|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 | |  |
| 实验名称 |  | | 实验周次 | |  |
| 姓名 | 学号 | | 专业 | | 班级 |
| 王卓 | 2021302191791 | | 网络空间安全 | | 9班 |
|  |  | |  | |  |
|  |  | |  | |  |
|  |  | |  | |  |
| 1. 实验目的及实验内容   （本次实验所涉及并要求掌握的知识；实验内容；必要的原理分析） | | | | | |
|  | | | | | |
| 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；具体的实验步骤） | | | | | |
| 1. **前期准备：实验环境的配置**    1. 下载freedos，将解压后的a.img命名为freedos.img并复制到工作目录下，用bximage生成一个软盘映像pm.img   20231016155217  **图x 前期准备**   * 1. 修改bochsrc   20231016160751  **图x 修改bochsrc**   * 1. 启动bochs，待freedos启动完毕后格式化B盘   20231016161223  **图x 启动bochs并格式化B盘**  这里的B盘为我们的pm.img镜像文件，格式化操作相当于将其转化为FAT16文件系统，我们通过终端查看pm.img内容时，可以看到其中已经写入了一些文件系统相关的东西。  20231016161351  **图x 查看pm.img中内容**   * 1. 修改pmtest1.asm代码中的07c00h为0100h，并重新编译   命令如下：   1. nasm pmtest1.asm -o pmtest1.com  *//汇编* 2. 将pmtest1.com复制到虚拟软盘pm.img上 3. sudo mkdir /mnt/floppy  *//创建空目录* 4. sudo mount -o loop pm.img /mnt/floppy  *//挂载pm.img* 5. sudo cp pmtest1.com /mnt/floppy/  *//拷贝* 6. sudo umount /mnt/floppy  *//取消挂载*   20231016162304  **图x 修改pmtest1.asm代码**   * 1. 重新启动bochs   到freedos中执行下列命令：   1. B：\pmtest1.com   20231016161928  **图x 环境配置完成展示**   1. **实验内容** 2. **认真阅读章节资料，掌握什么是保护模式，弄清关键数据结构：GDT、descriptor、selector、GDTR的含义，及其之间关系，阅读pm.inc文件中数据结构以及含义，写出对宏Descriptor的分析。**   略...     1. **调试代码，/a/掌握从实模式到保护模式的基本方法，画出代码流程图，如果代码/a/中，第71行有dword前缀和没有前缀，编译出来的代码有区别么，为什么，请调试截图。**    * 1. 实模式到保护模式的步骤：         1. 准备GDT;         2. 用lgdt加载gdtr；         3. 打开A20；         4. 置寄存器cr0的PE位；（给出cr0的结构示意图）         5. 跳转，进入保护模式      2. 有无dword：         1. 前期准备：   为了实现调试功能，我们在bochsrc中添加magic\_break: enabled=1  20231017100628  并在jmp指令前添加命令xchg bx, bx实现断点的效果。   * + - 1. 有dword：   运行结果：  20231016164405  **图x 有dword的运行结果展示**  查看编译后的二进制文件内容：  20231018124324  **图x 有dword的编译后代码**  调试查看寄存器内容：  20231017101916  **图x 有dword的调试截图**   * + - 1. 无dword:   运行结果：  20231016200613  **图x 无dword的运行结果展示**  查看编译后的二进制文件内容：  20231018124458  **图x 无dword的编译后代码**  调试查看寄存器内容：  20231017102456  **图x 无dword的调试截图**  综上内容，对比发现：是否删除dword，两个可执行文件在jmp指令执行前后的寄存值相同，所以运行产生的结果相同，但实际上该问是由于跳转偏移量为0，尽管发生偏移的截断也仍旧为0，不会产生影响，但如果目标地址的偏移量较大，就会产生问题。   