

Лабораторная работа №2. Исследование структуры таблиц ассемблера.

Цель работы: Исследование структуры таблиц ассемблера и алгоритмов работы с ними.

Содержание работы: В процессе выполнения лабораторной работы должны быть разработаны структуры таблиц ассемблера и алгоритмы, реализующие синтаксический разбор операторов исходной программы на языке ассемблера i8088** и заполнение таблиц символов и литералов.

В процессе трансляции исходной программы ассемблер работает с несколькими таблицами. Таблицы машинных операций (MOT) и псевдоопераций (POT) являются фиксированными и определяют структуру команд компьютера. Эти таблицы задаются (инициализируются) при запуске ассемблера. В процессе работы ассемблер обращается к MOT и POT для поиска машинных операций и псевдоопераций и определения их длин. В результате обработки исходной программы ассемблер строит таблицу символов (ST) и литералов (LT), которые используются для генерации машинных команд. Кроме того, ассемблер ведет счетчик адреса (LC), текущие значения которого заносятся в таблицы ST и LT. Таблица машинных операций MOT в общем случае содержит мнемонический код каждой команды и ее длину. Таблица псевдоопераций POT содержит мнемонический код каждой псевдооперации и информацию о действиях, которые должны выполняться для каждой псевдокоманды при ее обработке ассемблером.

Исходная программа на языке ассемблера состоит из операторов. Каждый оператор включает в себя поле метки, поле кода операции и поле адреса. Метка в поле метки не является обязательной. В поле кода операции находятся либо мнемонический код машинной команды, либо мнемонический код псевдооперации. Псевдооперациями считаются команды резервирования памяти, определения констант, объявления базисных регистров и т. п. Они задают операцию программы – ассемблера, в то время как машинные команды задают операции процессора (инструкции). Таким образом, в общем виде оператор языка ассемблера можно представить следующим образом:

<Метка> КОП <Адресная часть>.

Вид адресной части зависит от формата команды и может содержать только регистры общего назначения, регистр и адресуемую память, регистр и непосредственные данные (литералы), либо вообще не содержать адресной части. Ниже приведен пример простой программы (формально корректной!) на языке ассемблера i8088.

```
Title LIFT.ASM
LIFT Segment
Assume cs:LIFT,ds:LIFT
LIFTTO Proc Far
les si,SOURCE ;Загрузка адреса источника
mov cx,NUMB   ; Загрузка счетчика к-ва байт
mov ax,0      ;Заготовка смещения
mov ds,ax     ;в регистре ds
```

```

        mov bx,04f0h ;Формирование базисного адреса коммуникационной
                        ;области
more:   mov dl,byte ptr es:[si]; Очередной байт из источника -> dl
        mov [bx],dl  ; Теперь из dl -> в коммуникационную область
        inc bx       ; Нарращиваем базис
        inc si       ; и индекс
        loop more    ; Цикл по к-ву байт
        ret          ; Выход
SOURCE dw
NUMB   dw
LIFTTO endp
LIFT   ends
end

```

Первый оператор этой программы является неисполнимой псевдооперацией, содержащей информацию для формирования листинга. Второй оператор является неисполнимой псевдооперацией объявления программного сегмента для загрузчика, третий оператор является неисполнимой псевдооперацией объявления сегментных регистров, четвертый оператор является неисполнимой псевдооперацией объявления входной точки. Далее следуют машинные команды различного формата и, следовательно, различной длины. Операторы `dw` являются «исполнимыми» псевдооперациями резервирования памяти. «Исполнимость» этих псевдоопераций заключается в том, что они влияют на содержимое счетчика адреса `LC`.

Для формирования таблиц символов `ST` и литералов `LT` необходимо выполнить следующие действия:

1. Определять длину машинных команд из таблицы `MOT`;
2. Обработать псевдооперации типа `db` и `dw`;
3. В соответствии с длиной очередной машинной командой или псевдооперации корректировать значение счетчика адреса `LC`;
4. Запоминать символы (метки и символические адреса) и их значения (текущее значение `LC`) в таблице символов;
5. Формировать системные символы для литералов и запоминать их вместе со значениями (текущее значение `LC`) в таблице литералов. Один и тот же литерал, встречающийся несколько раз, должен быть помещен в таблицу только один раз. (Системные символы - это символы, начинающиеся с какого-либо символа, запрещенного для использования в пользовательском алфавите ассемблера, например, `$`, `!`, `?` и т. п.)

Порядок выполнения лабораторной работы.

Для выполнения настоящей лабораторной работы необходимо

1. Разработать структуру таблиц ассемблера `MOT`, `POT`, `ST` и `LT`.
2. Разработать формат исходных операторов языка ассемблера для хранения в файле исходной программы. Формат может быть позиционным, т. е., каждое поле исходного оператора имеет фиксированное количество позиций (как на перфокарте), или переменным, в котором поля отделяются друг от друга символами - разделителями (табуляция, пробел и т. п.)
3. На основе задания, полученного от преподавателя и системы команд микропроцессора сформировать таблицы `MOT` и `POT`, включив в них только те машинные операции и псевдооперации, которые присутствуют в задании.

4. Подготовить файл исходной программы на языке ассемблер в соответствии с полученным заданием .
5. Разработать приложение, реализующее следующие функции:
 - 5.1. Чтение операторов исходной программы на ассемблере (п. 4);
 - 5.2. Синтаксический разбор операторов исходной программы в соответствии с п.2 и общепринятыми правилами записи символов и меток, а именно: метка или символ могут начинаться только с латинской буквы и не могут содержать внутри себя пробелов, специальных символов и т.п. Символ, начинающийся с цифры, трактуется как целая десятичная литеральная константа. Символ, начинающийся знаком «'», трактуется как строковая литеральная константа. Имеющиеся в адресной части операторов символические обозначения регистров общего назначения игнорируются.
 - 5.3. Заполнение таблиц символов ST и литералов LT в соответствии с п. 1 и информацией, содержащейся в таблицах MOT и POT.

Содержание отчета.

Отчет по лабораторной работе должен включать в себя:

1. Описание структур разработанных таблиц;
2. Описание формата записи исходных операторов языка ассемблер;
3. Листинг разработанного приложения с комментариями.
4. Таблицы MOT, POT, ST и LT, полученные в результате выполнения работы.

Литература:

1. Х. Исида. «Программирование для микрокомпьютеров». М.: «Мир», 1988 г.
2. Р. Джордейн. «Справочник программиста персональных компьютеров типа IBM PC, XT и AT. М.: «Финансы и статистика», 1991 г.
3. Л. Бек. «Введение в системное программирование». М.: «Мир», 1988 г.