Lab5 笔记

2019年5月27日 21:24

Sector and Block

sector大小是磁盘的属性，而block size是系统的值，block size应该是sector大小的倍数，这个文件系统的block size取4K

Superblocks

这是一个特殊的block，记录了文件系统的block size，disk大小，以及定位根目录的元数据。该系统的super block为block 1。一般来说文件系统会维护多个super block，当一个block损坏后，其他的备份可以继续使用。

该文件系统，支持10个direct寻址和1个indirect寻址，所以一共支持1024+10=1034个block。

文件类型：该文件系统支持两种文件类型（1）普通文件（2）目录文件

Disk Access

要让文件系统能够有权限去访问disk，用EFLAGS中的IOPL位来表示是否允许保护模式下的代码能执行设备I/O操作。

Exercise1

- 修改env.c中的env_create函数，让文件系统环境有I/O权限。
- 文件系统环境类型标识为ENV_TYPE_FS
- 将eflags中的IOPL位置为1

The block Cache

文件系统有自己的地址空间，用内存来当disk的cache，我们不需要吧整个磁盘都读入内存，而是当有响应位置的访问时，产生一个page fault去读取。

Exercise2

- 实现bc_pgfault和flush_block函数
- Bc_pgfaullt是页错误处理函数，需要把该页对应的磁盘读入内存，用给定的接口ide_read读取，addr向下对齐
- Flush_block函数将block内容写回磁盘，检查该页是否有效，是否是dirty页，把addr向下对齐。

