?????? ? ????? ??????

Завершение программы

```
    mov rax, 60
    mov rdi, 0
    syscall
```

При использовании gcc можно

```
    ret
```

Код возврата

Linux

Без отладчика можно смотреть состояние одного регистра за счет копирования его в регистр rdi (Linux) при завершении программы.

```
global _start           ; делаем метку метку _start видимой извне

section .text           ; объявление секции кода
_start:                 ; объявление метки _start - точки входа в программу
    mov rdi, 23         ; помещаем в регистр rdi код возврата - 23
    mov rax, 60         ; 60 - номер системного вызова exit
    syscall             ; выполняем системный вызов exit
```

Затем выполняется приложение, команда $? выводит код завершения предыдущей команды

```
root@Eugene:~/asm# ./hello
root@Eugene:~/asm# echo $?
23
```

Windows:

```
global _start       ; делаем метку метку _start видимой извне

section .text       ; объявление секции кода
```

```
_start:             ; метка _start - точка входа в программу
    mov rax, 23     ; помещаем в регистр rax код возврата - 23
    ret             ; выход из программы
```

Получение кода возврата

```
echo %ERRORLEVEL%
```

Строки

Статичное определение строки и длины (для последующего вывода)

```
section .data
msg db "Hello!",10,0
    msg_len equ $ - msg  ; $ - текущая позиция ассемблера
    msg_half_len equ (msg_full_len - 2)/2 ; возможен такой расчет
```

Использование системного вызова

```
global main
section .data
msg db "Hello",10,0
    msg_len equ $ - msg
section .text
main:
mov rax, 1
    mov rdi, 1
    mov rsi, msg
    mov rdx, [msg_len] - 1
    syscall
    ret
```

Использование функции C

```
global main
extern printf
section .data
  fmtint db "Result: %d",10,0
  fmtstr db "Outstring: %s",10,0
  msg db "for textout",0
...
```

```
section .text
main:
...
; вывод числа
mov rsi, <переменная> ; в rsi то что нужно вывести
    mov rax, 0
mov rdi, fmtint
call printf
; вывод строки
mov rsi, msg
    mov rax, 0
mov rdi, fmtstr
call printf
```

Revision #4
Created 8 November 2025 15:38:22 by Admin
Updated 8 November 2025 17:33:37 by Admin