|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 武汉大学国家网络安全学院教学实验报告 | | | | | |
| 课程名称 | 操作系统设计与实践 | | 实验日期 | | 2024.10.12 |
| 实验名称 | 基本实验环境搭建 | | 实验周次 | | 第一周 |
| 姓名 | 学号 | | 专业 | | 班级 |
| 王亚鹏 | 2022302181161 | | 信息安全 | | 5班 |
| 杨依磊 | 2022302181159 | | 信息安全 | | 5班 |
| 杜泓波 | 20223021811 | | 信息安全 | | 5班 |
| 侯名扬 | 20223021811 | | 信息安全 | | 5班 |
| 1. 实验目的及实验内容   （要求掌握的知识；实验内容；原理分析） | | | | | |
| **总目标：理解中断与异常机制的实现机理**  **实验内容：**  1. 理解中断与异常的机制  2. 调试8259A的编程基本例程  3. 调试时钟中断例程  4. 实现一个自定义的中断向量，功能可自由设想  **原理分析：**  **本实验有如下几个要点**  **（1）中断与异常的相同与不同：**  中断是操作系统与外部硬件设备交互的重要手段。它允许外部设备在需要处理时向CPU发出信号，使CPU暂时停止当前任务并响应中断请求  异常则是CPU在执行过程中由于某些条件或错误而触发的内部事件，通常用于处理系统错误或调试  **（2）8259A可编程中断控制器的工作原理：**  8259A是一种可编程中断控制器，它负责管理多路外部中断信号，并根据优先级将这些信号传递给CPU。8259A通过中断请求寄存器IRR、中断服务寄存器ISR和中断屏蔽寄存器IMR来管理中断信号的接收、处理和屏蔽  **（3）中断描述符表IDT：**  在保护模式下，操作系统使用IDT来管理中断和异常。IDT的每一项对应一个中断或异常的处理程序入口地址，当中断或异常发生时，CPU会根据中断号查找IDT中的对应项并跳转执行相应的中断处理程序  **（4）时钟中断的控制：**  时钟中断是操作系统中重要的中断，用于定时任务调度、时间计数等系统功能。时钟中断通常通过8254计时器触发，由8259A中断控制器捕捉并传递给CPU。通过配置8259A的IMR寄存器，可以控制时钟中断的启用或屏蔽  **（5）自定义中断的实现：**  实现自定义中断的核心是理解如何通过调用修改与中断相关的寄存器、向量表等协同工作，来实现自定义功能，从而能够对中断的执行得到进一步认识  **（6）IOPL的作用与保护机制：**  IOPL即I/O特权级别决定了当前任务的I/O权限。只有在权限满足要求时，程序才能直接执行I/O操作。通过这种机制，操作系统能够限制用户程序访问底层硬件设备，确保保护系统的安全 | | | | | |
| 1. 实验环境及实验步骤   （本次实验所使用的器件、仪器设备等的情况；具体实验步骤） | | | | | |
| 实验环境：  **X86 32位Linux环境**  实验步骤：  **1. 理解中断与异常的机制**  本实验中通过8259A中断控制器和中断描述符表IDT来管理中断。主要的中断处理流程：8259A收到中断请求后，确认是否允许，再通知CPU进入处理，处理完通过EOI通知8259A  中断与异常的区别：中断由外部硬件发出，打断CPU来处理高优先级任务；异常是程序执行中产生的错误事件  **2. 调试8259A的编程基本例程**  位于pmtest9a.asm，运行结果如下：    8259A的格式如下：    我们可以通过对其寄存器的设置来屏蔽或打开相应的中断。8259A是可编程中断控制器对它的设置并不复杂是通过向相应的端口写入特定的ICWInitialization Command Word来实现的。主8259A对应的端口地址是20h和21h从8259A对应的端口地址是A0h和A1h。ICW共有4个每一个都是具有特定格式的字节，具体如下：    1. 往端口20h主片或A0h从片写入ICW1  2. 