# Лабораторная работа №2

## Система машинных команд Assembler

### Вариант 1

Цель работы: изучить способы задания операндов в командах ассемблера, представление команд в виде машинных кодов.

### Задание к работе

1. Ознакомиться с теоретическим материалом.
2. В соответствии со своим вариантом описать действия команд ассемблера и указать способы адресации операндов.
3. Осуществить ассемблирование и дизассемблирование заданных команд.

### Индивидуальное задание

MOV BX, 100  
 MOV BP, [BX]  
 MOV [BP+2], DL  
MOV AX, DX   
 ADD AX, [BX+DI+4]

Отчёт

### Способы адресации операндов

1. MOV BX, 100
   1. BX — регистровый операнд;
   2. 100 — непосредственный операнд.
2. MOV BP, [BX]
   1. BP — регистровый операнд;
   2. [BX] — операнд, заданный косвенной базовой адресацией.
3. MOV [BP+2], DL
   1. [BP+2] — операнд, заданный косвенной базовой адресацией со смещением;
   2. DL — регистровый операнд.
4. MOV AX, DX
   1. AX — регистровый операнд;
   2. DX — регистровый операнд.
5. ADD AX, [BX+DI+4]
   1. AX — регистровый операнд;
   2. [BX+DI+4] — операнд, заданный косвенно базовой адресацией.

### Действия команд

Для описания действия команд сперва нужно было ассемблировать программу и получить следующий листинг (часть):

|  |  |
| --- | --- |
| 01 | Turbo Assembler Version 4.1 10/18/14 04:00:24 Page 1 |
| 02 | prog.ASM |
| 03 |  |
| 04 | 1 MASM |
| 05 | 2 0000 MODEL SMALL |
| 06 | 3 0000 .STACK 256 |
| 07 | 4 0000 .CODE |
| 08 | 5 0000 main PROC |
| 09 | 6 0000 BB 0064 MOV BX, 100 |
| 10 | 7 0003 8B 2F MOV BP, [BX] |
| 11 | 8 0005 88 56 02 MOV [BP+2], DL |
| 12 | 9 0008 8B C2 MOV AX, DX |
| 13 | 10 000A 03 41 04 ADD AX, [BX+DI+4] |
| 14 | 11 |
| 15 | 12 000D B8 4C00 MOV AX, 4c00h |
| 16 | 13 0010 CD 21 INT 21h |
| 17 | 14 0012 main ENDP |
| 18 | 15 END main |

Разберём каждую команду:

1. MOV BX, 100. Двоичное представление команды: BB 00 6416 ­= 10111011 00000000 011001002. Побайтово:
   1. 10111011. 1011 — MOV с непосредственной адресации операнда; 1 — поле w, определяющее размер данных, которыми оперирует команда, как 16 бит; 011 — поле reg, определяющее используемый в команде регистр как BX;
   2. 00000000 01100100. Является двоичным представлением 10010.
2. MOV BX, [BP]. Двоичное представление команды: 8B 2F16 = 10001011 001011112. Побайтово:
   1. 10001011. 100010 — MOV без непосредственной адресации операнда; 1 — поле d, определяющее направление передачи данных из памяти в регистр; 1 — поле w, определяющее размер данных, которыми оперирует команда, как 16 бит.
   2. 00101111. 00 — поле mod, определяющее, что поле смещения в команде отсутствует и адрес операнда определяется содержимым базового регистра; 101 — поле reg, определяющее регистр BP; 111 — поле r/m определяющее эффективный адрес [BX].
3. MOV [BP+2], DL. Двоичное представление команды: 88 56 0216 = 10001000 01010110 000000102. Побайтово:
   1. 10001000. 100010 — MOV без непосредственной адресации операнда; 0 — поле d, определяющее направление передачи данных из регистра в память; 0 — поле w, определяющее размер данных как 8 бит.
   2. 01010110. 01 — поле mod, определяющее, что смещение в команде присутствует, занимает один байт и модифицируется содержимым базового регистра; 010 — поле reg, определяющее регистр DL; 110 — поле r/m, определяющее эффективный адрес [BP]+смещение8.
   3. 00000010. Смещение равное 2.
4. MOV AX, DX. Двоичное представление команды: 8B C216 = 10001011 110000102. Побайтово:
   1. 10001011. 100010 — MOV без непосредственной адресации операнда; 1 — поле d, определяющее направление передачи данных из памяти в регистр; 1 — поле w, определяющее размер данных, которыми оперирует команда, как 16 бит.
   2. 11000010. 11 — поле mod, определяющее, что операндов в памяти нет — они находятся в регистрах; 000 — поле reg, определяющее регистр AX, 010 — поле r/m, определяющее эффективный адрес DX.
5. ADD AX, [BX+DI+4]. Двоичное представление команды: 03 41 0416 = 00000011 01000001 000001002. Побайтово:
   1. 00000011. 000000 — ADD без непосредственной адресации операнда; 1 — поле d, определяющее направление передачи данных из памяти в регистр; 1 — поле w, определяющее размер данных, которыми оперирует команда, как 16 бит.
   2. 01000001. 01 — поле mod, определяющее, что поле смещения в команде присутствует, занимает один байт и модифицируется содержимым базового и индексного регистра; 000 — поле reg, определяющее регистр AX; 001 — поле r/m, определяющее эффективный адрес [BX+DI]+смещение8.
   3. 00000100. Смещение равное 4.

### Ассемблирование и дизассемблирование