注意：一定记得取消ENV_CREATE(fs_fs)和close_all的注释。。。。。。否则测试不通过。

The block bitmap

文件系统用bitmap来记录每个block的分配情况，每个bit表示一个block的分配情况。

Exercise3

实现allock_block函数

该函数寻找一块空的block，将其对应的bit标记为Used，如果没找到返回-E_NO_DISK

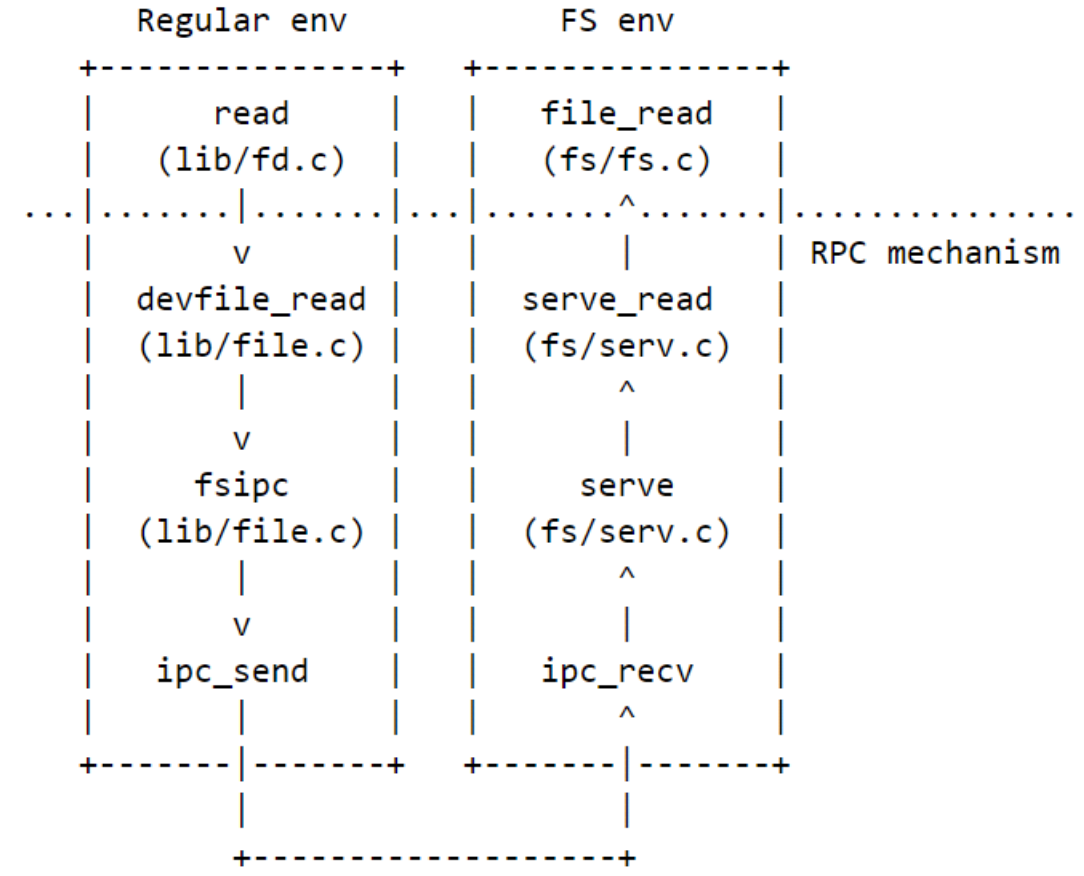
标记bitmap后应该立即调用flush_block保证文件系统的一致性

Exercise4

- 实现file_block_walk和file_get_block函数
- file_block_walk：找到f文件内filebno号block
  - 如果小于NDIRECT，直接获取
  - 如果大于等于NDIRECT，先查看INDIRECR block是否已经分配，如果没有，并且alloc为false，则不分配，
  - 否则现在分配block，在二级block中找到对应的地址。
  - 如果INDIRECT是新分配的，把该block全部置0，以便file_get_block函数可以判断是否需要分配新的block
- File_get_block：查找file中第fileno个block的地址
  - 调用file_get_block得到block number在file内存中的地址
  - 如果block number为0，分配一块新的block，并写入file中。
  - 将blk指向目标block对应的在内存中的地址

The file system interface

如图，文件系统提供RPC机制为其它环境提供操作文件的接口。



Exercise5

- 实现serve_read
- 用file_read读取文件内容
- 刷新fd的offset

注：此处对lab4进行了修改，权限检查有点问题，将debug变量置为1方便进行debug

Exercise6

- 实现serve_write和devfile_write
- Serve_write
  - Openfile_lookup打开文件，调用file_write写文件
  - 更新fd的offset
- Devfile_write
  - 设置fsipcbuf的fileid和req_n
  - 复制buf内容到fsipcbuf，
  - 利用fssipc调用serve_write

Exercise7

- spawn产生新环境，新环境需要sys_env_set_trapframe来初始化
- 实现sys_env_set_trapframe
- 把tf参数赋值给指定环境
- 把tf_cs设置为level3
- 用FL_IF设置interrupt enabled
- 最后在syscall函数内加上分发
- 修改init.c运行spawhello，make qemu-nox，成功。

Exercise8

- 修改duppage函数，如果页表PTE_SHARE被标记，直接复制拷贝映射。
- 实现copy_shared_pages函数，从UTEXT到USTACKTOP，复制PTE_P PTE_U和PTE_SHARE的页到子进程

The keyboard interface

为了让shell工作，需要一种输入方式，目前我们只能在monitor上输入，kern/console.c包含了键盘和一系列drivers。

Exercise9

- 修改trap.c，调用kbd_intr来handle IRQ_OFFSET+IRQ_KBD和serial_intr handle trap中的IRQ_OFFSET+IRQ_SERIAL
- 在trap_dispatch中分发这两个trap类型，并调用指定函数

Exercise10

- 实现重定向功能，在user/sh.c中添加i/o重定向<
- 在runcmd函数中，代码已经写好了解析<字符，但没写具体重定向实现。
- 具体实现可参考该函数内输出重定向>字符的实现
- 用open打开t文件，如果得到的文件描述符不是0，则用dup函数复制重定向到0，并关闭

以前的fd。

注：debug后，一定要记得把debug宏定义修改为0，不然运行超级慢，程序跑不过。。。。。