# Некоторые системные вызовы

## ОС Linux

1. Номер системного вызова записывается в регистр EAX

2. Параметры передаются через регистры (см. таблицу)

3. Для осуществления системного вызова выполняется прерывание с номером 0x80 (80h), команда

```
int    0x80
```

4.Результат системного вызова получается в EAX

*(значение из диапазона 0xFFFFF000 — 0xFFFFFFFF говорит об ошибке)*

| | EAX перед систем-ным вызовом - номер | EBX | ECX | EDX | EAX после системного вызова - результат |
|---|---|---|---|---|---|
| **write**<br><br>запись данных (печать) | 4 | дескриптор потока<br><br>1- стандартный вывод (на экран) | адрес, откуда печатать данные | сколько байтов записать (напечатать) | сколько байтов реально удалось записать (напечатать) |
| **read**<br><br>чтение данных | 3 | дескриптор потока<br><br>0- стандартный ввод (с клавиатуры) | адрес, куда записывать данные | сколько байтов попытаться прочитать (в потоке данных может быть меньше) | сколько байтов реально удалось прочитать (если 0 — поток ввода сразу был закрыт) |
| **_exit**<br><br>завершает программу | 1 | код возврата | | | |

Пример: выход из программы с нулевым кодом возврата (успешное завершение)

```
mov    eax,   1 ;номер системного вызова
mov    ebx,   0; код возврата
int      0x80
```

# ОС FreeBSD (и MacOS)

1. Номер системного вызова записывается в регистр EAX

2. Параметры передаются через стек (в обратном порядке), каждый параметр — двойное слово

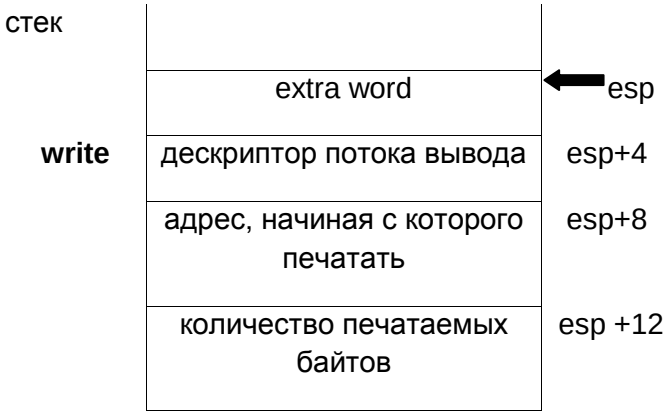
3. Для осуществления системного вызова выполняется прерывание с номером 0x80 (80h), команда

```
int    0x80
```

4.Результат системного вызова хранится во флаге CF:

CF=0 — системный вызов успешно завершен, CF=1 — произошла ошибка

Содержимое стека, необходимое для выполнения системного вызова write (перед int 0x80):

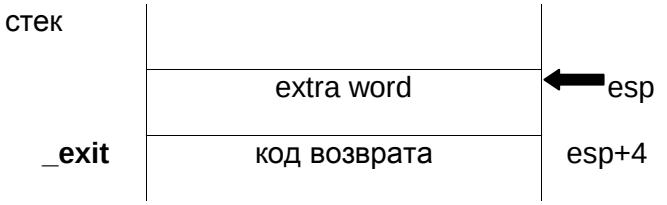



```
int     0x80
```

Далее необходимо очистить стек от параметров

```
add    esp,  16
```

Содержимое стека, необходимое для выполнения системного вызова _exit (перед int 0x80)




## Пример: печать строки

```
global _start

section .data
msg     db    'Hello', 10 ; строка
len     equ   $-msg ; длина строки

section .text
_start:
        push    dword len
        push    dword msg
        push    dword 1  ;1 — вывод на экран
        mov     eax, 4     ;номер вызова write
        push    dword 10; extra word - любое
        int     0x80
        add     esp,   16; очистка стека

        push    dword 0  ;0 — код возврата
        mov     eax, 1    ;номер вызова _exit
        push    dword 10; extra word - любое
        int     0x80
```