

Call Stack

Реализация стека вызовов в интерпретаторе StudentVM

Структура данных, предлагаемая мной для организации вызова функций является стеком, в котором каждая функция представляется StackFrame'ом, имеющим данные поля:

1. char* Name - Хранит имя(идентификатор) функции;
2. int ReturnAddress - Адрес возврата
3. function* Last - Указатель на предыдущий элемент
4. LocalVar* Var Указатель на массив локальных переменных(в случае необходимости он может расширяться)
5. Argument* Arg Указатель на массив аргументов(ситуация аналогична предыдущему пункту).

Для удобства присутствует указатель на последний индекс, отвечающий за выполняемую в текущий момент функцию. Подобная структура была выбрана мной для избегания ситуации общей памяти функций с основной памятью, что могло приводить к неэкономному использованию памяти системы, в которой работает Виртуальная Машина.

Также можно добавить базовый StackFrame-main(), таким образом стек вызовов всегда не пуст.

Команды, необходимые для реализации:

1. cll — Вызов функции по имени
2. arg <num> - Загрузка на стек аргумента с номером «num»
3. ret — Удаление текущего StackFrame'a и переход на адрес возврата

Обмен данными между функциями происходит через общий стек и общую память.

Достоинства подобной структуры:

1. Экономия памяти
2. Расширяемость количества локальных переменных;
3. Возможность адресации аргументов.