1. **调试代码，/b/，掌握GDT的构造，体会保护模式下地址空间的变化、从保护模式切换回实模式方法**   20231016171056  可以看到，程序打印出两行数字，第一行全部是零，说明内存地址5MB处都是0，而下一行变成了41 42 43...，即十六进制的A、B、C、...、H，说明写操作成功。同时，程序执行结束后不再像上一个程序一样进入死循环，而是重新出现DOS提示符B:\>，表明我们重新回到了实模式下的DOS。   1. **调试代码，/c/，掌握LDT的构造**   20231016172158  在[SECTION .S32]中打印完“In Protect Mode now”字符串后，出现一个红色的字符L。   1. **调试代码，/d/掌握一致代码段、非一致代码段、数据段的权限访问规则，掌握CPL、DPL、RPL之间关系，以及段间切换的基本方法。**   20231016172428   1. **调试代码，/e/掌握利用调用门进行特权级变换的转移的基本方法**   20231016172612   1. **从实模式到保护模式，关键步骤有哪些？为什么要关中断？为什么要打开A20地址线？从保护模式切换回实模式，又需要哪些步骤？**    1. 实模式到保护模式的关键步骤：       1. 初始化GDT描述符；       2. 加载gdtr；       3. 打开A20地址线；       4. 设置寄存器CR0的PE位为1，使之运行于保护模式；       5. 执行跳转指令，让系统进入保护模式    2. 为什么要进行关中断：   关中断是为了确保在切换到保护模式时，不会发生中断处理程序的执行。这是因为在实模式下，中断处理程序的地址是通过中断向量表中的中断向量来确定的，而在保护模式下，中断处理程序的地址是通过中断描述符表（IDT）中的中断门来确定的。如果在切换过程中发生中断，处理器可能会尝试执行错误的中断处理程序，导致系统异常。   * 1. 为什么要打开A20地址线：   A20地址线是用于扩展地址总线的一条线路。在实模式下，8086处理器只能访问1MB的物理内存空间，而在保护模式下，处理器可以访问更大的内存空间。打开A20地址线可以使处理器能够访问超过1MB的内存，以充分利用保护模式下的内存管理功能。   * 1. 保护模式切换回实模式的步骤：      1. 关中断：与从实模式到保护模式的切换一样，切换回实模式前需要关中断。      2. 清除控制寄存器：将控制寄存器CR0的第0位清零，以禁用保护模式。      3. 刷新段选择子：在切换回实模式后，需要刷新所有段选择子，以便它们指向实模式下的段描述符。      4. 重新加载段寄存器：由于实模式下的段寄存器与保护模式下的段寄存器不同，需要重新加载段寄存器，以确保正确访问内存。      5. 关闭A20地址线：如果在保护模式下打开了A20地址线，切换回实模式时可能需要关闭它，以限制对1MB内存空间的访问。  1. **阐述不同权限代码的切换方法，call, jmp，retf使用场景如何，能够互换吗？**   切换方法：   * 1. call指令：call指令用于调用一个过程或函数，并将控制权转移到目标代码段中的指定地址。当使用call指令时，处理器会将当前代码段的返回地址（即下一条指令的地址）压入堆栈，并跳转到目标代码段中的指定地址。call指令可以在不同的权限级别之间进行切换，例如从用户态（Ring 3）切换到内核态（Ring 0）。   2. jmp指令：jmp指令用于无条件跳转到目标代码段中的指定地址。与call指令不同，jmp指令不会将返回地址压入堆栈，因此无法直接实现权限级别的切换。但是，通过在目标代码段中设置适当的段选择子，可以实现从一个权限级别的代码段跳转到另一个权限级别的代码段。   3. retf指令：retf指令用于从过程或函数返回，并将控制权转移到调用者的代码段中的指定地址。retf指令会从堆栈中弹出返回地址，并跳转到该地址。与call指令类似，retf指令可以在不同的权限级别之间进行切换，例如从内核态（Ring 0）返回到用户态（Ring 3）。   使用场景：   * 1. call指令：call指令通常用于调用子程序或函数，将控制权转移到另一个代码段，并在返回时继续执行调用指令的下一条指令。它常用于实现函数调用、子程序调用和中断处理等场景。   2. jmp指令：jmp指令通常用于实现条件跳转或无条件跳转，将控制权直接转移到目标代码段中的指定地址。它常用于实现循环、条件语句和跳转表等场景。   3. retf指令：retf指令通常用于从过程或函数返回，并将控制权转移到调用者的代码段中的指定地址。它常用于实现函数返回、中断返回和任务切换等场景。   这些指令在某些情况下可以互换使用。