# JOS lab3

**1713429 曾志韬 计算机科学与技术**

## 1 用户进程

## 2错误处理

## 3 缺页，

## 4断点

## 5系统调用

## Code

**1 env\_init()：初始化Env表**

**2 env\_setup\_vm()：为新进程设置好其自己页目录表（虚拟内存），映射到UVPT方便用户进程读。**

**3 region\_alloc()：为新用户进程分配物理空间并映射到其虚拟地址上。   
4 load\_icode()：将被链接进内核的ELF，按其每段指定的用户虚拟地址，加载到新进程的虚拟空间。**

**5 env\_create()：创建新进程并载入到内存**

**6 env\_run()：让一个进程进入RUNNING状态**

**7 trapentry.S 宏定义各异常的处理函数 ,\_alltraps中保存一些信息，为给转跳到trap()进一步处理**

**8 trap.c 初始化 idt[]表 lidt指令载入**

**9 trap\_dispatch() 中断/异常号，运行的处理函数**

**10 T\_SYSCALL 系统调用中断**

**11 libmain()中thisenv设置为当前的envid，此处为user/hello使用**

**12 page\_fault\_handler()检查tf\_cs的RPL位，如果是内核模式则panic**

## 1问题

**1 不同的类型的中断需要不同的函数处理**

**如果不同的类型的中断指向同一个处理函数，**

**这就破环了保护机制**

**2 缺页异常 Gate的特权级是0.**

**用户特权用3**

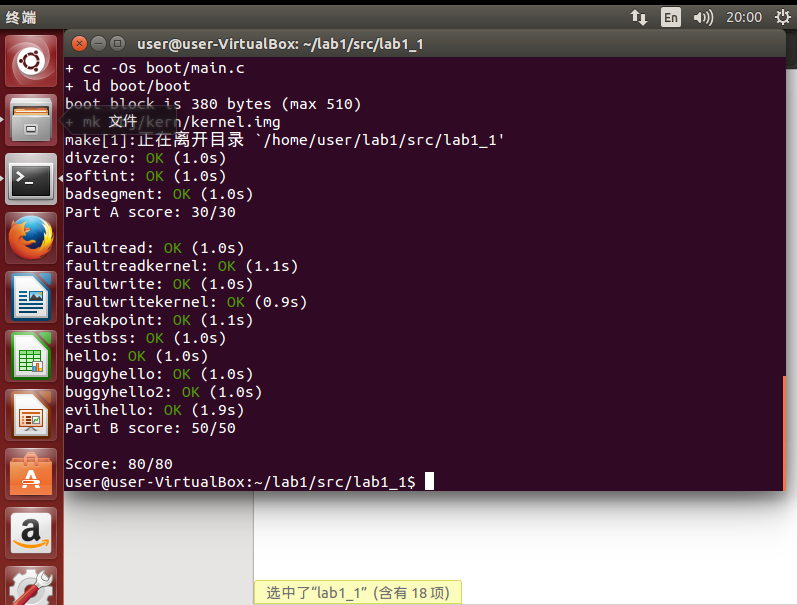
**所以需要通用保护异常**

## 2问题

**1 插入 int 3 造成断点异常 特权级DPL=3**

**2 保护机制**

## 3 Make Grade



## 4 Make Qemu

