**云南大学软件学院**

**实 验 报 告**

课程： 操作系统实验 任课教师： 储 星 实验指导教师： 谢 诚

姓名： 陈俊宏 学号： 20211060245 专业： 人工智能 日期： 2023-04-17 成绩：

**实验二 中断/异常的处理过程**

1. **实验目的**

1、学习进出操作系统内核的方式；

2、学习通过中断响应进出操作系统内核；

3、分析响应中断/异常时，CPU 做了哪些工作；

4、如何查看当前寄存器的状态；

5、如何查看当前栈顶的状态。

**二、实验内容**

1、设置某版本的内核为分析对象；

2、用 gdb 调试内核；

3、查看 C 语句编译之后对应的汇编指令片段；

4、分析响应中断/异常时，CPU 做了哪些工作；

5、查看当前寄存器的状态；

6、查看当前栈顶的状态。

**三、 实验要求**

1、本次实验不分组。

2、提交实验报告：纸质报告4月18日，交电子报告。

3、描述清楚实验过程，特别是除零异常分析、int指令分析、iret指令分析，撰写实验总结。

**四、 实验过程**

1、设置某版本的内核为分析对象：

#### 设置版本1内核为分析对象

首先解压版本1内核源码。使用cp命令将/data/workspace/myshixun/exp1中的1.tgz复制到~/os/目录下；

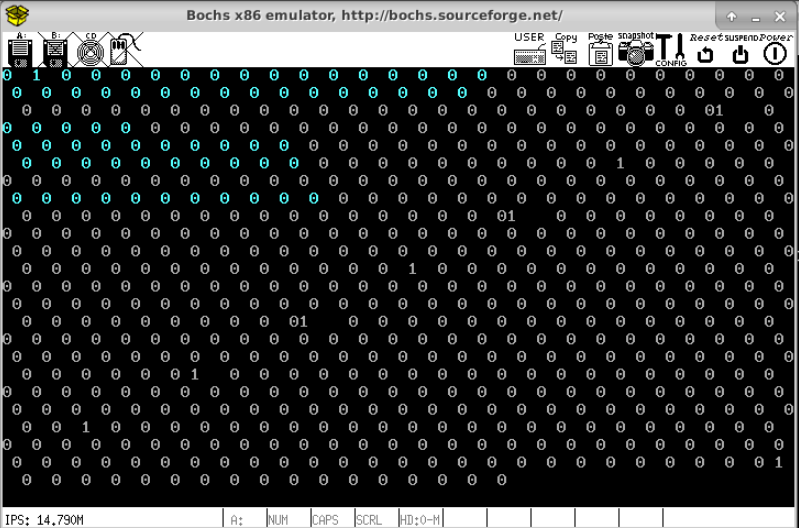
切换到~/os/linux-0.11-lab目录下，将1.tgz解压到当前目录下；

然后调整cur的指向。先使用rm -rf cur将cur删除，再使用ln命令创建符号链接。

现在可以编译和测试版本 1 内核。首先进入1/linux目录下编译内核；

确认内核映像文件Image已经生成；

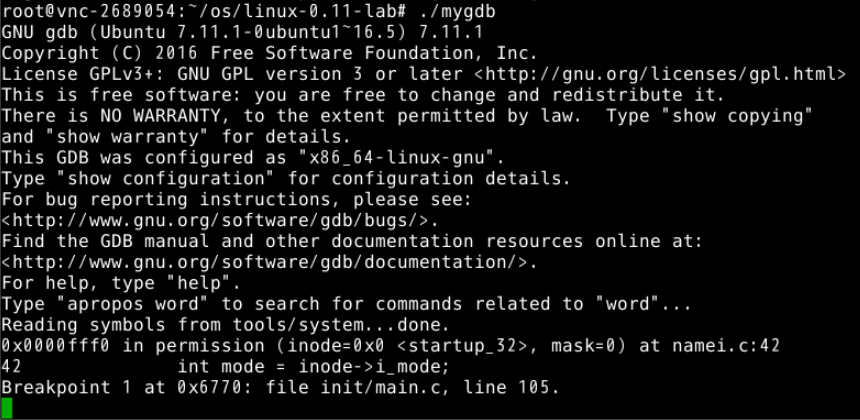
然后回到目录~/os/linux-0.11-lab，并使用./run启动虚拟机检测内核是否正常；