往端口21h主片或A1h从片写入ICW2  3. 往端口21h主片或A1h从片写入ICW3  4. 往端口21h主片或A1h从片写入ICW4  **这4步的顺序是不能颠倒的**  当每一次中断处理结束需要发送一个EOI给8259A以便继续接收中断。而发送  EOI是通过往端口20h或A0h写OCW2来实现的  最后在相应的位置添加调用Init8259A的指令之后对8259A的操作就结束了  这些具体可参见asm源码中的注释部分，写得很详细  **3. 调试时钟中断例程**  这个中断的代码位于pmtest9.asm  运行结果是屏幕上会打印出变化的文字，以下是几个快照      可以看到文字的变化，运行结果的视频随压缩包附上  （1）修改初始化8259A的代码时钟中断不再屏蔽代码    （2）把IDT修改成如下代码    （3）我们需要让程序停留在某个地方以便观察    **4. 实现一个自定义的中断向量**  实现了自定义中断test.asm，功能是打印出彩虹条带，运行结果如下图所示  实现的主要步骤如下：  （1）CustomHandler 中断处理程序实现了彩虹条带的打印功能。\_CustomHandler 中的逻辑通过在屏幕的多个行上打印不同颜色的字符实现了彩虹效果。每一行使用不同的颜色属性，从而实现视觉上的彩虹条带效果  （2）PrintLine 函数负责在特定行上显示一整行相同的字符，我们使用字符 \*来作为条带的组成元素  （3）在所有颜色条带都打印完毕之后，\_CustomHandler 通过以下代码发送 EOI 信号来通知 8259A 中断处理完成。最后，使用 iretd 指令返回中断前的程序执行，这样系统能够继续正常执行其他指令 | | | | | |
| 1. 实验过程分析   （详细记录实验过程中发生的故障和问题，进行故障分析，说明故障排除的过程及方法。根据具体实验，记录、整理相应的数据表格等） | | | | | |
| **故障及其排除方法**：  1.一开始在idt段中定义了一行081h后，没有预留空格，导致在实际编译运行过程中，发生报错，这是因为我改变了idt段的长度，而idtlen还是原来的数据，这一行在实际上并没有运行，才会导致 int 0x81未定义  2.一开始将字符和属性位置弄反了，导致打印出来是乱码，修改了PritLine 函数，以及CustomerHandler段中的内容后就没问题了 | | | | | |
| 1. 实验结果总结   （对实验结果进行分析，完成思考题目，并提出实验的改进意见） | | | | | |
| **结果分析**：  **思考题目：**  **1. 什么是中断，什么是异常**  查阅资料可得知相关内容  **（1）中断**：中断是一种由外部设备或系统硬件发出的信号，要求处理器暂停当前执行的任务，转而去处理这些信号对应的处理程序。在中断处理完成后，处理器会继续执行原任务。中断通常是**异步**的，发生在当前指令的执行之外，比如键盘、鼠标或其他外部硬件设备的输入。  中断有如下分类  **外部中断：**  外部硬件设备（如键盘、鼠标、网卡等）发出信号。例如，当按下键盘时，键盘控制器会产生一个中断，处理器会调用相应的中断处理程序。  **软件中断：**  由程序显式调用的中断指令（如x86中的INT指令）。这种中断用于切换到内核模式或调用系统服务  **（2）异常**：异常是由CPU在执行程序时遇到某些条件例如除零错误、非法操作码等而触发的事件。异常通常是**同步**的，因为它发生在指令的执行过程中。根据不同类型，异常有如下常见分类  **Fault：**异常发生时，处理器可以通过纠正错误重新执行导致异常的指令。例如，页面缺失异常是处理器在尝试访问内存页时发现页面不存在，从而触发缺页异常，操作系统会加载页面并重新执行该指令。  **Trap：**当异常发生时，处理器允许当前指令执行完毕后再处理异常，常用于调试程序。  **Abort：**严重错误（如硬件故障）导致无法继续执行程序，此时处理器无法恢复该指令执行。  **2. 8259A的工作原理是怎样的？怎么给这些中断号的处理向量初始化值？