第二次试验问题清单

By 张洪胤

一、 PPT内容

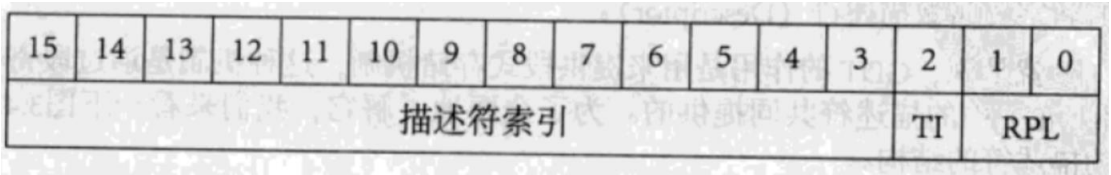
问题一：什么是实模式，什么是保护模式？

实模式使用基地址加偏移量的方式就可以直接拿到物理地址的模式。

保护模式是不能直接拿到物理地址的模式，需要进行地址转换换。

问题二：什么是选择子？

选择子总共16位，存放在段选择寄存器中，低2位表示请求特权级，第3位表示选择GDT方式还是LDT方式，高13位表示在描述符表中的偏移。



问题三：什么是描述符？

保护模式下引入描述符来描述各种数据段，所有的描述符均为8个字节(0-7)，由第5个字节说明描述符的类型。类型不同，描述符的结构也有所不同。

图 3.4 代码段和数据段描述符

问题四：什么是GDT，什么是LDT?

GDT是全局描述符表，是全局唯一的。存放一些公有的描述符和包含各进程局部描述符表首地址的描述符。

LDT是局部描述符表，每个进程都可以有一个。存放本进程中使用的描述符。

问题五：请分别说明GDTR和LDTR的结构。

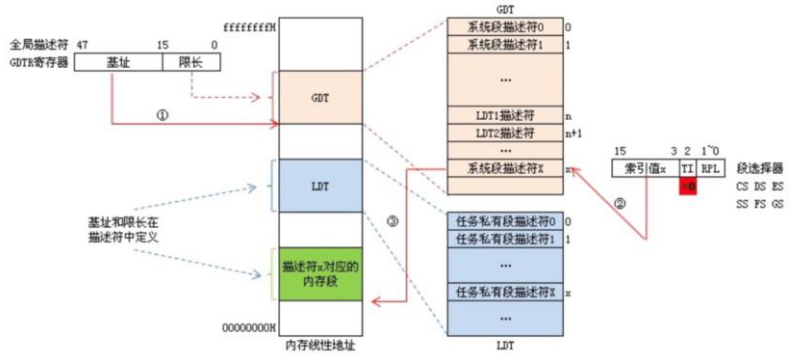
GDTR：48位寄存器，高32位放置GDT首地址，低16位放置GDT限长，限长决定了可寻址的大小，注意低16位放的不是选择子）

LDTR：16位寄存器，放置一个特殊的选择子，用于查找当前进程的LDT首地址。

问题六：请说明GDT直接查找物理地址的具体步骤。

1. 给出段选择子（放置在段选择寄存器中）+ 偏移量
2. 若选择了GDT方式，则从GDTR获取GDT首地址，用段选择子中的13位做偏移，拿到GDT中的描述符
3. 如果合法且有权限，用描述符中的段首地址加上1中的偏移量找到物理地址，寻址结束。

