PA3

硬链接

第一步是找到访问inode的方式，或者得知硬链接数的方式

首先用dirent.h中的函数和结构体获得并存储文件的目录信息，然后使用<sys/stat.h>中的stat结构体获得硬链接信息

```
struct stat {

    dev_t   st_dev;     //设备
    ino_t   st_ino;     // inode
    mode_t  st_mode;    //文件权限信息
    nlink_t st_nlink;   //硬连接数
    uid_t   st_uid;     //用户ID
    gid_t   st_gid;     //组ID
    dev_t   st_rdev;    //设备类型（如果是节点设备的话）
    off_t   st_size;    //文件大小（字节）
    unsignedlong st_blksize;    //块大小
    unsignedlong st_blocks;     //块数目
    time_t  st_atime;   //文件最后访问时间
    time_t  st_mtime;   //文件最后修改时间
    time_t  st_ctime;   //文件最后变动时间
};
```

```
avril@ubuntu:~/OSLab$ ./scanner
/home/avril/OSLab
current directory: /home/avril/OSLab
ilink 1 inode 726292 name lab3-191220083.zip,type =8
ilink 1 inode 728865 name lab4.zip,type =8
ilink 3 inode 721657 name os2021,type =4
ilink 2 inode 661523 name pa1,type =4
ilink 1 inode 660704 name lab3.zip,type =8
ilink 2 inode 670327 name pa2,type =4
ilink 1 inode 729046 name scanner,type =8
ilink 11 inode 667281 name .,type =4
ilink 4 inode 728904 name lab4-191220083,type =4
ilink 2 inode 721590 name Report,type =4
ilink 33 inode 671231 name ..,type =4
ilink 1 inode 721451 name scanner.c,type =8
ilink 1 inode 728922 name lab4-191220083.zip,type =8
ilink 4 inode 833946 name lab1-191220083,type =4
ilink 4 inode 721551 name lab3-191220083,type =4
ilink 4 inode 721546 name lab2-191220083,type =4
ilink 1 inode 721711 name lab2-191220083.zip,type =8
ilink 8 inode 667319 name .git,type =4
```

尝试输出 做一些观察

基于以上尝试构建代码:

```
#include <stdio.h>
#include <dirent.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/stat.h>
#include <stack>
#include <map>
#include <iostream>
#include <vector>
using namespace std;
int main()
{
    string dir_name;
    cin>>dir_name;//被检查的目录可以由用户指定

    stack<string> stk;
    stk.push(dir_name);
    multimap<ino_t,string> dict;
    struct dirent *stdinfo;
    struct stat state;
    while(!stk.empty())
    {
        cout<<"current directory: "<<dir_name<<endl;
        DIR *dir=NULL;
        dir=opendir(dir_name.c_str());
        if(!dir)
        {
            printf("failed to open directory\n");
            return 0;
        }
        while(1)
        {
            if((stdinfo=readdir(dir))==0)
            {
                closedir(dir);
                break;
            }
            printf("ilink %lu ",state.st_nlink);
            printf("inode %lu ",state.st_ino);
            printf("name %s,type =%d\n",stdinfo->d_name,stdinfo->d_type);

            if(stdinfo->d_type == 4)//是目录 之后还要检查该目录项下的文件
                stk.push(stdinfo->d_name);
            else if(stdinfo->d_type == 8)
            {
                stat(stdinfo->d_name,&state);
                if(state.st_nlink>=2)
                {
                    printf("add %lu %s",state.st_ino,stdinfo->d_name);
                    dict.insert(pair<ino_t,string>(state.st_ino,stdinfo-
>d_name));
                }
            }

        }
```

```
        dir_name=stk.top();
        cout<<dir_name<<endl;
        stk.pop();
        while(dir_name!=".git" && !stk.empty())
        {
            dir_name=stk.top();
            stk.pop();
        }

    }
    multimap<ino_t,string>::iterator it = dict.begin();
    for(;it!=dict.end();++it)
    {
        cout<<it->first<<" "<<it->second<<endl;
    }

    return 0;
}
```

调试时一开始以为是字符串转换失败导致目录无法被打开，输出检查后发现是类型判断写反了误把文件名当成目录名压栈。修正后发现仅依靠判断type==4 会检查到.git文件夹，其中branch会作为无法打开的目录出现，于是对此做了特殊处理。

结果如下：（为了测试提前建立了一个硬链接）

```
^[[Aavril@ubuntu:~/OSL./scanner
/home/avril/OSLab
current directory: /home/avril/OSLab
ilink 0 inode 140617392799538 name lab3-191220083.zip,type =8
ilink 1 inode 726292 name lab4.zip,type =8
ilink 1 inode 728865 name os2021,type =4
ilink 1 inode 728865 name scanner.cpp,type =8
ilink 1 inode 721451 name pa1,type =4
ilink 1 inode 721451 name lab3.zip,type =8
ilink 1 inode 660704 name pa2,type =4
ilink 1 inode 660704 name scanner,type =8
ilink 1 inode 726555 name .,type =4
ilink 1 inode 726555 name lab4-191220083,type =4
ilink 1 inode 726555 name 191220083.pdf,type =8
add 839012 191220083.pdfilink 2 inode 839012 name Report,type =4
ilink 2 inode 839012 name ..,type =4
ilink 2 inode 839012 name os_report.pdf,type =8
add 839012 os_report.pdfilink 2 inode 839012 name lab4-191220083.z
ip,type =8
ilink 1 inode 728922 name lab1-191220083,type =4
ilink 1 inode 728922 name lab3-191220083,type =4
ilink 1 inode 728922 name lab2-191220083,type =4
ilink 1 inode 728922 name lab2-191220083.zip,type =8
```

可见最后两行输出了该文件夹下唯一一个具有两个硬链接的文件，名字分别为191220083.pdf和os_report.pdf 共享的inode编号是839012

> 参考文献:
>
> https://blog.nowcoder.net/n/72f9fee9f1e34f4099c2421b9f7610d9
>
> https://www.freecplus.net/278e93836aa6471492a206cde2f768fe.html