

Lab7 Shell实现

实现命令汇总

1. ls dir, 列出指定目录下的文件和文件夹
2. mkdir dir, 创建文件夹dir
3. touch file, 创建文件file
4. rm file/dir, 删除文件或者文件夹
5. run prog, 执行指定程序prog
6. help, 打印支持的命令列表
7. exit, 退出nachos

文件系统相关命令

实现都较为简单, 对之前实现的文件系统的功能进行了一定的封装, 并且修复了之前文件系统中的一个隐藏bug, 包括

1. 文件系统的打开文件中保存了打开文件的文件头指针, 但是在部分不规则使用的代码处对该指针进行 delete 操作, 导致打开文件失败
 2. 扩充目录文件长度时, 分配磁盘块集成在写入操作中, 此时会对文件头进行变更, 因此保存目录文件时应该首先保存文件数据, 在保存文件头。
 3. 修改了目录文件的打印方法, 使打印数据更加完善, 包含目录项类型, 创建日期等。
- 实现功能代码如下:

```
else if (strcmp(op, "ls") == 0) {
    char dirPath[20] = {};
    scanf("%s", dirPath);

    OpenFile* dirFile = fileSystem->Open(dirPath);
    Directory* dir = new Directory();
    dir->FetchFrom(dirFile);
    dir->List();
} else if (strcmp(op, "mkdir") == 0) {
    char dirPath[20] = {};
    scanf("%s", dirPath);

    fileSystem->Create(dirPath, 0, FALSE);
} else if (strcmp(op, "touch") == 0) {
    char filePath[20] = {};
    scanf("%s", filePath);

    fileSystem->Create(filePath, 0, TRUE);
} else if (strcmp(op, "rm") == 0) {
    char filePath[20] = {};
    scanf("%s", filePath);

    fileSystem->Remove(filePath);
}
```

执行命令

执行一个用户程序，用户程序需要包含在nachos文件系统中，并且打印程序的 exit code，执行完成之前shell需要等待用户程序，无法进行其他操作，代码流程拷贝了系统调用中的 join 操作。代码如下：

```
if (strcmp(op, "run") == 0) {
    char prog[20] = { };
    scanf("%s", prog);

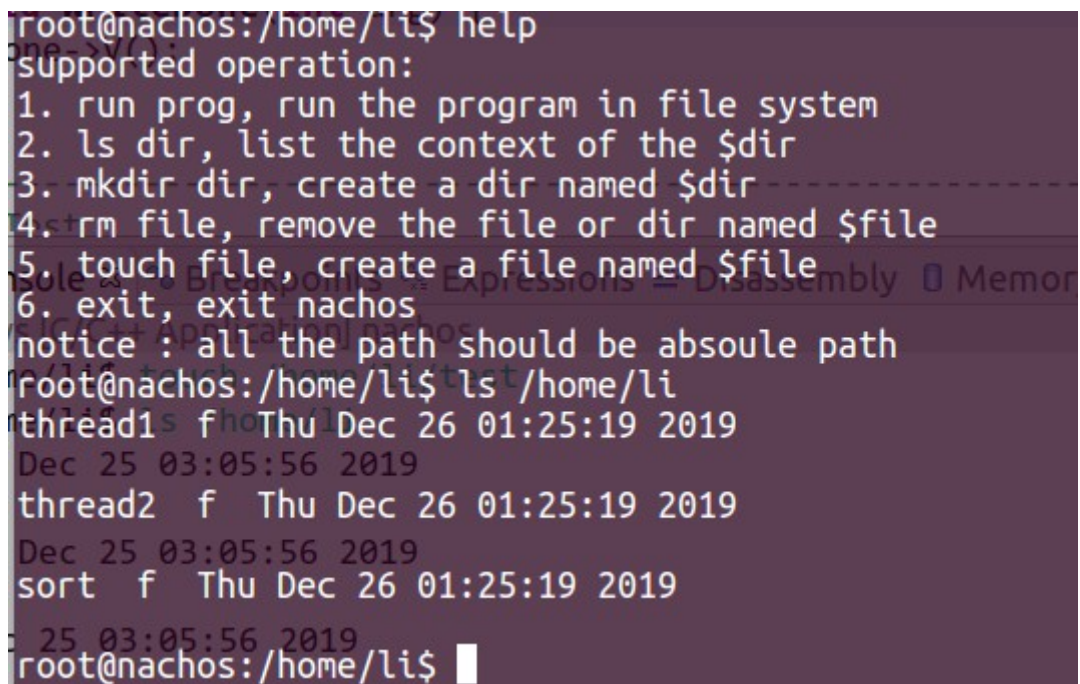
    Thread* thread = new Thread("run thread");
    thread->waitingList->Append((void*) currentThread);
    thread->Fork(StartProcess, prog);

    IntStatus oldLevel = interrupt->SetLevel(IntOff);
    currentThread->Sleep();
    interrupt->SetLevel(oldLevel);

    printf("exit code is %d\n", currentThread->exitCode);
}
```