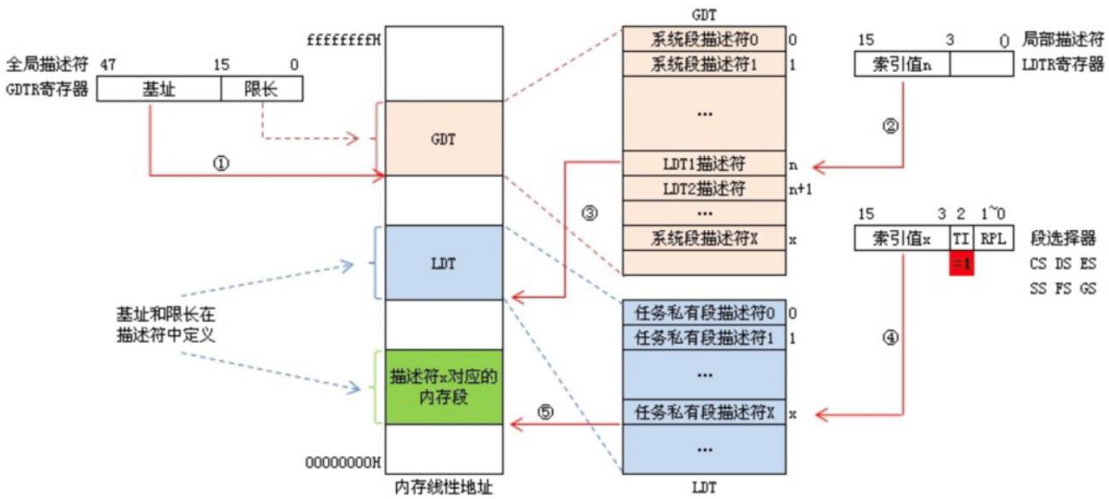
图解：GDT查询物理地址



问题七：请说明通过LDT查找物理地址的具体步骤。

1. 给出段选择子（放置在段选择寄存器中）+偏移量
2. 若选择了LDT方式，则从GDTR获取GDT首地址，用LDTR中的偏移量做偏移，拿到GDT中的描述符1
3. 从描述符1中获取LDT首地址，用段选择子中的13位做偏移，拿到LDT中的描述符2
4. 如果合法且有权限，用描述符2中的段首地址加上1中的偏移量找到物理地址，寻址结束

图解：LDT查找物理地址



问题八：根目录区大小一定么？扇区号是多少？为什么？

不一定，是通过BPB RootEntCnt来计算根目录的大小，根目录区的占用扇区数时BPB RootEntCnt * 0x20（一个目录项占0x20个字节）/BPB BytePerSec（一个扇区的字节数，比如0x200），一般是19号。

问题九：数据区第一个簇号是多少？为什么？

根据根目录大小计算，DataSectors = RootDirSectors + 引导扇区（1）+FAT1（9）+FAT2（9），一般值是33。

数据区的第一个簇是2，因为1.44M的软盘上，FAT前三个字节的值是固定的0xF0、0xFF、0xFF用于表示这是一个应用在1.44M软盘上的FAT12文件系统。本来序号为0和1的FAT表项应该对应簇0和簇1，但是由于这两个表项都是被设置为固定值，簇0和簇1没有存在意义。

问题十：FAT表的作用？

文件分配表被划分为紧密排列的若干个表项，每个表项都与表项都与数据区中的一个簇相对应，而且表项的序号也是与簇号一一对应。

问题十一：解释静态链接的过程。

静态链接是指在编译阶段直接把静态库加入到可执行文件中去，这样子可能会导致可执行文件比较大。

1.空间和地址分配：每一个.o文件都有自己的段属性，比如.text和.data等，链接的第一步就是将这些段属性合并在一起。

2.符号解析和重定位：重定位(修正地址)、重定位表(相对地址修正)

https://blog.csdn.net/weixin_41485748/article/details/96702672

问题十二：解释动态链接的过程。

动态链接是指链接阶段仅仅加入一些描述信息，而程序执行时再从系统中把相应动态库加载到内存中去。

1.**动态链接器自举**

动态链接器本身也是一个不依赖其他共享对象的共享对象，需要完成自举。

2.**装载共享对象**

将可执行文件和链接器自身的符号合并成为全局符号表，开始寻找依赖对象。加载对象的过程可以看做图的遍历过程；新的共享对象加载进来后，其符号将合并入全局符号表；加载完毕后，全局符号表将包含进程动态链接所需全部符号。

3.**重定位和初始化**

链接器遍历可执行文件和共享对象的重定位表，将它们GOT/PLT中每个需要重定位的位置进行修正。完成重定位后，链接器执行.init段的代码，进行共享对象特有的初始化过程（例如C++里全局对象的构造函数）。

