The New C Standard (Excerpted material)

An Economic and Cultural Commentary

Derek M. Jones
derek@knosof.co.uk

5.1.2.1 Freestanding environment

Commentary

Rationale As little as possible is said about freestanding environments, since little is served by constraining them.

---

freestanding environment startup

In a freestanding environment (in which C program execution may take place without any benefit of an operating system), the name and type of the function called at program startup are implementation-defined. 155

Commentary

The function called at program startup need not be spelled `main`.

The parameters to the called function are defined by the implementation and can be very different from those defined here for `main` (in a hosted environment). If there is no operating system, there are not likely to be any command line parameters, but information may be passed into the function called.

Without the benefit of an operating system, the issue of how to provide the ability to startup different programs becomes important. Having available a set of functions, which can be selected amongst just prior to program startup, provides one solution.

Common Implementations

In many cases there is no named program at all. Switching on, or resetting, the freestanding host causes the processor instruction pointer to be set to some predefined location in storage, whichever instructions are located at that and subsequent locations being executed. Traditionally there is a small bootstrap loader at this location, which copies a larger program from a storage device into memory and jumps to the start of that program (it might be a simple operating system). In other cases that storage device is ROM and the program will already be in memory at the predefined location. Translated program instructions are executed directly from the storage device. Once switched on, the host may have buttons that cause the processor to be reset to a different location, causing different functions to be invoked on startup.

---

environment freestanding implementation

Any library facilities available to a freestanding program, other than the minimal set required by clause 4, are implementation-defined. 156

Commentary

conforming freestanding implementation

The Committee wanted it to be practical to create a conforming implementation for a freestanding environment. In such an environment the basic facilities needed to provide all of the functionality required by the full library are often not available. The requirement to supply a full library needed to be relaxed. The full language can always be supported, although this may mean supplying a set of internal functions for handling arithmetic operations on **`long long`** and floating-point types. If a particular program does not make use of **`long long`** or floating-point types, a linker will usually take the opportunity to reduce the size of the final executable by not linking in these internal library functions.

Common Implementations

Freestanding environments vary enormously between those that offer a few hundred bytes of storage and no library, to those having megabytes of memory and an operating system offering full POSIX realtime functionality.

Coding Guidelines

Care needs to be taken, in a freestanding environment, when using library facilities beyond the minimal set required by Clause 4. There is no requirement for an implementation that supplies additional functions to completely follow the specification given in the standard. It is not uncommon to see only partial functionality implemented.

157 The effect of program termination in a freestanding environment is implementation-defined.

Commentary

It may not even be possible to terminate a program in a freestanding environment. The program may be the only execution environment there is. Switching the power off may be the only way of terminating a program.

Common Implementations

A **return** statement executed from the function called on program startup, or a call to `exit` (if the freestanding implementation supports such a call) may return control to a host operating system or return to the random address held on the top of the stack.

Coding Guidelines

Specifying how program termination, in a freestanding environment, should be handled is outside the scope of these coding guidelines.

References