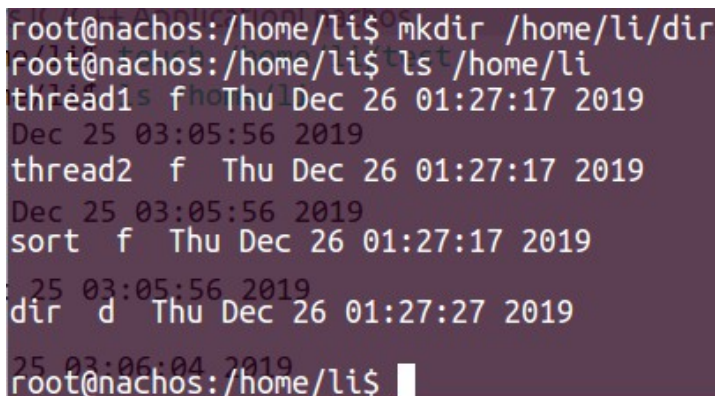
测试结果

1. help, ls:



```
root@nachos:/home/li$ help
supported operation:
1. run prog, run the program in file system
2. ls dir, list the context of the $dir
3. mkdir dir, create a dir named $dir
4. rm file, remove the file or dir named $file
5. touch file, create a file named $file
6. exit, exit nachos
notice : all the path should be absolute path
root@nachos:/home/li$ ls /home/li
thread1 f Thu Dec 26 01:25:19 2019
Dec 25 03:05:56 2019
thread2 f Thu Dec 26 01:25:19 2019
Dec 25 03:05:56 2019
sort f Thu Dec 26 01:25:19 2019
Dec 25 03:05:56 2019
root@nachos:/home/li$
```

2. mkdir, ls:



```
root@nachos:/home/li$ mkdir /home/li/dir
root@nachos:/home/li$ ls /home/li
thread1 f Thu Dec 26 01:27:17 2019
Dec 25 03:05:56 2019
thread2 f Thu Dec 26 01:27:17 2019
Dec 25 03:05:56 2019
sort f Thu Dec 26 01:27:17 2019
Dec 25 03:05:56 2019
dir d Thu Dec 26 01:27:27 2019
Dec 25 03:06:04 2019
root@nachos:/home/li$
```

3. touch, ls:

```

root@nuchos:/home/li$ touch /home/li/test
root@nuchos:/home/li$ ls /home/li
thread1 f Thu Dec 26 01:28:36 2019

thread2 f Thu Dec 26 01:28:36 2019
e/li$ touch /home/li/test
sort$ f Thu Dec 26 01:28:36 2019
Dec 25 03:05:56 2019
dir d Thu Dec 26 01:28:42 2019
Dec 25 03:05:56 2019
test f Thu Dec 26 01:28:48 2019
Dec 25 03:05:56 2019
root@nuchos:/home/li$

```

4. run:

```

root@nuchos:/home/li$ run /home/li/sort
exit code is 1024
root@nuchos:/home/li$

```

5. exit:

```

root@nuchos:/home/li$ exit
Machine halting!
e/li$ mkdir /home/li/dir
Ticks: total 1200218, idle 1185430, system 7670, user 7118
Disk I/O: reads 1159, writes 95
Console I/O: reads 10, writes 0
Paging: faults 0
Network I/O: packets received 0, sent 0
Dec 25 03:05:56 2019
Cleaning up...
lihaiyang@ubuntu:~/NachosLab/nuchos/nuchos_dianti/nuchos-3.4/code/filesys$

```