**  **8259A工作原理：**  **（1）中断请求：**  8259A是一种可编程中断控制器，它可以从外部设备接收中断请求，并通过INTR引脚向CPU发出中断信号。8259A管理着多个IRQ线，每条线对应不同的设备。  **（2）中断处理过程：**  当一个外部设备发出中断请求时，8259A将该请求的状态存储在中断请求寄存器IRR中  8259A会检查中断屏蔽寄存器IMR中是否屏蔽了该请求，如果没有屏蔽，则发送中断信号给CPU  CPU在接收到中断信号后，会发送中断确认信号INTA给8259A，8259A根据优先级确定哪个中断要处理，并把中断向量号发送给CPU。  中断向量号告诉CPU要执行的中断服务程序ISR  **中断号的处理向量初始化：**  向端口20h或A0h写入ICW1（初始化控制字），初始化中断控制器  向端口21h或A1h写入ICW2，设定中断向量基地址，通常只需设定INT0号中断，其他中断号依次递增  向21h或A1h写入ICW3，设定主从控制器之间的连接关系  向21h或A1h写入ICW4，完成初始化配置  **3. 如何建立IDT，如何实现一个自定义的中断**  **（1）建立IDT：**  IDT 是中断描述符表,在保护模式下用于管理中断的一个数据结构，表中每一项对应一个中断向量。每一项中存放着中断服务程序的入口地址和相关信息。  IDT表项通常包含：中断处理程序的段选择符和偏移地址、描述符的类型（中断门、陷阱门等）以及特权级别  **（2）实现自定义的中断过程：**  在IDT中为自定义的中断分配一个向量号  编写对应的中断服务程序，处理你自定义的功能  将该中断服务程序的地址和段选择符填入IDT的相应项  启用这个中断，并确保其对应的中断号在8259A或其他中断控制器上被正确配置  之后在自己写的汇编代码上调用自定义中断，看是否达到预期  我们组实现的自定义中断源代码： ，具体作用参见实验步骤  **4. 如何控制时钟中断，为什么时钟中断时候，没有看到int的指令？**  **（1）控制时钟中断：**  时钟中断由8253/8254计时器芯片发出，通常对应IRQ0，8259A会捕捉到该中断，并把相应的中断向量号发送给CPU  通过编程8259A的IMR寄存器，可以屏蔽或允许时钟中断  调整计时器芯片的控制寄存器可以修改时钟中断的频率  **（2）为什么没有看到int指令：**  时钟中断是由硬件触发的，而非软件通过int指令发起。因此，当计时器芯片触发中断时，CPU会通过硬件中断处理机制自动调用相应的中断服务程序，而不是通过软件的int指令显式调用。  **5. 简要解释一下IOPL的作用与基本机理IOPL——I/O特权级别：**  **（1）作用：**  IOPL 是x86架构中EFLAGS寄存器中的两个比特位，决定了当前任务的I/O访问权限。I/O操作包括直接与硬件设备交互（如通过in、out指令访问端口）。  作用：在特权级别低的情况下（如用户模式），I/O访问被限制。如果程序需要直接与硬件交互，则必须运行在内核模式下，或者通过系统调用来实现间接访问  **（2）基本机理：**  IOPL设置为较低级别（如3级）时，用户模式程序可以执行I/O指令；而IOPL设置为较高的级别（如0级）时，仅内核模式才有权限执行I/O指令。这一机制确保了I/O操作的安全性和系统的稳定性  **改进意见：**  1.可以写一个功能更复杂的中断，即键盘输入中断，主要功能如下：处理多个键：扩展代码以处理组合键和特殊字符。实时输入显示：在屏幕上实时显示用户输入的内容。支持不同键盘布局：允许用户选择不同的键盘布局进行输入  2.优化字符映射：目前的字符映射逻辑是硬编码的。可以考虑使用查找表或者更复杂的映射算法，以便更准确地转换扫描码为字符，并支持特殊键（如功能键、控制键等）  3.加入输入缓冲区为了提升用户体验，可以实现一个输入缓冲区，将用户输入的字符存储起来，允许用户查看和编辑输入的内容  4.增强错误处理：当前的中断处理程序未考虑错误情况，例如扫描码的无效输入。建议加入错误处理机制，并在屏幕上给出相应提示 | | | | | |
