



Homework Assignment #4: Searching a file



Outline

- **opendir()**
- **readdir()**
- **closedir()**
- **Homework Assignment #4**



Outline

- **opendir()**
- readdir()
- closedir()
- Homework Assignment #4

opendir()

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

```
DIR *opendir(const char *name);
```

```
EX: DIR* pDir = opendir( “.” );
```

■ DESCRIPTION

- Opens a directory stream corresponding to the directory *name*, and returns a pointer to the directory stream. The stream is positioned at the first entry in the directory.

■ RETURN VALUE

- Return a pointer to the directory stream. On error, NULL is returned



Outline

- `opendir()`
- `readdir()`
- `closedir()`
- Homework Assignment #4

readdir()

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

```
struct dirent *readdir(DIR *dirp);
```

EX:

```
struct dirent* pEntry = NULL;  
while ((pEntry = readdir(pDir)) != NULL) { ... }
```

■ DESCRIPTION

- Returns a pointer to a *dirent* structure representing the next directory entry in the directory stream pointed to by *dirp*.

■ RETURN VALUE

- On success, **readdir()** returns a pointer to a *dirent* structure.
- If the end of the directory stream is reached or error, NULL is returned

dirent structure

```
struct dirent {  
    ino_t d_ino;                /* Inode number */  
    off_t d_off;                 /* current position in the directory  
                                stream.*/  
    unsigned short d_reclen;      /* Length of this record */  
    unsigned char d_type;       /* Type of file; not supported by  
                                all filesystem types */  
    char d_name[256];            /* Null-terminated filename */  
};
```

dirent structure (cont.)

- The following macro *constants* for the value returned in *d_type*:
 - **DT_BLK** This is a block device.
 - **DT_CHR** This is a character device.
 - **DT_DIR** This is a directory.
 - **DT_FIFO** This is a named pipe (FIFO).
 - **DT_LNK** This is a symbolic link.
 - **DT_REG** This is a regular file.
 - **DT SOCK** This is a UNIX domain socket.
 - **DT_UNKNOWN** The file type could not be determined.
- Only Btrfs, ext2, ext3, and ext4 have full support for returning the file type in *d_type*.



Outline

- `opendir()`
- `readdir()`
- **`closedir()`**
- Homework Assignment #4

closedir()

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

```
int closedir(DIR *dirp);
```

```
EX: closedir(pDir);
```

■ DESCRIPTION

- Closes the directory stream associated with *dirp*. A successful call to **closedir()** also closes the underlying file descriptor associated with *dirp*.

■ RETURN VALUE

- Returns 0 on success. On error, -1 is returned



Outline

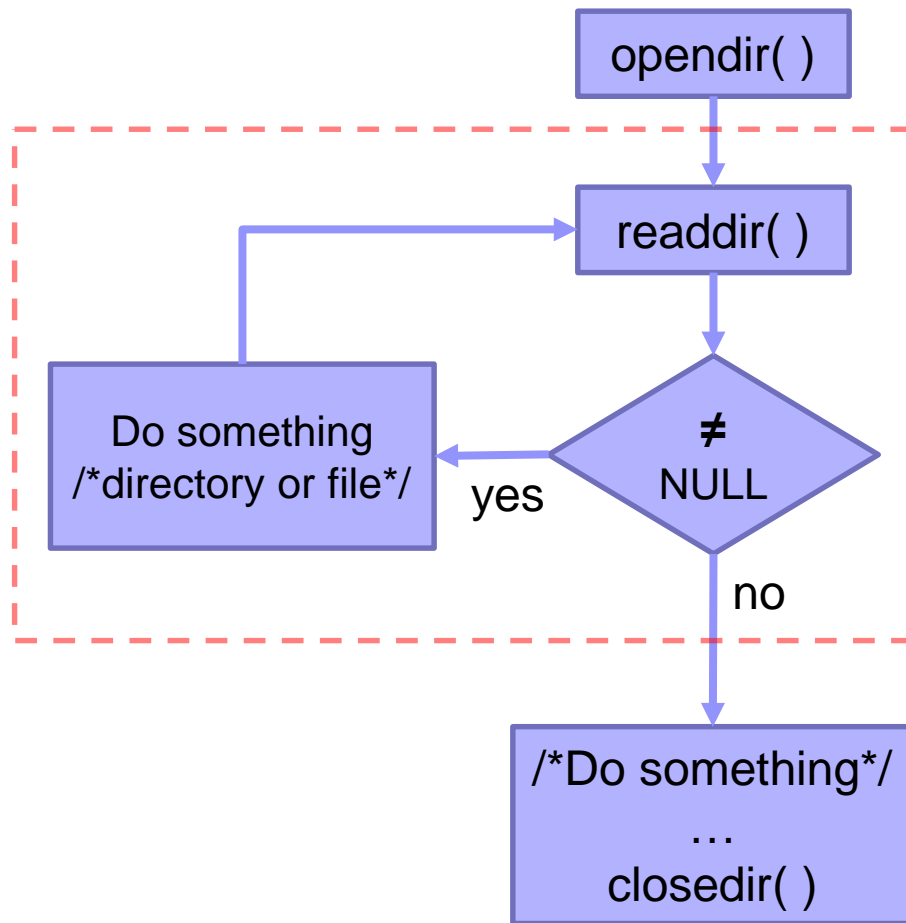
- `opendir()`
- `readdir()`
- `closedir()`
- **Homework Assignment #4**

