

OS2018 Lab2 实验报告

姓名：陈劭源

学号：161240004

June 19, 2018

1 测试的运行方法

直接运行 `make run`。测试的中间信息和结果会打印在终端中。

2 在 C 中实现多态

本实验中，需要实现若干个不同的文件系统。这些文件系统以一套统一的接口为用户提供服务，但需要以不同的方式实现。这种情况很适合用派生类建模，但在此之前需要先在 C 语言中实现多态。

这里实现了多态支持头文件 `<pmr.h>`，由多个宏组成，功能分别如下。

1. Member(...)

说明：加在函数声明的参数列表前，表示该函数是成员函数。

2. MemberOf(cname)

说明：加在函数体的最前面，表示该函数是 `cname` 的成员函数。用于引入 `base` 和 `this`。

3. this

说明：用于成员函数内，表示当前对象的左值（不是指针）。

4. base

说明：用于成员函数内，表示当前对象基类的左值（不是指针）。

5. Interface(pname)

...
End_Interface

说明：用于声明虚函数表结构体。虚函数表中必须包含构造函数和析构函数（可以为空指针）。该宏会自动产生一个析构函数。

6. Inherits(pname)

说明：用于指示继承关系。

7. Implementation(pname, cname)

说明：用于声明子类的虚函数表。

8. Invoke(p_object, method, ...)

说明：用于调用虚函数。

9. New(cname, ...)

说明：分配对象 `cname` 的空间，并调用构造函数。返回指向该对象的指针。

10. Delete(p_object)

说明：调用析构函数销毁对象，并回收内存。

2.1 API 设计

2.1.1 vfs 模块

```
MODULE {
    void (*init)();
    int (*access)(const char *path, int mode);
    int (*mount)(const char *path, filesystem_t *fs);
    int (*unmount)(const char *path);
    int (*fsls)(int (*fn)(const char *path, filesystem_t *fs));
    int (*open)(const char *path, int flags);
    ssize_t (*read)(int fd, void *buf, size_t nbyte);
    ssize_t (*write)(int fd, const void *buf, size_t nbyte);
    off_t (*lseek)(int fd, off_t offset, int whence);
    int (*close)(int fd);
} MOD_NAME(vfs);
```

其中，fsls 函数用于遍历文件系统。

2.2 filesystem 对象的方法

```
Interface(filesystem)
    int      (*_ctor)  Member (const char *name);
    int      (*_walk)  Member
        (int (*fn)(const char *path, inode_t inode, int mode, int length));
    inode_t  (*lookup) Member (const char *path);
    inode_t  (*create) Member (const char *path);
    int      (*access) Member (inode_t inode, int mode);
    file_t*  (*open)   Member (inode_t inode, int flags);
End_Interface
```

其中，walk 函数用于遍历当前文件系统中的文件。

2.3 file 对象的方法

```
Interface(file)
    int      (*_ctor)  Member (filesystem_t *fs, inode_t inode, int flags);
    ssize_t  (*read)   Member (char *buf, size_t size);
    ssize_t  (*write)  Member (const char *buf, size_t size);
    off_t    (*lseek)  Member (off_t offset, int whence);
End_Interface
```