

Лабораторные работы. Микропроцессорные системы.

Лабораторная работа №16

Программирование на Ассемблере с использованием predetermined наборов данных  
(диалект FASM)

**Оборудование:** компьютер с MS Windows XP / 7. Программное обеспечение emu8086.

При написании многих программ на ассемблере оптимальным вариантом определения данных может оказаться не вычисление а чтение наборов данных определенных в коде программы. Это относится к данным, которые известны заранее и не изменяются при всех вариантах выполнения программы.

В эмуляторе emu8086 примером такой программы могут служить stepper motor и traffic lights.

Рекомендуется размещать блок определения данных в конце кода программы. В любом случае нужно исключить возможность случайного попадания исполнителя в этот блок.

Для определения данных используются команды

*dw – 2 байта и db – 1 байт*

Например

```
datcw db 0000_0110b
      db 0000_0100b
      db 0000_0011b
      db 0000_0010b
```

В этом примере используется метка datcw чтобы определить начало блока. Перебор таких данных можно осуществить следующим образом.

```
mov bx, offset datcw      ; определяем адрес начала блока данных
mov si, 0                  ; первоначальное смещение 0
mov cx, 0 ; step counter  ;
next_step:
mov al, [bx][si]          ; помещаем в al адрес строки данных – [адрес начала],[смещение от начала]
out 7, al                 ; выводим данные
inc si                     ; увеличиваем смещение

cmp si, 4                  ; проверяем сколько блоков прочитали. Всего их 4
jb next_step
mov si, 0
```

### Задание №1

Загрузить программу stepper\_motor и изменить ее код так, чтобы двигатель вращался на 180 градусов по часовой стрелке, а потом на 180 против часовой.

В отчет: Код программы и скриншот выполнения

### Задание №1

Изменить код программы из задания 1 так, чтобы при вводе 1 в порт 110 двигатель останавливался

В отчет: Код программы и скриншот выполнения

## Лабораторные работы. Микропроцессорные системы.

Защита работы:

Для успешной защиты данной работы необходимо:

- Что такое бит и байт
- Структуру байта с учетом бита четности. Правило контроля четности
- Как нумеруются биты в слове. Как информация хранящаяся в битах может быть преобразована в понятный человеку формат
- Как слово передается из регистра процессора в память
- Что такое абсолютный адрес операнда и адрес в системе сегмент-смещение
- Для чего применяют сегментирование памяти, как работает сегментирования память процессора 8086
- Как выполняется адресация байт в слове
- Что такое «внутренние регистры процессора»
- Для чего служат регистры AX, BX, DX, CX?
- Для чего служат регистры CS, DS, ES, SS?
- Для чего служат регистры SP, BP, SI, DI, IP?
- Для сего служит регистр FLAGS?
- Каковы роли битов в регистре FLAGS?
- Знать команды пересылки данных, переходов, команды арифметических и логических операций языка ассемблер
- что такое стек?
-