

第四次实例分析主要关注操作系统中文件系统的实现，xv6 的文件系统相关的系统调用有以下这些：

open read write close mkdir chdir mknod pipe dup link unlink fstat

这些系统调用的实现都在 sysfile.c 文件中，实际操作的代码主要在 file.c 和 fs.c 中，请大家将整个文件系统的操作都看完，再着重准备要讲的部分。

第一部分 open 和 close：文件的打开（创建）和关闭

入口代码：sysfile.c 中的 sys_open 和 sys_close 函数

关注问题：

- 1、 open 和 close 系统调用的作用是什么，系统调用的参数有哪些，各自代表什么含义？
- 2、 open 和 close 中各种失败情况的原因都是什么
- 3、 sys_open 函数中 f 变量是什么类型的变量，里面的各个成员变量都代表什么，在 sys_open 函数中被设成了什么？和 fd 变量含义上有什么区别？
- 4、 sys_close 函数中 proc->ofile[fd] = 0 这一句代表了什么？ofile 变量还有什么时候被赋值过？
- 5、 filealloc 函数的作用是什么，ftable 是一个什么数据结构，里面总共多少项？
- 6、 fdalloc 函数的作用是什么？
- 7、 （进阶题）请比较 linux 内核代码中的 open 和 close 函数，它们的实现和 xv6 中有哪些不同，为什么？（包括 open 和 close 进一步调用的函数，以及类似 ftable 的数据结构）

第二部分 read 和 write (part 1): read 的实现和 buffer cache 管理

入口代码: sysfile.c 中的 sys_read 和 sys_write 函数, file.c 中的 fileread 和 filewrite 函数, fs.c 中的 readi 和 writei 函数

关注问题:

- 1、 fileread 和 filewrite 的三个参数分别是什么?
- 2、 fileread 和 filewrite 函数中对 f->type 的判断代表什么? (f->type == FD_PIPE 的情况由负责第五部分的同学来讲) 文件的权限管理是怎样实现的?
- 3、 f->off 变量代表了什么? 它的值是怎么变化的?
- 4、 readi 和 writei 函数的参数 ip, 它的变量类型是什么, 包含的成员变量都是什么含义?
- 5、 请逐行介绍 readi 函数中文件操作实际执行到的内容。(writei 与 readi 功能相似, 请也简要对比介绍一下, 详细介绍由第三部分同学完成)
- 6、 bread 函数最终调用 bget 函数, 请介绍 bget 函数的功能。特别是 bget 函数中设计 buffer cache 的替换, 请点明替换算法是什么。
- 7、 buffer cache 中的 dirty 块是由哪个函数写入磁盘的? (跟踪 B_DIRTY, 具体内容又第三部分介绍)
- 8、 (进阶题) linux 内核代码中 buffer cache 的替换是如何实现的? 替换算法和 xv6 有何不同?

第三部分 read 和 write (part 2): write 的实现和日志管理

入口代码：fs.c 中的 writei 函数

关注问题：

- 1、 writei 函数中调用了 log_write 函数，它的功能是什么？
- 2、 阅读 end_op 函数代码，介绍 end_op 函数何时会被调用，end_op 函数的功能是什么。
- 3、 end_op 函数调用了 commit 函数，commit 函数的流程是什么，每个语句分别完成了什么功能？（介绍相关调用函数）
- 4、 阅读 initlog 函数，log 的初始化是何时完成的，里面调用了 recover_from_log 函数，为什么要调用它，它的作用是什么？
- 5、 （进阶题）linux 代码中实现了哪些日志文件系统，和 xv6 中的日志管理有什么不同？

第四部分 mkdir, chdir 和 mknod：文件目录的操作和设备文件的创建

入口代码：sysfile.c 中的 sys_mkdir 函数，sys_chdir 函数以及 sys_mknod 函数

关注问题：

- 1、 请逐行介绍 sys_mkdir, sys_chdir 和 sys_mknod 函数的功能。
- 2、 create 函数调用了 dirlookup 函数，它实现了什么功能，是怎么实现的？
- 3、 create 函数还调用了 ialloc 函数，它的功能是什么？
- 4、 namei 函数和 create 函数最终都调用 fs.c 文件中的 namex 函数，请逐行介绍它的功能。
- 5、 （进阶题）linux 内核代码中，和 dirlookup 函数功能类似的函数是什么，实现上有何不同？

- 6、（进阶题）linux 内核代码中，和文件目录相关的系统调用还有什么，它们的实现是怎么样？

第五部分 pipe：管道的实现

入口代码：sysfile.c 文件中的 sys_pipe 函数； pipe.c 中的 pipealloc、pipeclose、pipewrite、piperead 函数

关注问题：

- 1、 pipe 类型的文件操作实现的是什么样的功能？ pipealloc 函数的两个参数分别代表什么？
- 2、 piperead 和 pipewrite 函数的流程是什么样的？请介绍出报错(return -1) 的原因，以及核心语句。
- 3、 pipe 类型的文件读写和 inode 类型的读写有什么区别？ pipe 类型操作的实现目的是什么，有什么优点？
- 4、（进阶题）linux 内核代码中 pipe 的实现和 xv6 有哪些不同？

第六部分 剩下的一些文件操作（dup，link，unlink 和 fstat）

！特别注意：这一部分 xv6 的代码已经比较少了，请同学尽量完成一下进阶题的内容。特别是上课有介绍过，但是 xv6 代码中没有涉及的部分

入口代码：sysfile.c 文件中的 sys_dup, sys_link, sys_unlink 以及 sys_fstat 函数

关注问题：

- 1、 dup 系统调用的功能是什么，为什么要设计这样一个系统调用？
- 2、 link 和 unlink 两个系统调用实现了什么功能？请介绍它们的实现流程。

- 3、 fstat 系统调用返回了文件的哪些状态？
- 4、 （进阶题）linux 内核代码中有哪些和文件系统相关的系统调用，除了第三部分同学介绍的那些之外，比 xv6 中还多出了哪些，请一一介绍它们的功能。（特别注意 fsync 的实现）