如果正常虚拟机在加载完毕之后将会出现如下画面。

2、用 gdb 调试内核：

#### 开始使用 gdb 调试内核

先关闭bochs虚拟机，然后打开两个终端，其中一个终端在linux-0.11-lab目录下运行rungdb脚本，以启动 bochs 虚拟机并等待 gdb 连接；

在另一个终端里切换到目录~/os/linux-0.11-lab/，然后启动脚本./mygdb，这个命令会启动 gdb 并读入内核符号信息，同时会通过执行0.gdb中的调试命令来连接到 bochs 虚拟机，并进而跟踪到 main 函数入口。

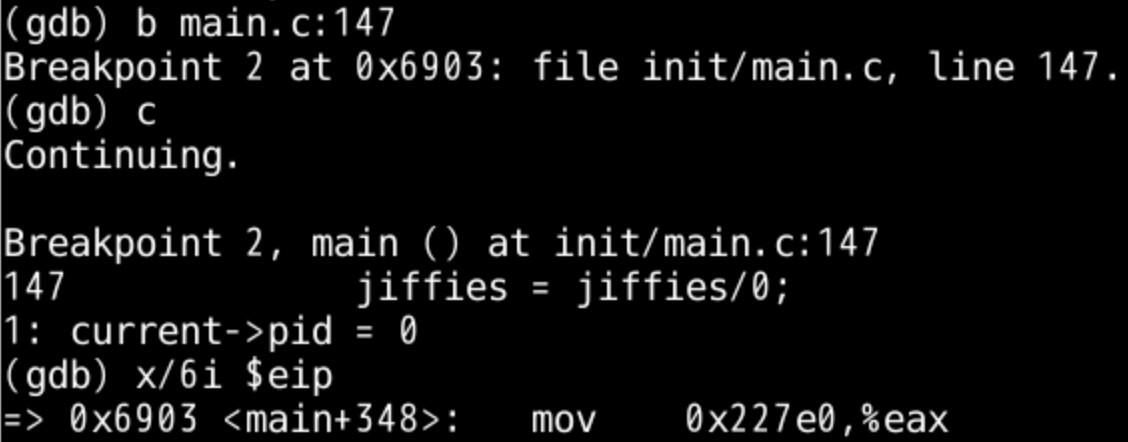
、

3、查看 C 语句编译之后对应的汇编指令片段

#### 查看 C 语句编译之后对应的汇编指令片段

如果要查看某条 C 语句编译之后对应的汇编指令片段，可以在该 C 语句处设置断点，并跟踪到该断点，然后反汇编，所看到的当前指令之后的一段汇编指令就对应于该 C 语句。

例如，jiffies = jiffies/0;是文件 main.c 的第 147 行，可以如下方式查看：



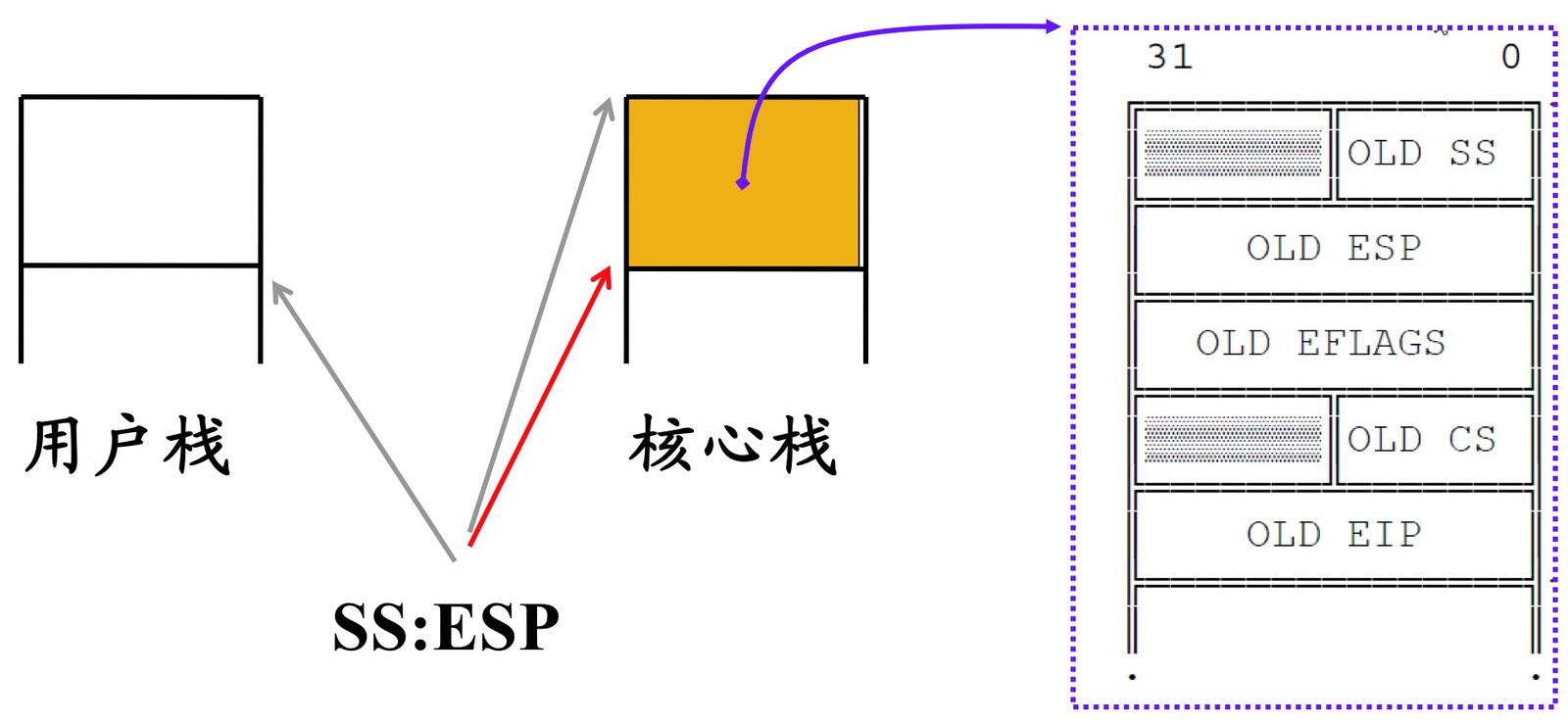
上面显示的汇编指令中，有一行前面有箭头标识，此即为当前指令，即马上将要执行的指令。

