

NAME

`lint` — a C program verifier

SYNOPSIS

`lint [-abchmnpvux] file ...`

DESCRIPTION

Lint attempts to detect features of the C program *files* which are likely to be bugs, or non-portable, or wasteful. It also checks the type usage of the program more strictly than the compilers. Among the things which are currently found are unreachable statements, loops not entered at the top, automatic variables declared and not used, and logical expressions whose value is constant. Moreover, the usage of functions is checked to find functions which return values in some places and not in others, functions called with varying numbers of arguments, and functions whose values are not used.

By default, it is assumed that all the *files* are to be linked together; they are checked for mutual compatibility. Function definitions for the standard C and UNIX system libraries are available to *lint* by default. A subset of the standard C library is used when *lint* is invoked with the `-p` option.

Any number of the options in the following list may be used. The `-D`, `-U`, and `-I` options of *cc(1)* are also recognized as separate arguments.

- `-p` Attempt to check portability to the IBM and GCOS dialects of C.
- `-h` Apply a number of heuristic tests to attempt to intuit bugs, improve style, and reduce waste.
- `-b` Report **break** statements that cannot be reached. (This is not the default because, unfortunately, most *lex* and many *yacc* outputs produce dozens of such comments.)
- `-v` Suppress complaints about unused arguments in functions.
- `-x` Report variables referred to by extern declarations, but never used.
- `-a` Report assignments of long values to int variables.
- `-c` Complain about casts which have questionable portability.
- `-u` Do not complain about functions and variables used and not defined, or defined and not used (this is suitable for running *lint* on a subset of files out of a larger program).
- `-n` Do not check compatibility against the standard library.
- `-m` Do not complain about user-defined functions that have the same name as functions in a referenced library. (The difference between this and the previous option is that the `-m` option will still allow compatibility of usage to be checked for library functions not redefined by the user.)

Exit(2) and other functions which do not return are not understood; this causes various lies.

Certain conventional comments in the C source will change the behavior of *lint*:

```
/*NOTREACHED*/
```

at appropriate points stops comments about unreachable code.

```
/*VARARGSn*/
```

suppresses the usual checking for variable numbers of arguments in the following function declaration. The data types of the first *n* arguments are checked; a missing *n* is taken to be 0.

```
/*NOSTRICT*/
```

shuts off strict type checking in the next expression.

```
/*ARGSUSED*/
```

turns on the `-v` option for the next function.

`/*LINTLIBRARY*/`

at the beginning of a file shuts off complaints about unused functions in this file.

FILES

<code>/usr/lib/lint[12]</code>	programs
<code>/usr/lib/lib-lc</code>	declarations for standard functions
<code>/usr/lib/lib-port</code>	declarations for portable functions
<code>/usr/tmp/lint.*</code>	temporaries

SEE ALSO

`cc(1)`.