

opendir()

The `opendir()` function opens a directory stream for reading. It's defined in the "dirent.h" system header.

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

// "dirname" is the absolute/relative path to the directory
DIR *opendir(const char *dirname);
```

Returns a pointer to a DIR if success. In case of failure, returns NULL and sets "errno".

Common "errno" values:

- "ENOENT" → Directory doesn't exist
- "EACCES" → Permission denied
- "ENOTDIR" → Path is not a directory

DIR

? In older UNIX systems (pre-POSIX), "DIR" and related types were sometimes defined in "sys/types.h".

Some systems require both "dirent.h" and "sys/types.h" for completeness.

DIR is a directory stream type, defined in "dirent.h". It represents an open directory and is used to read its contents.

- Created by "opendir()", freed by "closedir()"
- It's an opaque type because its internal structure is system-dependent

Example Usage

List directory contents:

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

int main(void)
{
    DIR *dir;
    struct dirent *entry;

    dir = opendir(".");
    if (!dir)
    {
        perror("opendir failed");
        return (1);
    }
    while ((entry = readdir(dir))
        printf("%s\n", entry->d_name);
    closedir(dir);
    return (0);
}
```

List regular files only:

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

int main(void)
{
    DIR *dir;
    struct dirent *entry;

    dir = opendir(".");
    if (!dir)
        return (1);
    while ((entry = readdir(dir)))
    { // "DT_REG" filters by regular files
        if (entry->d_type == DT_REG)
            printf("File: %s\n", entry->d_name);
    }
    closedir(dir);
    return (0);
}
```