P&E, GNU, and ELF/DWARF

Using GNU Cross-compiling Tools and the ELF/DWARF File Format with P&E Microcomputer Systems In-circuit Debugger and Flash Programming Software

This document provides a general overview of using the GCC cross-compiler with P&E's ICD and PROG software in a Windows environment. It explains the steps required to produce ELF/DWARF target code with GNU tools, code that may be downloaded to the target system and debugged using P&E software.

Detailed explanation regarding GNU command-line switches, make files, and program linking will not be provided here. For comprehensive information regarding GNU software, see the appropriate manual at the GNU website:

http://www.gnu.org/manual/manual.html

Supported Formats

P&E's In-circuit Debugger may be used to debug C source code compiled to the ELF/DWARF file format. The host GNU installation must be configured to produce an ELF/DWARF file for the desired target processor. Examples of GNU cross-compilers supported by the P&E In-circuit Debugger include powerpc-eabi-gcc and m68k-elf-gcc.

Together, the ELF and DWARF standards provide a target binary file that includes both executable and debug information.

The ELF file format is based on the System V Application Binary Interface, Edition 4.1. This standard provides the binary executable format for the target application. The specification may be downloaded at the SCO website:

http://www.sco.com/developers/devspecs/

The DWARF debugging format provides symbolic debugging information to the P&E In-circuit Debugger. Currently, the debugger will only accept files compiled with version 2.0 of the DWARF specification. The debugger will also accept ELF files with no debug information. The specification may be downloaded from Eager Consulting:

http://www.eagercon.com/dwarf/dwarf3std.htm

Specification supplements are available for various target processors. These supplements include target-specific additions to both the ELF and DWARF standards.

The specification for the C programming language (INCITS/ISO/IEC 9899-1999) may be purchased from NSSN:

http://www.nssn.org/

GCC Command Line Options

To produce debug information that is readable by the P&E In-circuit Debugger, include the following switches with the options passed to GCC during the application build process:

-g : produce debugging information in the operating system's native format

-gdwarf-2 : compile with DWARF 2.0 debugging information

Check the compiler output messages to be sure that the -gdwarf-2 flag is supported by the host GNU installation.

For compatibility between the P&E debugger and GCC, it may be necessary to provide full paths to source files. The user would pass the full path names to GCC on the command-line, in a batch file, or in a make file. If the P&E In-circuit Debugger is unable to locate a source file that is listed in its Source Modules window, try specifying full paths during compilation.

Loading ELF/DWARF Files in P&E's In-circuit Debugger

To work with the target file in the debugger, issue the BETA command followed by the HLOAD command. Then, select the desired ELF/DWARF file.

The debugger can load debug information for a target application previously programmed to flash memory, or it can download the application to the target RAM as well as load its debug information. If downloading code to the target RAM, the target system must be configured properly before issuing the HLOAD command.

ELF Program Headers and GNU

Program headers, included in every executable ELF/DWARF file, describe how the application code is to be loaded to the target. Two values in each header entry, defined by the System V ABI as p_paddr and p_vaddr, are available to provide a load address for a particular group of code. For executable files, the p_vaddr field is typically used to provide the load address. However, GNU tools may utilize the p_paddr field instead.

When the P&E debugger loads the target ELF/DWARF file, it may detect the use of the non-standard p_paddr field in the ELF program header. In this instance, the debugger will display a dialog box that will ask the user what to do. Generally, when using the GNU tools, click "Yes" in the dialog box to load code using the non-standard p_paddr field.

This situation arises when the GNU linker encounters the AT command in a linker script. The AT command may be used to specify both an execution address and a load address for a section of object code. For more information about the GNU linker (ld), see the GNU Manuals Online. To verify which program header field that GNU utilizes, examine the program headers of the ELF/DWARF file using a utility such as GNU readelf.

For detailed information about ELF program headers, see the System V ABI, Edition 4.1.

Using GNU with P&E's Flash Programming Software

Currently, the P&E Flash Programming Software accepts the Motorola S-record format for programming target data. Convert the target ELF/DWARF file to the Motorola S-record format before programming the target flash memory. (However, the P&E debugger can download code to RAM without converting to S-records). To convert your target file to S-record format, use the GNU utility objcopy. Include the following option when executing objcopy:

-O srec : convert to Motorola S-record format

Be sure to check the objcopy output messages to determine if this format is supported by the host GNU installation.

The converted S-record file can now be used with P&E's Flash Programming Software.

Examining Contents of ELF/DWARF Files

Use the GNU readelf utility, part of the GNU Binutils package, to examine the contents of an ELF/DWARF file. The utility is useful for displaying various header information, debug information, and application data.