


操作系统实验

Labs of Operating Systems

Lab 4 保护模式下中断的实现

 **中山大学** 计算机学院（软件学院）
SUN YAT-SEN UNIVERSITY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

实验安排

实验-4： 保护模式下中断的实现

进入保护模式后，了解内核如何接管操作系统，了解保护模式下计算机如何处理中断程序，如何对8259A芯片进行编程，添加处理时钟中断函数，理解两类中断-“外设中断”“异常”等。

1. 使用混合编程编写内核；
2. 实现在保护模式下中断的处理；
3. 在8259A芯片基础上，实现处理时钟中断函数；
4. 了解混合编程下，Makefile的使用和编写；

更多详情请阅读[gitee实验教程](#)！

2

实验代码项目结构化

代码组织结构：
将内核、编译、库、工具等代码分类整理放到不同的文件夹中；

```

├── build
│   └── makefile
├── include
│   ├── asm_utils.h
│   ├── boot.inc
│   ├── os_type.h
│   └── setup.h
├── run
│   ├── gdbinit
│   ├── hd.img
│   └── src
│       ├── boot
│       │   ├── bootloader.asm
│       │   ├── entry.asm
│       │   └── mbr.asm
│       ├── kernel
│       │   ├── setup.cpp
│       │   └── utils
│       │       └── asm_utils.asm

```

```

└─ kern
   ├── debug
   ├── driver
   ├── fs
   ├── init
   ├── libs
   ├── mm
   ├── sync
   ├── trap
   ├── libs
   └─ tools
      ├── boot.ld
      ├── function.mk
      ├── gdbinit
      ├── grade.sh
      ├── kernel.ld
      ├── sign.c
      ├── vector.c
      ├── .projectile
      └─ Makefile

```

uCore

```

boot drivers kernel Makefile os_image others README
cpf@cpf-VirtualBox:~/OSCourse/References/writing-a-simple-operating-system-from-scratch/v15_organize_code_base$

```

3

中断

	中断类型		
异步的	外部的中断		隐式的
同步的		陷阱	
		系统调用	显式的
	外部的/硬件	内部的/软件	

- 内部中断/软中断，可在程序中使用int指令调用。
- 在实模式下，BIOS中集成了一些中断程序，在BIOS加电启动后这些中断程序便被放置在内存中。
- 但是，BIOS内置的中断程序是16位的，在保护模式下不再适用。在保护模式下，我们需要自己去实现中断程序。

4



中山大学

SUN YAT-SEN UNIVERSITY

计算机学院（软件学院）

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

保护模式下中断向量号约定

向量号	助记符	说明	类型	错误号	产生源
0	#DE	除出错	故障	无	DIV或IDIV指令
1	#DB	调试	故障/陷阱	无	任何代码或数据引用，或是INT 1指令
2	--	NMI中断	中断	无	非屏蔽外部中断
3	#BP	断点	陷阱	无	INT 3指令
4	#OF	溢出	陷阱	无	INTO指令
5	#BR	边界范围超出	故障	无	BOUND指令
6	#UD	无效操作码（未定义操作码）	故障	无	UD2指令或保留的操作码。（Pentium Pro中加入的新指令）
7	#NM	设备不存在（无数学协处理器）	故障	无	浮点或WAIT/FWAIT指令

5



中山大学

SUN YAT-SEN UNIVERSITY

计算机学院（软件学院）

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

保护模式下中断向量号约定

8	#DF	双重错误	异常终止	有(0)	任何可产生异常、NMI或INTR的指令
9	--	协处理器段超越（保留）	故障	无	浮点指令（386以后的CPU不产生该异常）
10	#TS	无效的任务状态段TSS	故障	有	任务交换或访问TSS
11	#NP	段不存在	故障	有	加载段寄存器或访问系统段
12	#SS	堆栈段错误	故障	有	堆栈操作和SS寄存器加载
13	#GP	一般保护错误	故障	有	任何内存引用和其他保护检查
14	#PF	页面错误	故障	有	任何内存引用
15	--	(Intel保留，请勿使用)		无	

6



保护模式下中断向量号约定

16	#MF	x87 FPU浮点错误（数学错误）	故障	无	x87 FPU浮点或WAIT/FWAIT指令
17	#AC	对起检查	故障	有 (0)	对内存中任何数据的引用
18	#MC	机器检查	异常 终止	无	错误码（若有）和产生源与CPU类型有关 （奔腾处理器引进）
19	#XF	SIMD浮点异常	故障	无	SSE和SSE2浮点指令（PIII处理器引进）
20- 31	--	(Intel保留，请勿使用)			
32- 255	--	用户定义（非保留）中断	中断		外部中断或者INT n指令



保护模式下中断程序处理过程

1. 中断前的准备，准备IDT
2. CPU检查是否有中断信号。
3. CPU根据中断向量号到IDT中取得处理这个向量的中断描述符。
4. CPU根据中断描述符中的段选择符到GDT中找到相应的段描述符。
5. CPU根据特权级的判断设定即将运行程序的栈地址。
6. CPU保护现场。
7. CPU跳转到中断服务程序的第一条指令开始处执行。
8. 中断服务程序运行。
9. 中断服务程序处理完成，使用iret返回。

更多详情请阅读gitee实验教程！



中山大學 计算机学院（软件学院）
SUN YAT-SEN UNIVERSITY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

谢谢

9