

伪算法描述:

asm.s 文件控制中断前的处理过程和中断后的恢复过程, 控制形式为硬中断

begin

声明全局函数

记录中断

无错误码情况:

```
change addr    //函数进栈
push ax
push bx
push dx
...
push 0          //没有错误码, 入栈 0
esp + 44 → edx
push edx
init ds, es, fs
pop 0, edx //这里使用 esp 段指针, 直接弹出两个数据, 在对应 C 函数使用
pop fs
pop es
...
pop eax
```

出现的中断情况:

```
int 0
int 1
int 2
...
int 9
int 15    //保留中断入口
send sign //发出中断信号
answer    //cpu 相应中断
```

有错误码情况:

```
change addr    //函数进栈
push ax
push bx
push dx
...
push eax        //错误码入栈
esp + 44 → eax
push eax
init ds, es, fs
use Cfun        //调用 C 函数, 参数已经入栈 (错误码+函数地址)
pop ebx, eax    //这里使用 esp 段指针, 直接弹出两个数据, 在对应 C 函数
```

```
    使用
    pop fs
    pop es
    ...
    pop eax
    出现的中断情况:
    int 10
    int 11
    ...
    int 13
end
```

补充栈:

栈特点: 先进后出, 后进先出

栈操作: 栈的操作就两种, 分别为出栈和入栈

栈实现: 栈的实现主要有两种, 一种是基于数组的实现, 叫做顺序栈, 另外一种是基于链表的实现, 叫做链式栈