Codegen #4
20.03.19

Work flow

Origin

$ bin/coolc bool.cl

-> bool.s

$ bin/spim bool.s

-> exec

Your code

$ make cgen

$ ./mycoolc bool.cl

-> bool.s

$ bin/spim bool.s

-> exec

Agreement registers (include/emit.h)

- $zero – всегда нулевой регистр
- $a0 – аккумулятор
- $sp – указатель на стек
- $fp – указатель на фрейм
- $ra – адрес возврата из функции
- $t1 – temporary register
- $s0 – указатель на текущий обьект (SELF)
- $a1 – для аргументов функций (not use)

Mips assembler code

Код логически делится на 2 сегмента:

- “.data” - сегмент данных. Область Static Data - не содержит кода. У данных отсюда фиксированный адрес (может быть как r/o так и r/w)
  - “.globl SYM” - указывает, что SYM является глобальным и на него можно ссылкться из других файлов (здесь - trap.handler)
- “.text” - сегмент кода. Область кода - содержит бинарный код. Для большинства языков размер фиксирован и область read-only.

Repo

- Основной файл – cgen.cc со всей логикой разделен на 3 файла:
  - cgen.cc – методы code(генерация кода) и calc_temp(подсчет количества переменных) для каждой ноды дерева разбора программы (tree_node)
  - cgen_class_table.cxx – содержит определения методов для ClassTable и CgenNode
  - cgen_static_init.cxx – содержит статические функции для работы с ассемблером(emit)

CgenClassTable & CgenNode

- CgenClassTable – класс со списком всех классов по имени (наследуется от SymbolTable< Symbol, CgenNode >) - работа с ним ведется через конструктор и метод code()
- CgenNode – один узел для описания и генерации полей класса
  - typedef CgenNode *CgenNodeP;

Конструктор CgenClassTable

- install_basic_classes():
  - Заполнение таблицы для работы с классами: No_class, SELF_TYPE, prim_slot
  - Создание нод (CgenNode) базовых класов : Object, IO, Int, Bool, Str
- install_classes():
  - Создание нод для всех пользовательских классов
  - Выставление currclasstag для всех классов – то есть у всех классов появляются идентификаторы.
- build_inheritance_tree():
  - Создание дерева наследования классов

	CgenClassTable::Code()
INFO_IN;	Макрос дебажной печати.
code_global_data();	Код глобальных ссылок на функции
code_select_gc();	Код для инициализации Garbage collector Здесь можно поменять алгоритмы и поведение gc для запускаемой программы
code_constants();	Генерация описания всех констант программы
code_classNameTab();	Генерация таблицы с именами классов
code_classObjTab();	Генерация таблицы со ссылками на функции прототипов и инициализации
code_dispTabs();	Генерация таблицы методов классов
code_protObjs();	Генерация прототипов обьектов
code_global_text();	Начало секции кода (text), указатель на начало кучи
code_initCode();	Генерация кода инициализации обьектов
code_methods();	**Генерация кода методов обьектов**

CgenClassTable::code_methods()

Для всех нод в списке – обход всех методов класса. Для каждого метода:

- Создание линка(LABEL) для метода
- Сохранение контекста и выделение памяти под локальные переменные
- Генерация expression метода (рекурентный вызов features[i]->code(...))

Глобальные обьекты cgen.cc

- “CgenClassTable**P** codegen_classtable” – текущая таблица классов
- “CgenNode**P** curr_node” - текущая исполняемая нода

Аттрибуты CgenNode

- “genNodeP parentnd” - указатель на родителя класса
- “List<CgenNode> *children” - список детей класса
- “Basicness basic_status” - признак базового обьекта (нужен для того, чтобы узнать, нужно ли кодировать методы)
- “int id” - идентификатор обьекта
- “Environment attrTable” - список аттрибутов класса с индексом
- “Environment methodTable” - список методов класса с оффсетами

Runtime – исполнение ассемблера

- При запуске происходят следующее:
  - В куче создается копия Main-обьекта (**прототип**). Затем инициализируется вызовом **Main_init**. Main_init должен запустить инициализацию **родительских** для Main классов и инициализировать **аттрибуты** Main-класса (если они существуют).
  - Далее управление передается в метод Main.main, при этом указатель на вновь созданный объект Main сохраняется в регистре $ a0. Регистр $ra содержит адрес возврата.
  - Если управление возвращается из Main.main, выполнение останавливается с сообщением «COOL program successfully executed».
- (см cool-toor.pdf : 7.5 Execution Startup)

QtSpim example

- → goto QtSpim lecture

- → codegen_3.pdf +17