# 高级页面仔公众号学习笔记

微信公众号：Soulghost 高级页面仔 （专门科普iOS底层知识）


两个大神的ARM汇编教程
(知兵)[https://zhuanlan.zhihu.com/p/31168191]
(刘坤的技术博客)[https://blog.cnbluebox.com/blog/2017/07/24/arm64-start/]

## iOS汇编入门教程（一） ARM64汇编基础

以反调试为例，我们知道，通过调用ptrace函数可以阻止调试器依附。

```
ptrace(31,0,0,0)
//可以用facebook的fishhook,轻松的hook绕过这个函数.
```

```
mov x0 ,#31
mov x1 ,#0
mov x2 ,#0
mov x3 ,#0
mov x16 ,#26 ;??，存储的是函数编号，通过Apple提供的System Call Table 可以查出ptrace的编号为26
svc #80      ;发起系统调用
```

防止被hook的ptrace调用

```
//用inline的方式将函数再调用处，强制展开，防止被hook和追踪符号.
static __attribute__((always_inline)) void anti_debug()
```

```
{
#ifdef __arm64__
    //volatitle修饰符能够防止汇编指令被编译器忽略
    __asm__ __volatile__(
        "mov x0,#31\n"
        "mov x1,#0\n"
        "mov x2,#0\n"
        "mov x3,#0\n"
        "mov x16,#26\n"
        "svc #80\n"
    )
}
```

**汇编入门最难的地方在于对栈的理解**
**在汇编中没有变量的概念，只有寄存器和内存**

```
#clang编译为特定指令集汇编代码，编译ARM64指令集
clang -S -arch arm64 -sysroot `xcrun --sdk iphoneos --show-sdk-path`
hello.c
```

**知识点：**

```
str和ldr是一堆指令，str的全称是store register,即将寄存器的值存储到内存中，ldr的全称
是load register,即将内存中的值读到寄存器，
因此他们的第一个参数都是寄存器，第二个参数是内存。
[sp,#12]  ;代表的是sp + 12地址.
[sp,#-12] ;代表的是sp - 12地址.
str w0,[sp,#12] ;代表的是w0 寄存器的数据存放到 sp + 12这个位置，占用（sp+12） ~
(sp+16)空间 ,也就是4字节。
```

**在编译器生成汇编时，首先会计算需要的栈空间大小，并利用sp指针向低地址开辟相应的空间**

```
;开辟一个16字节的空间，3个变量a, b, res
;int占用4字节，arm64补齐占用16字节。

;栈是由高地址向低地址生长的数据结构。
;因此sp - 16 就相当于开辟了一个16字节空间
sub sp,sp,#16 ;sp = sp - 16

str w0,[sp,#12];将w0 32位寄存器的值 保存到sp + 12 地址中. int a
str w1,[sp,#8] ;将w1 32位寄存器的值 保存到sp + 8  地址中. int b
```

**只有寄存器才能参与运算**

```
ldr w0,[sp,#12] ;把sp+12的值给取出来保存到w0中
ldr w1,[sp,#8]  ;把sp+8的值取出来保存到w1中
;相加两个寄存器保存到w0中
add w0,w0,w1
```