошибок.
Предусматривается выдача нескольких (не более заданного числа) сообщений об ошибках во время компиляции.

¹ Конвертор - это программа, преобразующая текст на одном языке программирования в эквивалентную программу на другом языке программирования. Конвертор должен выполнять исчерпывающий анализ входной программы: лексический, синтаксический, контекстный. Любая правильная программа на исходном языке должна быть преобразована в эквивалентную программу на выходном языке. Любая неправильная - отвергнута. Крайне желательно, чтобы результирующая программа была удобочитаема: структурный отступ, отсутствие избыточности, внятные обозначения.

36. *Отладчик для языка «О».

Предусматривается пошаговое выполнение программы на языке «О» при нажатии кнопки F8. Текст программы и значения всех переменных показываются на экране. Текущая выполняемая строка выделяется (цветом).

37. Компилятор языка «О» с типами REAL и INTEGER и необходимыми функциями.

Предусматривается модернизация виртуальной машины. Расширяется набор ее команд. Можно использовать четырехбайтовый формат слова; для представления типа INTEGER - четырехбайтовые целые, а типа REAL - четырехбайтовые вещественные.

38. Компилятор языка «О» с диагностикой ошибок во время выполнения.

Контролируются переполнения при выполнении арифметических операций, деление на ноль, переполнение стека. При возникновении ошибки во время выполнения на экран выводится номер строки исходного текста программы, в которой произошла ошибка, сама эта строка и сообщение о типе ошибки. Модернизируется модуль виртуальной машины без изменения ее системы команд.

39. Профилировщик-1.

После выполнения программы выводится ее исходный текст с указанием сколько раз при работе программы выполнялись операторы каждой строки. Возможна модернизация виртуальной машины без изменения ее системы команд.

40. Профилировщик-2.

После выполнения программы выводится ее исходный текст с указанием, сколько тактов виртуальной машины потрачено на выполнение каждой строки и общее число тактов на выполнение всей программы. Возможна модернизация виртуальной машины без изменения ее системы команд.

41. Профилировщик-3.

После выполнения программы выводится ее исходный текст с указанием сколько раз при работе программы выполнялся каждый ее оператор. Возможна модернизация виртуальной машины без изменения ее системы команд.

42. *Компилятор языка «О» в IL-код .NET.

43. *Транслятор языка «О» в IL-ассемблер .NET.

44. *Компилятор языка «О» в Wasm.

45. *Компилятор языка ««О» с процедурами».

46. *Компилятор подмножества языка Си, эквивалентного языку «О».

47. *Компилятор подмножества языка Ява, эквивалентного языку «О».

48. *Компилятор подмножества языка Си#, эквивалентного языку «О».

49. *Компилятор подмножества языка Go, эквивалентного языку «О».

50. *Компилятор подмножества языка Kotlin, эквивалентного языку «О».

51. *Компилятор подмножества языка Python, эквивалентного языку «О».

52. Интеграция виртуальной машины, реализованной на языке Си, с компилятором, написанном на Python'е.

53. *Компилятор языка «О» в LLVM.

54. Компилятор языка «о». Язык «о» — это модификация языка «О» в котором служебные слова записываются маленькими буквами, а использование точки с запятой устранено или минимизировано.

Знаком * отмечены более сложные варианты.

Любые расширения языка «О» выполняются в соответствии с правилами языка Оберон-2.

В заданиях 1-22 изменение виртуальной машины не предусматривается.

Использование языка ассемблера или фрагментов на языке ассемблера для написания компилятора или реализации виртуальной машины не предусматривается.

Минимальными требованиями при выдаче сообщений об ошибках во время компиляции являются: вывод целиком строки, содержащей ошибку, с указанием места ошибки с помощью стрелки; вывод сообщения об ошибке.

Если язык реализации не указан явно, то задания можно выполнять на любом (в разумных пределах) языке.