1. **什么是实模式，什么是保护模式?**

   实模式就是用基地址加偏移量就可以直接拿到物理地址的模式;

   保护模式就是不能直接拿到物理地址的模式,需要进行地址转换.

2. **什么是选择子?**

   描述符在描述符表中的相对偏移。

   1. 选择子共16位，放在段选择寄存器里
   2. 低2位表示请求特权级
   3. 第3位表示选择GDT还是LDT方式
   4. 高13位表示在描述符表中的偏移（故描述符表的项数最多是2的13次方）

3. **什么是描述符?**

   描述一个段是否在内存中；保护模式下引入描述符来描述各种数据段，描述符为8字节，有第五个字节说明描述符的类型。

4. **什么是GDT，什么是LDT?**

   GDT：全局描述符表，全局唯一。存放公用的描述符、和包含各进程局部描述符表首地址的描述符；

   LDT：局部描述符表，每个进程都可以有一个，存放本进程内使用的描述符。

5. **请分别说明GDTR和LDTR的结构。**

   GDTR：48位寄存器，高32位放GDT首地址，低16位放GDT限长（限长决定了可寻址的大小）

   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

6. **请说明GDT直接查找物理地址的具体步骤。**

   （1）给出段选择子（放在段选择寄存器里）+ 偏移量

   （2）若选择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。

7. **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。

8. **根目录区大小一定么？扇区号是多少？为什么?**

不一定；19；1(引导扇区) + 9(FAT1) + 9(FAT2) = 19

9. **数据区第一个簇号是多少？为什么？**

   第一个簇号为2.在1.44M软盘上，FAT前三个字节的值必须是固定的，分别是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表项应该对应于簇0和簇1，但是由于这两个表项被设置成了固定值，簇0和簇1就没有存在的意义了，所以数据区起始于簇2。

10. **FAT表的作用?**

    记录硬盘中有关文件如何被分散存储在不同扇区的信息（也可以回答为了找到所有的簇(扇区)）

11. **解释静态链接的过程。**

    相似段合并；重定位

12. **解释动态链接的过程。**

    动态链接器自举；装载共享对象；重定位和初始化

13. **静态链接相关PPT中为什么使用ld链接而不是gcc**

    为了避免gcc进行glibc的链接

14. **linux下可执行文件的虚拟地址空间默认从哪里开始分配**

    从0x08048000开始分配

---

1. **BPB指定字段的含义（引导扇区包含数据和代码，数据为BPB（BIOS Parameter Block））**

Offset (decimal)	Offset (hex)	Size (in bytes)	Meaning
0	0x00	3	The first three bytes EB 3C 90 disassemble to JMP SHORT 3C NOP. (The 3C value may be different.) The reason for this is to jump over the disk format information (the BPB and EBPB). Since the first sector of the disk is loaded into ram at location 0x0000:0x7c00 and executed, without this jump, the processor would attempt to execute data that isn't code. Even for non-bootable volumes, code matching this pattern (or using the E9 jump opcode) is required to be present by both Windows and OS X. To fulfil this requirement, an infinite loop can be placed here with the bytes EB FE 90.
3	0x03	8	OEM identifier. The first 8 Bytes (3 - 10) is the version of DOS being used. The next eight Bytes 29 3A 63 7E 2D 49 48 and 43 read out the name of the version. The official FAT Specification from Microsoft says that this field is really meaningless and is ignored by MS FAT Drivers, however it does recommend the value "MSWIN4.1" as some 3rd party drivers supposedly check it and expect it to have that value. Older versions of dos also report MSDOS5.1, linux-formatted floppy will likely to carry "mkdosfs" here, and FreeDOS formatted disks have been observed to have "FRDOS5.1" here. If the string is less than 8 bytes, it is padded with spaces.
11	0x0B	2	The number of Bytes per sector (remember, all numbers are in the little-endian format).
13	0x0D	1	Number of sectors per cluster.
14	0x0E	2	Number of reserved sectors. The boot record sectors are included in this value.
16	0x10	1	Number of File Allocation Tables (FAT's) on the storage media. Often this value is 2.
17	0x11	2	Number of directory entries (must be set so that the root directory occupies entire sectors).
19	0x13	2	The total sectors in the logical volume. If this value is 0, it means there are more than 65535 sectors in the volume, and the actual count is stored in the Large Sector Count entry at 0x20.
21	0x15	1	This Byte indicates the media descriptor type.
22	0x16	2	Number of sectors per FAT. FAT12/FAT16 only.
24	0x18	2	Number of sectors per track.
26	0x1A	2	Number of heads or sides on the storage media.
28	0x1C	4	Number of hidden sectors. (i.e. the LBA of the beginning of the partition.)
32	0x20	4	Large sector count. This field is set if there are more than 65535 sectors in the volume, resulting in a value which does not fit in the *Number of Sectors* entry at 0x13.

每个扇区的字节数

每个簇的扇区数

Boot record占用的扇区数

FAT的数量，一般为2

根目录文件数的最大值

扇区数

FAT表的数量

36	0x024	1	Drive number. The value here should be identical to the value returned by BIOS interrupt 0x13, or passed in the DL register; i.e. 0x00 for a floppy disk and 0x80 for hard disks. This number is useless because the media is likely to be moved to another machine and inserted in a drive with a different drive number.
37	0x025	1	Flags in Windows NT. Reserved otherwise.
38	0x026	1	Signature (must be 0x28 or 0x29).
39	0x027	4	VolumeID 'Serial' number. Used for tracking volumes between computers. You can ignore this if you want.
43	0x02B	11	Volume label string. This field is padded with spaces.
54	0x036	8	System identifier string. This field is a string representation of the FAT file system type. It is padded with spaces. The spec says never to trust the contents of this string for any use.
62	0x03E	448	Boot code.
510	0x1FE	2	Bootable partition signature 0xAA55.

Boot代码

Magic number 0xAA55

Boot record占据了第一个扇区！

2. **如何进入子目录并输出（说明方法调用）**

   （递归方法）

3. **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）**

```
(ptr->startSec + 31) * 512;//数据区第一个簇为2号，文件数据起始于对应33扇区，+31关系
```

   cat中：

```
fseek(fat12, table[idx].startPos, 0);
fread(catContent, 1, 1024, fat12);
```

4. **如何进行C代码和汇编之间的参数传递和返回值传递**

   extern，global，esp参数地址获取参数，ret

5. **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**

   eax获取颜色值，设定好颜色，ecx问i输出内容，edx为长度。