

readdir()

The `readdir()` function reads the next entry from a directory stream opened by "opendir()". It's defined in the "dirent.h" system header.

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

// "dirp" is the directory stream to read from
struct dirent *readdir(DIR *dirp);
```

If success, returns a pointer to the next "struct dirent" entry.

In case of failure, returns NULL and sets "errno".

If the end of directory is reached, returns NULL and "errno" is set to 0.

uint8_t

The `uint8_t` is a fixed-width unsigned integer type from the C Standard Library "stdint.h".

Guarantees exactly 8 bits, with a range from 0 to 255 values.

It's used when precise control over memory size is needed.

_DIRENT_HAVE_D_TYPE

`_DIRENT_HAVE_D_TYPE` is a preprocessor macro defined by some systems to indicate "d_type" support for "struct dirent".

It's set internally by the C library, not part from POSIX standard, but widely available.

Practical usage:

```
#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)) != NULL)
    {
        #ifdef _DIRENT_HAVE_D_TYPE
            printf("%s [Type: %d]\n", entry->d_name, entry->d_type);
        #else
            printf("%s [Type: Unknown]\n", entry->d_name);
        #endif
    }
    closedir(dir);
    return (0);
}
```

struct dirent

The `struct dirent` represents a single entry in a directory. It may be a file or a subdirectory. It's defined in "dirent.h".

Key POSIX standard members of "struct dirent":

- `char[] d_name` → Null-terminated filename
- `uint8_t d_type` → File type
 - Optional, check system support with "_DIRENT_HAVE_D_TYPE"
 - If "d_type" is "DT_UNKNOWN", or unsupported, use "stat()"
 - Common "d_type" values in Linux:
 - "DT_REG" (8) → Regular file
 - "DT_DIR" (4) → Directory
 - "DT_LNK" (10) → Symbolic link
 - "DT_UNKNOWN" (0) → Unknown type (fallback)

Example Usage

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

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

    dir = opendir(".");
    if (!dir)
    {
        perror("opendir");
        return (1);
    }
    while ((entry = readdir(dir)) != NULL)
        printf("%s [Type: %d]\n", entry->d_name, entry->d_type);
    closedir(dir);
    return (0);
}
```