Developer's Guide

This is the developer's guide to the Memtest86+ source code. The user's guide can be found in the README.md file in the top level directory.

Code Organisation

The source code is divided into the following categories, each contained in a subdirectory of the same name:

- app

  The main application and framework for running the memory tests.

- boot

  The code that runs from the BIOS or bootloader entry point to the start of the main application.

- lib

  The subset of the C standard library that is used by Memtest86+ plus other hardware-independent low-level support functions.

- system

  Low-level support functions that interface to the hardware.

- tests

  The individual memory tests.

The boot code is mostly written in AT&T syntax x86 assembly language. The remaining code is written in C with a smattering of inline assembly code.

Each category is further subdivided into multiple source files, splitting the code into small units of closely related functionality. For the C code, the API for each unit is defined in a header (`.h`) file and the implementation (if required) is found in the correspondingly named source (`.c`) file.

Code Documentation

Doxygen can be used to automatically generate HTML documentation for the API for each code unit from the comments in the C header files. To regenerate the documentation, change directory into the `doc` subdirectory and run `doxygen`.

Coding Conventions

C Code

Macro names and enumeration values are written in upper case separated by underscores. All other identifiers are written in lower case separated by underscores. Both excessive length and excessive abbreviation are avoided.

Line length is normally limited to 120 characters.

Indentation is 4 spaces. No hard tab characters are used.

Opening braces for function bodies are put on a new line. Other opening braces are put on the same line as the construct that introduces them.

The C11 dialect is used. In particular, variable declarations and statements are interspersed and loop variables are declared within the `for` loop construct.

Assembly Code

Labels are written in lower case separated by underscores. Op-codes and register names are written in lower case.

Line length is normally limited to 120 characters.

Indentation is 8 spaces using hard tab characters.

The C preprocessor is used for defining constant values and expressions. Macro names are written in upper case separated by underscores.