CSCI 4061: Recitation 5

Today: Directories

- Directories
- mkdir(), rmdir(),
- opendir(), readdir(), closedir(), chdir()
- getcwd().

- Files
- stat()

Directories: mkdir 1/2

```
#include <sys/stat.h>
#include <sys/types.h>

int mkdir(const char *pathname, mode_t
mode);
```

- Attempts to create a directory.
- To create a directory
  - Specify directory name.
  - Specify permission. E.g. 0740;

Directories: mkdir 2/2

```
mode_t perms = 0740;  // Who can access this?
if (mkdir (“test_dir”, perms)) == -1) {
    perror(“failed to create directory”);
    return -1;
}
```

Try mkdir.c

Directories: rmdir

```
#include <unistd.h>

int rmdir(const char *pathname);
```

- Deletes a directory which must be empty.
- You must have permission to delete a directory.

Directories: opendir 1/2

```
#include <sys/types.h>
#include <dirent.h>

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

- Opens a directory stream corresponding to the directory name.

Directories: opendir 2/2

```
DIR * dir = opendir (“.”); //opens current directory

if(dir == NULL)
//opendir returns NULL if couldn’t open directory
{
    perror (“Failed to open directory”);
}
```

Try is_dir.c

Directories: readdir 1/3

```
#include <dirent.h>

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

- Returns a pointer to directory structure representing the net directory entry.
- Used for reading the contents of a directory
- **Returns NULL on reaching the end of the directory steam of if an error occurred.**

Directories: readdir 2/3

```
#include <dirent.h>

struct dirent * readdir(DIR * dirp);

  While((de = readdir(dr))!= NULL)
        printf(“%S\n”, de->d_name);
```

Directories: readdir 3/3

```
struct dirent {
    ino_t          d_ino;       /* inode number */
    off_t          d_off;       /* offset to the next dirent */
    unsigned short d_reclen;    /* length of this record */
    unsigned char  d_type;      /* type of file; not supported
                                   by all file system types */
    char           d_name[256]; /* filename */
};
```

- Try the test file myls.c

Directories: closedir

```
#include <sys/types.h>
#include <dirent.h>

int closedir(DIR * dirp);
```

- Closes the directory stream associated with dirp.

Directories: getcwd 1/2

```
#include <unistd.h>

char * getcwd(char * buf, size_t size);
```

- Copies an absolute pathname of the current working directory to the array pointed to by buf.
- Try getcwd.c

Directories: chdir

```
#include <unistd.h>

int chdir(const char * path);
```

- Changes the current working directory of the calling process.

Files

- Abstraction provided by OS
- Collection of related information
- Directories are files as well
- Have attributes like type, size, permission, etc.

Files: stat 1/2

```
#include <sys/stat.h>

int stat(const char *path, struct stat
*buf);
```

Gets information about a file; `stat()` stats the file pointed to by `path` and fills in `buf` with the information about the file.

Run perm.c & file_exists.c

Files: stat 2/2

```
struct stat {
   dev_t     st_dev;     /* ID of device containing file */
   ino_t     st_ino;     /* inode number */
   mode_t    st_mode;    /* protection */
   nlink_t   st_nlink;   /* number of hard links */
   uid_t     st_uid;     /* user ID of owner */
   gid_t     st_gid;     /* group ID of owner */
   dev_t     st_rdev;    /* device ID (if special file) */
   off_t     st_size;    /* total size, in bytes */
   blksize_t st_blksize; /* blocksize for file system
I/O */
   blkcnt_t  st_blocks;  /* number of 512B blocks allocated */
   time_t    st_atime;   /* time of last access */
   time_t    st_mtime;   /* time of last modification */
   time_t    st_ctime;   /* time of last status change */
};
```

Questions?

Exercise

Write a program that takes a path as input and iterates through the contents of the directory:

- If in encounters a subdirectory, print the name, type and the number of files(all types) in it.
- If it encounters a file, print the name, type , owner, size and the Inode number.
- If anything else, print the name

Expected Output

```
baod@baod-MS-7D25:~/csci4061-spring/Recitations/spring_2022/05$ ./solution .
Regular File: exercise.c
        Owner: 1000
        Size: 758.000000
        Inode: 30413854
Regular File: file_exists.c
        Owner: 1000
        Size: 273.000000
        Inode: 30413878
Regular File: mkdir.c
        Owner: 1000
        Size: 825.000000
        Inode: 30413880
Regular File: solution.c
        Owner: 1000
        Size: 1818.000000
        Inode: 30413875
Regular File: getcwd.c
        Owner: 1000
        Size: 1225.000000
        Inode: 30413882
Directory: test
        Entries: 2
Regular File: is_dir.c
        Owner: 1000
        Size: 336.000000
        Inode: 30413879
Regular File: myls.c
        Owner: 1000
        Size: 1288.000000
        Inode: 30413881
Regular File: rec05-slides.pptx
        Owner: 1000
        Size: 135878.000000
        Inode: 30413874
Regular File: solution
        Owner: 1000
        Size: 12984.000000
        Inode: 30411779
Regular File: perm.c
        Owner: 1000
        Size: 290.000000
        Inode: 30413877
baod@baod-MS-7D25:~/csci4061-spring/Recitations/spring_2022/05$
```