Homework Assignment #4

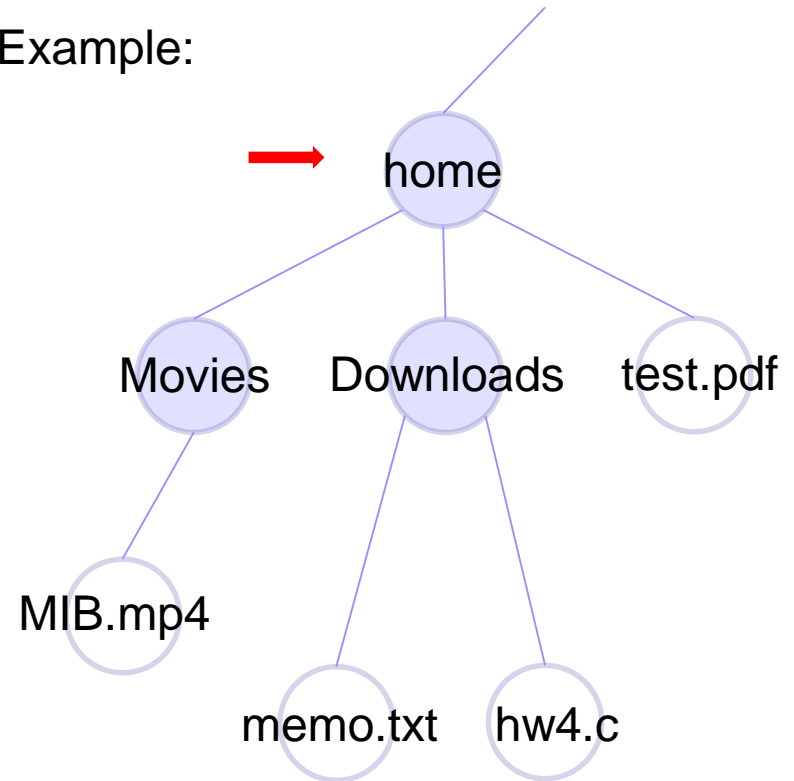
■ Searching a file

- Write a program that takes a *file name* and a *directory name*.
- Start at a given directory and descends the file tree from that point to search the file with the given file name.
- If the user does not give the starting directory parameter, assume that the starting directory is the current directory.
- If found, please print the corresponding pathname of the file.
- Otherwise, please show a “cannot find the corresponding file” message.

Flowchart



Example:



Note : `readdir()` will return `'.'` and `'..'`, so skip them to avoid the loop.

Homework Assignment #4

- Ex1: Searching '*passwd*' at root directory

```
test@test: ~/Documents
test@test:~/Documents$ ./hw4 / passwd
found in /usr/share/lintian/overrides/passwd
found in /usr/share/bash-completion/completions/passwd
found in /usr/bin/passwd
found in /etc/passwd
found in /etc/pam.d/passwd
found in /etc/cron.daily/passwd
test@test:~/Documents$
```

