

**NAME**

`cc` – C compiler

**SYNOPSIS**

`cc [-c] [-p] [-f] [-Dn=v] [-Idir] [-O] [-S] [-P] [-Un] files ...`

**DESCRIPTION**

`Cc` is the UNIX C compiler. It accepts three types of arguments:

Arguments whose names end with ‘.c’ are taken to be C source programs; they are compiled, and each object program is left on the file whose name is that of the source with ‘.o’ substituted for ‘.c’. The ‘.o’ file is normally deleted, however, if a single C program is compiled and loaded all at one go.

The following flags are interpreted by `cc`. See `ld(I)` for load-time flags.

- c** Suppress the loading phase of the compilation, and force an object file to be produced even if only one program is compiled.
- p** Arrange for the compiler to produce code which counts the number of times each routine is called; also, if loading takes place, replace the standard startup routine by one which automatically calls the *monitor(III)* subroutine at the start and arranges to write out a *mon.out* file at normal termination of execution of the object program. An execution profile can then be generated by use of *prof(I)*.
- f** In systems without hardware floating-point, use a version of the C compiler which handles floating-point constants and loads the object program with the floating-point interpreter. Do not use if the hardware is present.
- D** The name *n* is *defined*, and is given the value *v*, if specified.
- O** Invoke an object-code optimizer.
- S** Compile the named C programs, and leave the assembler-language output on corresponding files suffixed ‘.s’.
- P** Run only the macro preprocessor on the named C programs, and leave the output on corresponding files suffixed ‘.i’.
- U** The name *n* is *undefined*.
- I** The *include* preprocessor statement looks in directory *dir* if it can’t find the specified file in the local directory or in **/usr/include**.

Other arguments are taken to be either loader flag arguments, or C-compatible object programs, typically produced by an earlier `cc` run, or perhaps libraries of C-compatible routines. These programs, together with the results of any compilations specified, are loaded (in the order given) to produce an executable program with name **a.out**. If desired, a different name can be used; see the **-o** option of `ld(I)`.

**FILES**

<code>file.c</code>	input file
<code>file.o</code>	object file
<code>a.out</code>	loaded output
<code>/tmp/ctm?</code>	temporary
<code>/lib/c[01]</code>	compiler
<code>/lib/fc[01]</code>	floating-point compiler
<code>/lib/c2</code>	optional optimizer
<code>/lib/cpp</code>	pre-processor
<code>/lib/crt0.o</code>	runtime startoff
<code>/lib/mcrt0.o</code>	runtime startoff of profiling

/lib/fcrt0.o	runtime startoff for floating-point interpretation
/lib/libc.a	C library; see section III.
/lib/liba.a	Assembler library used by some routines in libc.a

**SEE ALSO**

*C Reference Manual* by D. M. Ritchie.

*Programming in C – A Tutorial* by B. W. Kernighan.

adb(I), cdb(I), ld(I), prof(I), monitor(III)

**DIAGNOSTICS**

The diagnostics produced by C itself are intended to be self-explanatory. Occasional messages may be produced by the assembler or loader (see *as*(I) and *ld*(I)). Of these, the most mystifying are from the assembler, in particular “m,” which means a multiply-defined external symbol (function or data).