|  | | | | | |
| 1. 各人实验贡献与体会（每人各自撰写） | | | | | |
| 实际上每个人都各自独立地完成了实验  1.王亚鹏：全程参与实验，与杨依磊合作编写自定义中断test.asm  **1. 对中断机制的理解**  通过本次实验，对中断的概念和其在操作系统中的重要性有了更加直观的理解。通过配置 IDT来将中断向量和具体的处理程序关联起来，我理解到中断系统是如何为操作系统提供快速响应机制的。中断机制使得 CPU 能够及时应对外部设备的请求，从而在系统中实现硬件与软件的有效协作。通过 EOI信号通知中断控制器中断已处理完毕，我也了解到了异常和中断的区别  **2. 操作显存以及打印文字的体验**  在自定义函数中，彩虹条带的实现是通过直接操作显存完成的。使用 gs 段寄存器直接访问内存，把不同颜色的字符写入屏幕指定位置。PrintLine 函数通过循环的方式将字符 \* 和颜色属性一起写入显存，逐行打印不同颜色的条带，实现彩虹效果。经过查阅资料得知不同颜色的代码  **3. 实现自定义中断的**  实现彩虹条带的过程中，最大挑战在于颜色属性和显存的配合。每个字符的显示在显存中占用两个字节：一个字节用于存储字符，另一个字节用于存储颜色属性。在 PrintLine 函数中，我们需要准确地调整显存的偏移位置（di 寄存器），并将字符和颜色属性组合后写入显存。通过这一过程，学会了如何通过汇编代码实现精准的内存操作，并在显示器上呈现出预期的图案  2.杨依磊：全程参与实验，与王亚鹏合作编写自定义中断test.asm  1. 中断与异常机制的理解  在实验的初始部分，通过调试8259A中断控制器和时钟中断例程，我了解了中断处理机制。中断机制允许CPU在处理当前任务时响应硬件请求，这种优先级机制提高了系统的实时性和响应速度。在实验中，通过观察8259A的初始化及其响应流程，我加深了对中断请求寄存器（IRR）、中断服务寄存器（ISR）以及中断屏蔽寄存器（IMR）的理解。  2. 8259A的编程与调试  8259A编程部分让我理解了如何设置主、从片的工作模式，并通过ICW和OCW命令字的配置来控制中断的屏蔽与响应顺序。尤其是在调试时钟中断时，通过观察中断请求到处理器实际执行的过程，我理解了中断向量的初始化及中断服务程序的响应过程。  3. 自定义中断向量的实现  我选择了一个未使用的中断向量0x81，并设计了一个复杂的中断处理程序——打印彩虹旗。该处理程序在屏幕上显示了彩虹旗的六种颜色（实际上是五种，我只能查到红，黄，绿，蓝，紫色的码），并通过操作视频内存直接控制字符颜色和显示位置。这不仅让我掌握了如何手动管理中断向量表（IDT），还让我掌握了视频内存的操作方式。  实验过程中的问题  3.杜泓波：全程参与实验  在本次实验中，我对中断和异常的机制以及8259A中断控制器的编程有了更加深入的理解，并通过调试时钟中断例程进一步掌握了系统中断的处理流程。  我在理解中断与异常的过程中，清晰区分了两者的工作原理。中断通常来自外部设备，是异步触发的，而异常则是由于程序本身执行中的错误或特定条件而引发，属于同步事件。通过实验，我学会了如何使用中断向量表（IVT）来处理不同类型的中断与异常。在调试过程中，尤其是观察中断嵌套和优先级的问题，我加深了对中断屏蔽和优先级机制的理解。  其次，在8259A的调试中，经历了从ICW（初始化命令字）设置到操作控制字（OCW）的配置。通过实际代码的编写与调试，我掌握了如何通过8259A控制器管理多个中断请求，并对中断服务程序（ISR）的编写有了实操经验。通过调整ICW2来设定不同的中断向量基址，也让我理解了中断地址的重定位原理。实验中我遇到的一些问题是未正确初始化8259A，导致中断信号无法正确发送。通过排查寄存器配置和观察中断请求线的状态，最终成功解决了该问题。  最后，在时钟中断的实验中，我学习了如何通过8253/8254计时器产生定时中断，并在中断服务程序中更新系统时钟。这个部分让我理解了操作系统如何利用定时中断实现多任务调度和时间片轮转。调试时，我通过在ISR中加入调试输出，确认中断是否按预期的频率触发，并通过观察系统时钟变量的变化验证了时钟中断处理的正确性。