6.堆栈操作

名字	含义	备注
push	减少esp，再将源操作数复制到堆栈	16位esp减少2，32位esp减少4
pop	首先将esp指向的内容复制到目的操作数中，再增加esp值	
pushfd	把32位eflags寄存器压入堆栈	
popfd	把栈顶单元内容弹出到eflags寄存器	
pushad	将所有32位通用寄存器压入堆栈	顺序：eax,ecx,edx,ebx,esp,ebp,esi,edi
popad	与pushad相反	
pusha	与pushad类似，只不过变为16位寄存器	
popa	与pusha相反	

举例子：

```
;字符串翻转
.386
.model
.stack 4096
ExitProcess PROTO,dwExitCode:DWORD
.data
aName BYTE "Abraham Lincoln",0
nameSize = ($ - aName) - 1
.code
main PROC
;将名字压入堆栈
mov eax,nameSize
mov esi,0
L1:
   movzx eax,aName[esi] ;获取字符
   push eax
   inc esi
   loop L1
;将名字按逆序弹出堆栈
;并存入aName数组
   mov ecx,nameSize
   mov esi,0
L2:
   pop eax    ;获取字符
   mov aName[esi],al ;存入字符串
   inc esi
   loop L2
   INVOKE ExitProcess,0
main endp
END main
```

7.过程

7.1 PROC和ENDP伪指令，过程用PROC和ENDP来定义

```
;举例子：
main PROC
..
..
..
main ENDP
```

汇编可以从一个过程中goto到另外一个过程只需要在标号后面加::俩个:号定义为全局标号。

7.2 CALL和RET

名字	含义	备注
call	调用一个过程	将返回地址压入栈中，再把被调用过程复制到指令指针寄存器
ret	结束返回一个过程	类似 pop eip (不可以这样写: pop eip,但是ret含义与此相似)

7.3 USES

和PROC一起使用，让程序员列出在该过程中可能修改的寄存器，在过程开始处生成push，结尾处生成pop。