Прерывания
Вывод и ввод символов

Курсовой проект "Эмулятор PDP-11"
Занятие 7

Тест урока

- putchar — напечатать один символ *
- hello — напечатать Hello, world!

Регистры внешних устройств

- Внешние устройства:
  дисплей, клавиатура, принтер, диски
- Взаимодействие через ячейки памяти
  (последние 8Кб)
- Дисплей: адреса (восьмеричные!)
  0177564 — команды и статус
  0177566 — данные
- Клавиатура: адреса
  0177560 — команды и статус
  0177562 — данные

Печать 1 символа

- ASCII код записывается в **байт** по адресу **0177566**
- **'a** — ASCII код символа a (одна одинарная кавычка).
  Осюда пошло обозначение константы 'a' в C.
- псевдокоманда ассемблера
  odata = 177566 ; регистр данных дисплея
- movb #'a, @#odata
  или
  movb #'a, odata
- Как напечатать hello?
  movb #'h, odata
  movb #'e, odata
  уже не работает, movb быстрее, чем вывод на экран

Регистр состояния устройства

- 16 бит, описывают состояние устройства,
  7 бит - бит готовности устройства
  - 1 — устройство готово выводить символ
  - 0 — устройство занято
- Ждем, пока устройство станет готовым
  Печатаем символ
- 7 бит — знаковый (1 — отрицательное число)
- ostat = **177564**; регистр состояния дисплея
  odata = **177566**; регистр данных дисплея

Проверка готовности устройства

- ostat = 177564
  odata = 177566

```
        .= 1000
        movb #'*, R0      ; ascii код символа *

    putchar:
        tstb 177564       ; бит готовности во флаг N
        bpl putchar       ; устройство не готово? ждем
        movb R0, odata ; печатаем символ


        halt
```

Псевдокоманды (директивы) ассемблера

- Начинаются с точки
- Не мнемоники машинного кода, а управляют ассемблером или влияют на размещение данных
- **.=** адрес — изменить текущий адрес размещения кода
- **.BYTE** 3, 7, -11
  Разместить числа по 1 числу на байт
- **.BLKB** 4
  Резервировать 4 байта в памяти, начиная с текущего
- **.WORD** 3, 7, -11
  .BLKW 4

Константы и метки ассемблера

- A = 123
  .= 1000
  B: .WORD 15, 12. , 'z, A, B
- 001000 000015 **восьмеричная** константа 15 легла по адресу, указанному в .= как текущий
- 001002 000015 **десятичное** 12 есть 15 в восьмеричной системе
- 001004 000172 это **ascii-код** символа z
- 001006 000123 значение **константы** A
- 001010 001000 значение константы (**метки**) B

Запись строки

- **.BYTE** 'h, 'e, 'l, 'l, 'o, 0
- **.ASCII** #Hello, world!#
  .ASCII SHello, world!S
  .ASCII /Hello, world!/
  .ASCII #Hello,#<15><12>#world!# - после запятой напечатаются \r\n (символы возврата каретки и новой строки)
- **.ASCIZ** #Hello, world!# ; ноль в конце
- **.EVEN** - следующее слово с четного адреса (пропускает 1 байт, если нужно)

Hello, world!

```
            .= 200        ; блок данных
A:          .ASCIZ   /Hello, world!/
            .= 1000
            mov      #A, R1          ; R1 — адрес символа
puts:       movb    (R1)+, R0        ; R0 — ascii-код символа
            beq      END             ; это конец строки?
putchar:                             ; печать 1 символа
            tstb     ostat           ; устройство готово?
            bpl      putchar         ; ждем если не готово
            movb    R0, odata        ; печатаем 1 символ
            br       puts            ; идем к следующему
END:        halt
```