

Lab5挑战性任务

杨新宇 王宇翔 2021.6.7

任务1：解决writef输出被打断的问题

难度：☆

在系统运行过程中会出现writef输出被打断的情况，请分析原因并解决这个问题。

任务2：IDE磁盘读取限制

难度：★

在实验中，用户态使用了系统调用来实现了读写特定的内存地址，文件系统进程也使用了固定的进程号来标识。这种设计有着一定的安全风险的同时有违微内核的设计思路。在微内核设计领域，针对这一问题已经提出了内存控制、权限分派、IPC转发等等的解决方案。

请你调研相关资料，修改系统调用或编写新的系统调用，使得只有文件系统进程能读写IDE磁盘，且文件系统进程仅能读写IDE磁盘而不能读写其他的内核地址。

任务3：虚拟设备机制

难度：★★☆ ~ ★★★☆

虚拟设备机制主要用于管理进程对外设的共享过程，通过虚拟设备机制实现设备分配可以提高外设的利用率。

- 部分设备（如打印机，在我们的系统中可以使用console进行模拟）具有独享的特性，如果由多个进程交替使用不加以控制会出现错误，此时需要**对设备的分配进行控制（例如实现锁机制）**使得多进程可以安全地共享使用设备。
- 模拟**SPOOLing技术**，使用SPOOLing程序来管理对设备的I/O操作，将独享型的设备转换为共享型的设备。
- 实现虚拟设备后可以尝试能否模拟出死锁（可以自定义新的设备），并且使用理论课的知识避免死锁的产生。

以上加粗部分为必做内容。本题开放性较强，除以上实现角度之外，还可以选择其他角度进行实现。实现时需要简要说明希望解决的问题并提供详细的测试结果。

任务4：完善文件系统功能

难度：★★★★ ~ ★★★★★

我们希望能进一步完善文件系统的用户接口部分。

- 对当前工作目录进行管理，实现getcwd, chdir函数。
- 对文件操作的附加参数进行考虑，例如实现 `open(const char *path, int mode)` 函数中的附加参数。
 - 指定O_CREAT参数时，当想要打开的文件不存在时需要进行创建

- 指定**O_APPEND**参数时，每次写时都追加到文件的尾端
 - 指定**O_DIRECTORY**参数时，当path引用的不是目录则出错
- 实现**openat**函数，其形式为：`openat(int fd, const char *path, int mode)` 函数，该函数在指定路径时有三种情况。
 - path参数指定绝对路径，此时fd参数被忽略。
 - path参数指定相对路径，fd参数指定相对路径名在文件系统中的开始地址。
 - path参数指定相对路径且fd参数具有特殊值AT_FDCWD，此时路径名在当前工作目录中获取。
- 文件的链接机制：实现**link**、**unlink**、**symlink**函数，用于创建和解除链接。
- 实现更多的文件操作，例如用于创建目录的**mkdir**、**mkdirat**函数、用于删除目录的**rmdir**函数、用于重命名文件的**rename**函数。

该任务的灵活性较强，除以上的必做内容之外还可以基于MOS的文件系统实现更加丰富文件操作。对于实现的文件操作需要提供测试样例和运行结果。

对于系统中设计不合理的地方可以进行优化（函数功能、文件组织方面），以便于之后的功能实现。在原本的文件系统中普通文件和目录文件的区分并不明确，在实现功能前需要先改写部分函数将二者进行区分。

操作的具体含义可以参考《UNIX环境高级编程》中的第3、4章。MOS中定义的文件操作和UNIX标准操作有一定区别，在实现时应侧重于**完成函数的功能**，对于细节部分的定义（例如可选参数的取值）合理即可。

任务5：多文件系统

难度：★★★★ ~ ★★★★★

MOS中的部分文件操作实现使用了文件系统IPC机制，这样的机制可以被视为文件系统抽象层（VFS）。文件系统抽象层在Linux等大型操作系统中的应用十分广泛，它使得操作系统能够于多个不同类型的文件系统进行交互（如FAT32、EXT2等），提升了文件系统的可用性。我们希望你多文件系统进行简单的实现，体会文件系统抽象层的用途。具体要求如下：

- 使用**多个文件系统进程**对不同文件系统进行管理，用户可以**指定需要交互的进程**。在每个文件系统进程中都需要进行初始化的操作。
 - 在实现时建议在用户接口函数中对路径进行调整来指定需要交互的文件系统进程。例如 `open("/root1/...", ...)` 访问的是1号文件系统的内容，即该用户需要和1号文件系统进程进行交互。同理 `open("/root2/...", ...)` 访问的是2号文件系统的内容，需要和2号文件系统进程进行交互。
- 对不同文件系统进程进行**设备权限控制**，例如仅允许一个文件系统进程对控制台进行访问，其余文件系统无法访问。
- 考虑文件系统进程出现停止服务的情况，此时需要对内存中的缓存数据进行处理。
- 对不同的文件镜像指定不同的文件格式（对fsformat.c进行修改）。可以尝试一些通用的文件系统格式，例如FAT32（难度较大）。

注：使用 `gxemul -E testmips -C R3000 -M 64 -d 0:gxemul/fs.img -d 1:gxemul/fs2.img elf-file` 可以挂载多个文件系统镜像，其中0, 1指定了不同的虚拟磁盘的磁盘ID。

除以上的必做内容（加粗部分）之外，还可以对多文件系统进行更细致的实现。

在Lab5的挑战性任务中，**前两个任务为必做**，并在后三个任务选择一个至两个进行完成。在书写实验报告和准备申优答辩时，请加入以下内容：

- 对于任务的实现思路，并配合**关键代码**进行说明。
- 对于功能的**测试程序**，以及运行测试程序得到的**运行结果**。
- 完成挑战性任务过程中**遇到的问题及解决方案**。