4.**转交控制权**

完成所有工作，将控制权转交给程序的入口开始执行。

https://www.cnblogs.com/linhaostudy/p/10544917.html

问题十三：静态链接相关PPT中为什么使用ld链接而不是gcc

gcc被称为gcc工具链，包含了很多的工具，其中用于连接的是ld，ld是binutils工具集的底层部件，所以工作会在汇编级别

问题十四：linux下可执行文件的虚拟地址空间默认从哪里开始分配。

0x08048000

二、 实验相关内容

问题一：BPB指定字段的含义

表 4.1 FAT12 引导扇区的格式

名称	偏移	长度	内容	Orange'S的值
BS_jmpBoot	0	3	一个短跳转指令	jmp LABEL_START nop
BS_OEMName	3	8	厂商名	'ForrestY'
BPB_BytsPerSec	11	2	每扇区字节数	0x200
BPB_SecPerClus	13	1	每簇扇区数	0x1
BPB_RsvdSecCnt	14	2	Boot 记录占用多少扇区	0x1
BPB_NumFATs	16	1	共有多少 FAT 表	0x2
BPB_RootEntCnt	17	2	根目录文件数最大值	0xE0
BPB_TotSec16	19	2	扇区总数	0xB40
BPB_Media	21	1	介质描述符	0xF0
BPB_FATSz16	22	2	每 FAT 扇区数	0x9
BPB_SecPerTrk	24	2	每磁道扇区数	0x12
BPB_NumHeads	26	2	磁头数（面数）	0x2
BPB_HiddSec	28	4	隐藏扇区数	0
BPB_TotSec32	32	4	如果BPB_TotSec16是0，由这个值记录扇区数	0
BS_DrvNum	36	1	中断 13 的驱动器号	0
BS_Reserved1	37	1	未使用	0
BS_BootSig	38	1	扩展引导标记（29h）	0x29
BS_VolID	39	4	卷序列号	0
BS_VolLab	43	11	卷标	'OrangeS0.02'
BS_FileSysType	54	8	文件系统类型	'FAT12'
引导代码及其他	62	448	引导代码、数据及其他填充字符等	引导代码（剩余空间被 0 填充）
结束标志	510	2	0xAA55	0xAA55

问题二：如何进入子目录并输出（说明方法调用）

类似人查找文件，我将文件夹、文件和镜像文件统一继承自Item项，如果当前Item项的名字符合本级名称，并且为文件夹则调用其getSubItems方法来获取其子目录列表，等同于进入子目录。

问题三：如何获得指定文件的内容，即如何获得数据区的内容（比如使用指针等）

首先通过文件系统，找到包含有文件相关信息和指针的项

FileItem，然后调用readContent()方法来获取文件内容。

如果簇号>=0xFF0则出现了坏簇，终止。如果簇号>=0xFF8，则说明已经是最后一个簇了。

从初始簇号开始，从数据区读取一个簇大小的字符(簇号-2)，然后调用nextCluster方法从FAT表中找到对应的下一个簇的位置，直到终止。最后返回对应长度的字符。

问题四：如何进行C代码和汇编之间的参数传递和返回值传递

参数传递，可以通过栈esp完成，C调用时的函数参数会被放置在esp中，然后汇编从esp+4开始，逐一拿出参数。

汇编语言的返回值，放置在eax中进行返回。

https://blog.csdn.net/pk_20140716/article/details/103177828

问题五：汇编代码中对 I/O 的处理方式，说明指定寄存器所存值的含义

```
global print

; print(pointer: char*): void
print:
    mov ecx, dword[esp + 4]
    mov eax, ecx
    ; edx 是长度
    mov edx, 0

strLength:
    cmp [ecx], byte 0
    je strFinished
    inc edx
    inc ecx

    jmp strLength

strFinished:
    mov ecx, eax
    mov eax, 4
    mov ebx, 1
    int 80h
    ret
```

我使用 global 将 print 函数声明为全局，使用 ecx 存放传递进来的需要打印字符串首地址，然后将 edx 存放需要打印的长度，eax 存放系统调用号为 4，ebx 表示标准输出 std-out。