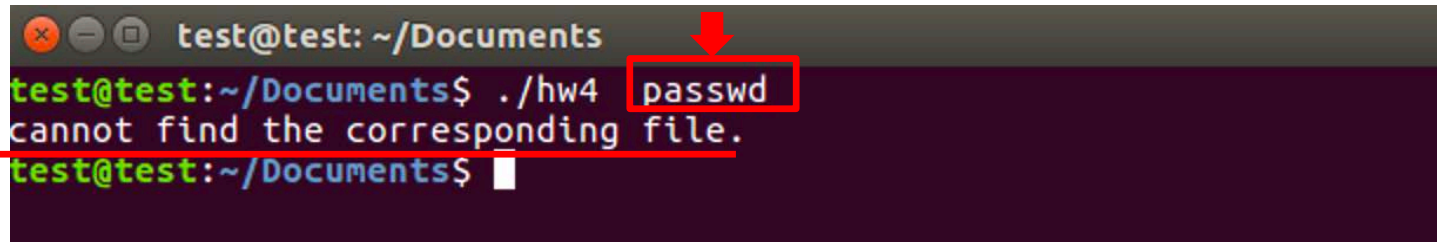
Your Program

```
test@test: ~
test@test:~$ sudo find / -iname 'passwd' -type f
[sudo] password for test:
/usr/share/lintian/overrides/passwd
/usr/share/bash-completion/completions/passwd
/usr/bin/passwd
find: '/run/user/1000/gvfs': Permission denied
/etc/passwd
/etc/pam.d/passwd
/etc/cron.daily/passwd
test@test:~$
```

For Checking

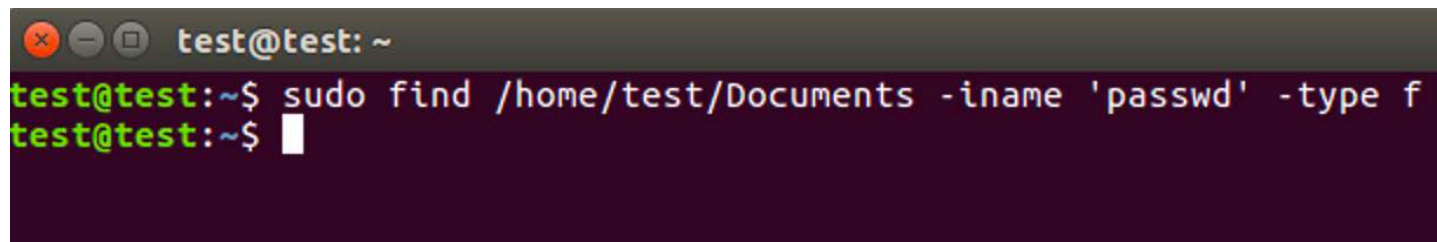
Homework Assignment #4

- Ex2: Searching '*passwd*' at default directory (current directory)



A terminal window with the title bar 'test@test: ~/Documents'. The prompt is 'test@test:~/Documents\$'. The user has entered './hw4 passwd', where 'passwd' is highlighted with a red box and a red arrow points to it from above. The output is 'cannot find the corresponding file.' followed by a new prompt 'test@test:~/Documents\$'.

```
test@test:~/Documents$ ./hw4 passwd
cannot find the corresponding file.
test@test:~/Documents$
```



A terminal window with the title bar 'test@test: ~'. The prompt is 'test@test:~\$'. The user has entered 'sudo find /home/test/Documents -iname 'passwd' -type f'. The output is empty, followed by a new prompt 'test@test:~\$'.

```
test@test:~$ sudo find /home/test/Documents -iname 'passwd' -type f
test@test:~$
```



Note

- Please use Linux platform.
- Make sure that all necessary header file be included.
- Leave an appropriate comment.

Turn in

- Deadline

2020/05/27 23:59:59

- Upload to i-learning

- File name

- Source code (e.g. HW4_7105056035.c)
- Word file (e.g. HW4_7105056035.docx)

- If you don't hand in your homework on time, your score will be deducted 10 points every day.

TA

- Name: 蘇唯銘
- Email: fff2456student@smail.nchu.edu.tw
- Title format: File System HW# - [your name]
- Lab: OSNET(1001A)

Reference

- Websites:

- ☐ <http://man7.org/linux/man-pages/man3/opendir.3.html>
- ☐ <http://man7.org/linux/man-pages/man3/readdir.3.html>
- ☐ <http://man7.org/linux/man-pages/man3/closedir.3.html>

- Book:

- ☐ The Linux Programming Interface - A Linux and UNIX System Programming Handbook

THE **LINUX** PROGRAMMING INTERFACE

A Linux and UNIX[®] System Programming Handbook

MICHAEL KERRISK



ANY QUESTION?