在多次实验后，我学会了调优时钟中断频率，以更好地匹配实际系统的需求。  **由于汇编语言的代码不容易理解，我查阅了C语言资料，得到了C语言代码，这样更容易阅读理解**  void init\_8259A() {  outb(0x20, 0x11); // ICW1：边沿触发、级联  outb(0x21, 0x20); // ICW2：中断向量基址0x20  outb(0x21, 0x04); // ICW3：设置主从关系  outb(0x21, 0x01); // ICW4：x86模式  }  void timer\_interrupt() {  // 更新系统时钟  system\_ticks++;  // 切换任务或执行其他定时任务  }  void init\_timer() {  // 设置8253/8254计时器  outb(0x43, 0x36); // 模式选择：模式3  outb(0x40, 11932 & 0xFF); // 低字节  outb(0x40, 11932 >> 8); // 高字节  }  通过这次实验，我巩固了对中断与异常机制的理论理解，提高了在硬件编程和系统底层中断管理方面的能力。  4.侯名扬：全程参与实验  我通过编写程序初始化8259A中断控制器，观察到当硬件设备（如定时器）发出中断请求时，IRR会首先记录该请求，随后处理器会从IRR中选择最高优先级的请求，进入对应的中断服务程序。ISR用于跟踪当前正在处理的中断，而IMR可以控制是否允许某些中断。  我实现了一个时钟中断处理程序。通过配置8259A和IDT（中断描述符表），使得定时器芯片发出的时钟中断能够正确触发。观察了中断请求到处理器执行中断服务程序的整个过程，确认中断处理过程顺利完成。  在编写自定义中断服务程序时，操作视频内存的过程中出现了颜色显示错误。经过调试发现，问题在于颜色代码的设置不正确，重新查阅了VGA模式下的颜色编码后，解决了这一问题。  通过此次实验，我系统地掌握了中断和异常处理机制的基本原理，了解了8259A中断控制器的工作原理，并成功实现了自定义中断服务程序。通过Bochs调试工具的使用，我进一步加深了对硬件中断和CPU响应过程的理解。这为我今后实现更加复杂的中断管理和操作系统的开发奠定了基础 | | | | | |
| 1. 教师评语   （实验报告的考评：依据实验内容完整度、实验步骤清晰度、实验结果与分析正确性、实验心得与思考的全面性、实验报告文档的规范性等五个维度综合考评）   |  |  | | --- | --- | | 85-100 | * 实验内容完整或者有超出课程实验大纲的内容； * 实验步骤详尽，能够体现完整的实验过程； * 实验结果正确且实验数据分析得当； * 实验心得与思考全面并且有自己的独立思考； * 实验报告文档规范、排版整齐。 | | 75-84 | * 实验内容较为完整； * 实验步骤较为详尽，能够体现实验过程； * 实验结果正确且实验数据分析较为得当； * 实验心得与思考全面； * 实验报告文档规范、排版较为整齐。 | | 60-74 | * 实验内容有缺失； * 实验步骤不够详尽，不能够体现完整的实验过程； * 实验结果部分正确； * 实验心得与思考无或者不够深入； * 实验报告文档规范性有待增强。 | | 60以下 | * 实验内容严重缺失、实验态度不够端正 * 实验步骤不够详尽，不能够体现完整的实验过程； * 实验结果部分正确； * 实验心得与思考无或者不够深入； * 实验报告文档规范性有待增强。 | | | | | | |
|  | | | | | |
| **教师评分（请填写好姓名、学号）** | | | | | |
| 姓名 | | 学号 | | 分数 | |
| 王亚鹏 | | 2022302181161 | |  | |
| 杨依磊 | | 2022302181159 | |  | |
| 杜泓波 | | 2022302181162 | |  | |
| 侯名扬 | | 2022302181165 | |  | |
| 教师签名：  年 月 日 | | | | | |