

Week06 Assignment

班级: 192112
学号: 19373073
姓名: 何潇龙

1. 文件和目录的作用有什么不同?

文件: 计算机中的信息都要保存在文件中, 在Linux系统中, 将所有的外部设备都抽象为文件, 因此文件操作是Linux编程的最基本的操作。

目录: 从用户角度看, 文件都是属于某个目录下面的, 从逻辑上看, 文件系统以目录树的层次结构组织文件。目录本身在 Linux 中也是一种文件, 不过其内容是其所包含的文件或者目录信息的目录项而不是具体的文件中的数据。

2. 简述标准文件 I/O 和系统调用 I/O 的各自优缺点是什么?

文件I/O又称为低级磁盘I/O, 遵循POSIX相关标准。文件I/O中用文件描述符表现一个打开的文件, 可以访问不同类型的文件如普通文件、设备文件和管道文件等。通过文件I/O读写文件时, 每次操作都会执行相关系统调用。这样处理的好处是直接读写实际文件, 坏处是频繁的系统调用会增加系统开销,

标准I/O可以看成是在文件I/O的基础上封装了缓冲机制。先读写缓冲区, 必要时再访问实际文件, 从而减少了系统调用的次数。而标准I/O中用FILE (流) 表示一个打开的文件, 通常只用来访问普通文件。

3. 符号链接和硬链接的异同点是什么?

同: 链接可以如同原始文件一样来对待。链接可以与普通的文件一样被执行、编辑和访问。对系统中的其他应用程序而言, 链接就是它所对应的原始文件。

异: 硬链接中, 一个文件发生了修改之后, 另外一个文件也能够立即看到这种变化。符号链接中源文件发生变化, 链接文件的属性不会发生变化, 只是访问时会发生错误。

4. 查看权限

假设目前系统的 umask 为 0022, 执行 `CREATE("a.txt", 765)` 代码后, 请写出文件 `a.txt` 所有者、组用户及其他用户对该文件的权限。

umask写成二进制为 000010010

其反码为 111101101

765写成二进制为 111110101

umask和765作&运算的结果为111100101

则权限为-rwxr--r-x

文件所有者的权限: 可读可写可执行

组用户的权限: 只读

其他用户的权限: 可读可执行

5. 文件锁的使用

关于文件锁的使用，当某进程对某个文件上了读取锁或者写入锁后，请就下面几种情况分别编写程序进行分析，并解释原因。

(1) 创建子进程，请问子进程对该文件是否拥有同样的锁？

fork函数调用会创建子进程，子进程的地址空间是在调用fork时父进程地址空间的拷贝。因为子进程地址空间跟父进程一样，所以调用fork时，子进程继承了父进程中的锁。

(2) 该进程执行 exec 调用，请问该进程对文件的锁是否依然有效？

执行 `exec` 后，新程序继承原程序的锁。此时该进程对文件的锁不再有效。

(3) 当该进程对文件建立锁的文件描述符关闭后，这些锁是否仍然有效？如果对这些文件描述符执行 dup 操作，锁是否对新的文件描述符仍然有效？

锁的文件描述符关闭后锁就无效了。dup会复制文件描述符，仍然有效。

(4) 假设该进程对某个文件加了读取锁/写入锁，然后又同时对该进程进行写入/读取操作，将会发生什么情况？

--	阻塞读	堵塞写	非阻塞读	非阻塞写
读写锁	正常读取数据	阻塞	正常读写数据	返回 EAGAIN 错误
写入锁	阻塞	阻塞	返回 EAGAIN 错误	返回 EAGAIN 错误