例如，可以使用Jmp指令实现从一个权限级别的代码段跳转到另一个权限级别的代码段，然后使用Retf指令返回到原来的权限级别。   1. **课后动手改**    1. **自定义添加1个GDT代码段、1个LDT代码段，GDT段内要对一个内存数据结构写入一段字符串，然后LDT段内代码段功能为读取并打印该GDT的内容；**   对于该问题本人参考pmtest2.asm和pmtest3.asm文件进行修改，本人拟打印的字符串为自己的名字，具体步骤如下：   1. 修改[SECTION .data1]数据段中内容   将StrTest中的内容修改为StrTest: db “Wang Zhuo”, 0   1. 编写GDT代码段   在pmtest3.asm的[SECTION .S32]代码段基础上，call DispReturn和；Load LDT之间加上call TestRead和call TestWrite  20231018130928  TestRead代码用来实现从内存中读取数据并显示在屏幕上，如下所示：  20231018131400  TestWrite代码用来实现将字符串数据写入内存，如下所示：  20231018131524   1. 编写LDT代码段，用来读取GDT内容并打印   20231018132931   1. 编译执行   20231018132024  **图x 课后动手改1(王卓)**  整个代码运行流程为：   1. 从实模式jmp到保护模式进入[SECTION .s32]，显示字符串In Protect Mode now. ^-^； 2. [SECTION .s32]中通过lldt加载LDT的ldtr，然后跳转到[SECTION .la]； 3. [SECTION .la]段显示字符串StrTest即Wang Zhuo，然后jmp回实模式。    1. **自定义3个GDT代码段A、B、C，B与A、C分属于不同特权级，功能自定义，要求实现A-->B的跳转，以及B-->C的跳转。**   洋写了 | | | | | |
| 1. 实验过程分析   （实验分工，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） | | | | | |
| **实验问题与故障分析：**   1. 没有完全理解不同特权级之间跳转的过程，导致代码中止运行   具体来说，我没有正确理解如何利用调用门进行低特权级到高特权级的转换，因此无法实现三个代码段之间的跳转。我尝试使用直接的call指令进行跳转，但结果却是跳转失败。这个错误和程子洋同学出现的情况一致，即我只能出现“A This is A”。  为了解决这个问题，我进一步学习如何使用调用门和call指令进行特权级之间的跳转。经过仔细研究并和程子洋同学交流，修改代码，我成功地完成了实验，并实现了代码段之间的跳转。  注：此处由于实验中忘记对错误结果截图，故无图展示 | | | | | |
| 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） | | | | | |
| 王卓：   1. 提供更详细的实验指导：在每个实验步骤中，提供更详细的指导和说明，比如预期结果等，这样能帮助我们更好地完成实验； 2. 介绍实验目的和背景：实验开始之前提供实验的目的和背景，解释一下为什么需要进行该实验以及其与操作系统的关系，这能帮助我们更好理解实验的意义和重要性。 | | | | | |
| 1. 各人实验贡献与体会（每人各自撰写） | | | | | |
| 同学：王卓  此次实验为本人独立完成大部分实验内容，并主要针对实验题7，8以及课后动手改1进行对应部分内容的实验报告的撰写。这次实验给我的体会如下：   1. 深入理解保护模式：通过实验，我深入理解了保护模式的概念和工作原理。我学会了如何构造和加载GDT表、LDT表，以及如何进行特权级的切换和段间的切换。这使我对操作系统的内存管理和权限控制有了更深入的理解。 2. 团队合作与交流：在实验过程中，我与同学们进行了讨论和交流，分享了彼此的经验和问题。这促进了团队合作和互助，使我们能够共同解决实验中遇到的困难和挑战。这种合作与交流的经验对我今后的团队合作能力和沟通能力有着积极的影响。 3. 实验的挑战与收获：实验内容相对复杂，需要深入理解和掌握多个概念和技术。在实验过程中，我遇到了一些困难和挑战，但通过不断的学习和尝试，我成功地完成了实验，并从中获得了很多收获和成就感。 | | | | | |
| 1. 教师评语 | | | | | |
|  | | | | | |
| **教师评分（请填写好姓名、学号）** | | | | | |
| 姓名 | | 学号 | | 分数 | |
|  | |  | |  | |
|  | |  | |  | |
|  | |  | |  | |
|  | |  | |  | |
| 教师签名：  年 月 日 | | | | | |