4、分析响应中断/异常时，CPU 做了哪些工作

#### 分析响应中断/异常时，CPU 做了哪些工作

* 切换到核心栈，并在其中保存中断现场。
* 转到中断处理程序去运行，并切换到核心态。

如下图所示：



上图显示了栈中中断现场的结构，（OLD SS:OLD ESP) 描述了用户栈顶的位置，（OLD CS:OLD EIP) 描述了恢复点的位置。

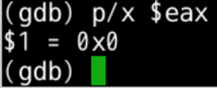
5、查看当前寄存器的状态；

#### 如何查看当前寄存器的状态

使用 gdb 调试命令 info registers 即可，如下所示：



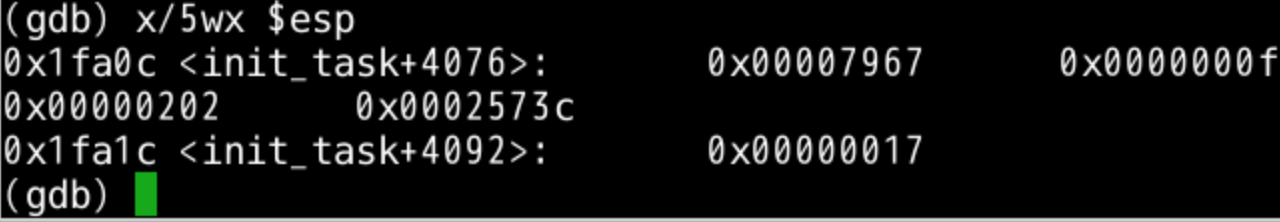
也可以单独查看某一个寄存器：



6、查看当前栈顶的状态。

#### 如何查看当前栈顶的状态

可以使用命令 x 来查看：



上面显示了栈顶的 5 个长字，是某异常发生时的中断现场，其中存储的用户栈顶的位置是 0x17:0x2573c ，存储的恢复点的位置是 0xf:0x7967 。需要注意的是，x86 中栈是从高地址向低地址方向增长的，这里的栈顶位置是 0x1fa0c 。

除零异常分析：



int 指令分析：



iret指令分析：



**五、实验总结**

答：

本次实验主要学习了进出操作系统内核的方式，以及如何通过中断响应进出操作系统内核。我们使用了gdb调试工具来进行内核调试，并学习了如何查看C语句编译后对应的汇编指令片段。此外，我们还分析了响应中断/异常时，CPU做了哪些工作，并学习了如何查看当前寄存器的状态和当前栈顶的状态。

总的来说，通过本次实验，我们深入了解了操作系统内核的工作原理和调试方法，对于进一步学习操作系统的原理和开发具有重要的意义。