2020年11月8日 星期日

问答题

- 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，非常不安全
  - 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，从 80386 开始是现代操作系统的主要模式

- 2. 什么是选择子?
  - 共 16 位，放在段选择寄存器里，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

- 3. 什么是描述符?
  - 共 32 位，包括段基址、段界限、段属性
  - 段基址 32 位，规定线性地址空间中段的开始地址
  - 段界限 20 位，规定段的大小
  - 段属性规定段的主要特性

- 4. 什么是 GDT ，什么是 LDT ?
  - GDT：全局描述符表，是全局唯一的。存放一些公用的描述符、 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 位放置 GDT 首地址，低 16 位放置 GDT 限长
  - 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、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、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，需要根据根目录项的最大个数计算
  - 19 号扇区，前面有 1 个引导扇区，2 个占 9 扇区的文件配置表

- 9. 数据区第一个簇号是多少?为什么?
  - 2 号，FAT 表中 0 号和 1 号的项是固定的，簇号 0 和 1 没有存在的意义

- 10. FAT 表的作用?
  - 记录文件所在位置的表格，FAT 项的值代表文件的下一个簇号

- 11. 解释静态链接的过程。
  - 空间和地址分配、符号解析和重定位

- 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、转交控制权

- 13. 静态链接相关 PPT 中为什么使用 ld 链接而不是 gcc
  - gcc 是工具链，用于链接的是 ld

- 14. Linux 下可执行文件的虚拟地址空间默认从哪里开始分配。
  - 0x08048000

- 1. BPB 指定字段的含义

名称	偏移	长度	内容	软盘参考值
BS_jmpBoot	0	3		jmp LABEL_START nop
BS_OEMName	3	8	厂商名	ForrestY'
BPB_BytsPerSec	11	2	每扇区字节数	0x200（即十进制512）
BPB_SecPerClus	13	1	每簇扇区数	1
BPB_RsvdSecCnt	14	2	Boot记录占用多少扇区	1
BPB_NumFATs	16	1	共有多少FAT表	2
BPB_RootEntCnt	17	2	根目录文件数最大值	0xE0 （224）
BPB_TotSec16	19	2	扇区总数	0xB40（2880）

BPB_Media	21	1	介质描述符	F0
BPB_FATSz16	22	2	每FAT扇区数	9
BPB_SecPerTrk	24	2	每磁道扇区数	12
BPB_NumHeads	26	2	磁头数	2
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	0xB40（2880）
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	29
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	62	448	引导代码、数据及其他填充字符等	
结束标志	510	2		AA55

- 2. 如何进入子目录并输出（说明方法调用）

  - getFileEntry(const string path, vector<FileEntry>* const fileEntries) 用来获取 path 路径下的文件项并保存在 fileEntries 中，如果是个目录就返回子文件，如果是文件则返回本身。首先把路径按“/”分割，从根目录开始，一层一层地判断当前项是否是存在的目录，是的话就读取当前项的文件项列表，再在这个列表中找下一个目录。迭代直到找到 path 或者返回不存在

- 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）

  - readFile(const string path, string* const content) 用来获取 path 这个文件的内容并保存到 content 中。先用判断 getFileEntry 判断这个文件是否存在且是个普通文件，再根据这个文件的大小以及下一个簇号，循环读取这个簇中的内容追加到 content 中

- 4. 如何进行 C 代码和汇编之间的参数传递和返回值传递
  - C 中的函数参数被入栈，在汇编中根据 esp 计算参数在栈中的位置
  - 返回值被放在 eax 中

- 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
  - 通过 esp 计算传入的参数位置，eax 存系统调用号，ebx 存文件描述符，ecx 存起始地址，edx 存长度