Lab 08: Files and Directories

Directory Access

Directories should not be accessed with the ordinary `open`, `close` and `read` functions. Instead, they require specialized functions whose corresponding names end with `"dir"`: `opendir`, `closedir` and `readdir`.

The `opendir` function provides a handle of type `DIR *` to a directory stream that is positioned at the first entry in the directory.

```
SYNOPSIS

   #include <dirent.h>

   DIR *opendir(const char *dirname);
                                                    POSIX
```

If successful, `opendir` returns a pointer to a directory object. If unsuccessful, `opendir` returns a null pointer and sets `errno`.

The `DIR` type, which is defined in `dirent.h` represents a directory stream. A directory stream is an ordered sequence of all of the directory entries in a particular directory. The order of the entries in a directory stream is not necessarily alphabetical by file name.

The `readdir` function reads a directory by returning successive entries in a directory stream pointed to by `dirp`. The `readdir` returns a pointer to a `struct dirent` structure containing information about the next directory entry. The `readdir` moves the stream to the next position after each call.

```
SYNOPSIS

   #include <dirent.h>

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

If successful, `readdir` returns a pointer to a `struct dirent` structure containing information about the next directory entry. If unsuccessful, `readdir` returns a `NULL` pointer and sets `errno`. The only mandatory error is `EOVERFLOW`, which indicates that the value in the structure to be returned cannot be represented correctly. The `readdir` function also returns `NULL` to indicate the end of the directory, but in this case it does not change `errno`.

The `closedir` function closes a directory stream, and the `rewinddir` function repositions the directory stream at its beginning. Each function has a `dirp` parameter that corresponds to an open directory stream.

SYNOPSIS

```
#include <dirent.h>
int closedir(DIR *dirp);
void rewinddir(DIR *dirp);                                          POSIX
```

If successful, the `closedir` function returns 0. If unsuccessful, it returns –1 and sets `errno`. The `closedir` function has no mandatory errors. The `rewinddir` function does not return a value and has no errors defined.

Accessing file status information

This section describes three functions for retrieving file status information. The `fstat` function accesses a file with an open file descriptor. The `stat` and `lstat` functions access a file by name.

The `lstat` and `stat` functions each take two parameters. The `path` parameter specifies the name of a file or symbolic link whose status is to be returned. If `path` does not correspond to a symbolic link, both functions return the same results. When `path` is a symbolic link, the `lstat` function returns information about the link whereas the `stat` function returns information about the file referred to by the link. The `buf` parameter points to a user-supplied buffer into which these functions store the information.

SYNOPSIS

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

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

POSIX

If successful, these functions return 0. If unsuccessful, they return –1 and set `errno`. The `restrict` modifier on the arguments specifies that `path` and `buf` are not allowed to overlap.

The `struct stat` structure, which is defined in `sys/stat.h`, contains at least the following members.

```
dev_t     st_dev;       /* device ID of device containing file */
ino_t     st_ino;       /* file serial number */
mode_t    st_mode;      /* file mode */
nlink_t   st_nlink;     /* number of hard links */
uid_t     st_uid;       /* user ID of file */
gid_t     st_gid;       /* group ID of file */
off_t     st_size;      /* file size in bytes (regular files) */
                        /* path size (symbolic links) */
time_t    st_atime;     /* time of last access */
time_t    st_mtime;     /* time of last data modification */
time_t    st_ctime;     /* time of last file status change */
```

Task: Implement *ls* command:

NAME

ls - list directory contents

SYNOPSIS

ls [OPTION]... [FILE]...

DESCRIPTION

List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuSUX nor --sort.

Mandatory arguments to long options are mandatory for short options too.

-l use a long listing format

ls-l is used for listing directory contents in long format.