2.3 Search Path

By default, the preprocessor looks for header files included by the quote form of the directive `#include "file"` first relative to the directory of the current file, and then in a preconfigured list of standard system directories. For example, if `/usr/include/sys/stat.h` contains `#include "types.h"`, GCC looks for `types.h` first in `/usr/include/sys`, then in its usual search path.

For the angle-bracket form `#include <file>`, the preprocessor's default behavior is to look only in the standard system directories. The exact search directory list depends on the target system, how GCC is configured, and where it is installed. You can find the default search directory list for your version of CPP by invoking it with the `-v` option. For example,

```
cpp -v /dev/null -o /dev/null
```

There are a number of command-line options you can use to add additional directories to the search path. The most commonly-used option is `-Idir`, which causes *dir* to be searched after the current directory (for the quote form of the directive) and ahead of the standard system directories. You can specify multiple `-I` options on the command line, in which case the directories are searched in left-to-right order.

If you need separate control over the search paths for the quote and angle-bracket forms of the '#include' directive, you can use the `-iquote` and/or `-isystem` options instead of `-I`. See Invocation, for a detailed description of these options, as well as others that are less generally useful.

If you specify other options on the command line, such as `-I`, that affect where the preprocessor searches for header files, the directory list printed by the `-v` option reflects the actual search path used by the preprocessor.

Note that you can also prevent the preprocessor from searching any of the default system header directories with the `-nostdinc` option. This is useful when you are compiling an operating system kernel or some other program that does not use the standard C library facilities, or the standard C library itself.