对源码文件系统的修改

1. **修改内容**
   1. **增加系统调用**

对文件系统新增五个系统调用,分别有lseek, dup, stat, pread, pwrite。为了达到效果,新增了十个文件并修改了部分函数,此处新增的代码量为600行。

* 1. **更改文件系统的目录结构**

源码采用的是扁平化目录结构，欲将其改为多级目录结构。

1. **具体实现**
   1. **增加系统调用**

**lseek:**

作用:移动文件内部指针,以辅助读写操作；

系统调用函数位置:lib/lseek.c，

实现函数位置:fs/lseek.c，

新增消息的名称为OFFSET，

参数1,文件修饰符(int)，

参数2,偏移数量(int)，

参数3,偏移参考点(SEEK\_SET,SEEK\_CUR,SEEK\_END)，

返回当前文件偏移后的文件指针的位置,以SEEK\_SET为偏移参考点。

**dup:**

作用:复制文件修饰符,重定向文件；

系统调用函数位置:lib/dup.c，

实现函数位置:fs/dup.c，

新增消息的名称为DUP，

参数1,文件修饰符(int)，

返回复制后的文件修饰符。

**stat:**

打印文件除内容外的所有信息；

系统调用函数位置:lib/stat.c，

实现函数位置:fs/stat.c，

新增消息的名称为STAT，

参数1,文件修饰符(int)，

参数2,文件节点(i\_node)，

返回文件的路径,并打印文件的信息。

**pread:**

在指定位置进行文件的读操作；

系统调用函数位置:lib/pread.c，

实现函数位置:fs/pread.c，

新增消息的名称为PREAD，

参数1,文件修饰符(int)，

参数2,字符串指针(void\*)，

参数3,读取的字节数(int)，

参数4,读取的位置(int)，

返回读取的字节数。

**pwrite:**

在指定位置进行文件的写操作；

系统调用函数位置:lib/pwrite.c，

实现函数位置:fs/pwrite.c，

新增消息的名称为PWRITE，

参数1,文件修饰符(int)，

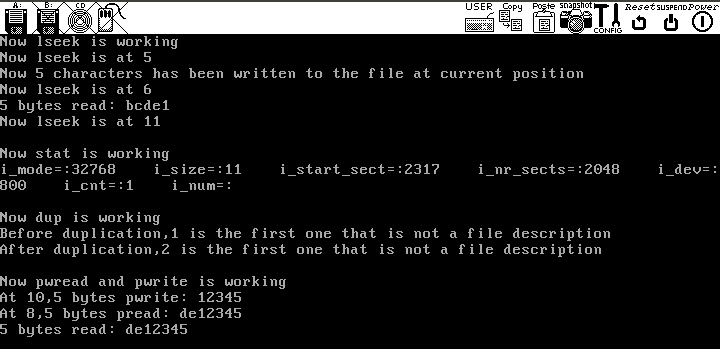
参数2,字符串指针(void\*)，

参数3,写入的字节数(int)，

参数4,写入的位置(int)，

返回读取的字节数。

五个系统调用的测试结果如下图,按顺序分别为lseek, stat, dup, pwread和pwrite：



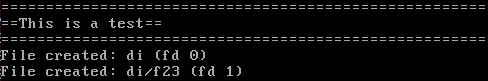
* 1. **更改文件系统目录结构**

原文件目录结构是扁平的，只有根目录一个目录，现欲将其改为多级目录结构。

将目录以目录文件的形式存放。当inode结点的i\_mode设为I\_DIRECTORY时表示其为目录文件，目录文件中存放该目录下的所有文件的dir\_entry，当需要读取某个目录下的文件时，需要先找到该目录文件，从目录文件中读取对应名字的dir\_entry，得到目标文件的inode\_nr，再通过该inode\_nr找到对应的inode，即可找到文件在磁盘上的位置。当有多级目录时以此类推。

创建目录时使用添加的open\_dir函数，将消息类型设置为OPENDIR，从而让文件系统去创建一个目录文件。

创建一个名为di的目录文件以及在di目录下创建一个名为f23的普通文件的测试如下图：

。