

opendir 、readdir 小结

1. opendir()

DIR* opendir (const char * path); （获取 path 子目录下的所有文件和目录的列表，如果 path 是个文件则返回值为 NULL）

DIR 结构体的原型为：struct_dirstream

在 linux 系统中：

```
typedef struct __dirstream DIR;

struct __dirstream
{
    void *__fd; /* `struct hurd_fd' pointer for descriptor. */
    char *__data; /* Directory block. */
    int __entry_data; /* Entry number `__data' corresponds to. */
    char *__ptr; /* Current pointer into the block. */
    int __entry_ptr; /* Entry number `__ptr' corresponds to. */
    size_t __allocation; /* Space allocated for the block. */
    size_t __size; /* Total valid data in the block. */
    __libc_lock_define (, __lock) /* Mutex lock for this structure. */
};
```

2. readdir

struct dirent* readdir(DIR* dir_handle); （个人理解循环读取 dir_handle, 目录和文件都读）

读取 opendir 返回值的那个列表

返回 dirent 结构体指针，dirent 结构体成员如下，（文件和目录都行）

```
struct dirent
{

    long d_ino; /* inode number 索引节点号 */

    off_t d_off; /* offset to this dirent 在目录文件中的偏移 */

    unsigned short d_reclen; /* length of this d_name 文件名长 */
```

```
    unsigned char d_type; /* the type of d_name 文件类型 */

    char d_name [NAME_MAX+1]; /* file name (null-terminated) 文件名,
最长 255 字符 */

}
```

代码

```
#include<stdio.h>
#include<dirent.h>

int main(void)
{
    DIR *dirptr=NULL;
    int i=1;
    struct dirent *entry;
    if((dirptr = opendir("test.dir"))==NULL)
    {
        printf("opendir failed!");
        return 1;
    }
    else
    {
        while(entry=readdir(dirptr))
        {
            printf("filename%d=%s\n",i,entry->d_name);
            i++;
        }
        closedir(dirptr);
    }
    return 0;
}
```

输出:

filename1=dir1

filename2=file3

filename3=..

filename4=dir3

filename5=.

filename6=file1

filename7=file2

filename8=dir2