NAME

obpsym – Kernel Symbolic Debugging for OpenBoot Firmware

SYNOPSIS

modload –p misc/obpsym

AVAILABILITY

SUNWcsr

DESCRIPTION

obpsym is a kernel module that installs OpenBoot callback handlers that provide kernel symbol information to OpenBoot. OpenBoot firmware user interface commands use the callbacks to convert numeric *addresses* to kernel symbol names for display purposes, and to convert kernel symbol names to numeric *literals* allowing symbolic names to be used as input arguments to user interface commands.

Once **obpsym** is installed, kernel symbolic names may be used anywhere at the Open-Boot firmware's user interface command prompt in place of a literal (numeric) string. For example, if **obpsym** is installed, the OpenBoot firmware commands **ctrace** and **dis** typically display symbolic names and offsets in the form *modname:symbolname* + *offset.* User interface Commands such as **dis** can be given a kernel symbolic name such as **ufs:ufs_mount** instead of a numeric address.

Placing the command

forceload: misc/obpsym

into the **system**(4) file forces the kernel module **misc/obpsym** to be loaded and activates the kernel callbacks during the kernel startup sequence.

obpsym may be useful as a kernel debugger in situations where other kernel debuggers are not useful. For example, on SPARC machines, if **obpsym** is loaded, you may be able to use the OpenBoot firmware's **ctrace** command to display symbolic names in the stack backtrace after a watchdog reset.

Kernel Symbolic Name Syntax

The syntax for a kernel symbolic name is:

[*module-name* **:**] *symbol-name*

Where *module-name* is the name of the kernel module that the symbol *symbol-name* appears in. A **NULL** module name is taken as "all modules, in no particular order" by **obpsym**. The module name **unix** is equivalent to a **NULL** module name, so that conflicts with words defined in the firmware's vocabulary can be avoided.

Typically, OpenBoot firmware reads a word from the input stream and looks the word up in its internal *vocabulary* before checking if the word is a *literal*. Thus, kernel symbols, such as **reset** may be given as **unix:reset** to avoid the unexpected side effect of the firmware finding and executing a matching word in its vocabulary.

FILES

/etc/system system configuration information file
/platform/*platform-name***/kernel/misc/obpsym**

SEE ALSO

kadb(1M), **kernel**(1M), **modload**(1M), **modunload**(1M), **system**(4), **uname**(1)

OpenBoot Command Reference, Version 2.x, 1994, SunSoft Press, ISBN Number 0-13-107194-7

WARNINGS

Some OpenBoot firmware user interface commands may use system resources incompatibly with the way they are used by the Unix kernel. These commands and the use of this feature as a kernel debugger may cause interactions that the Unix kernel is not prepared to deal with. If this occurs, the Unix kernel and/or the OpenBoot firmware user interface commands may react unpredictably and may panic the system, or may hang or may cause other unpredictable results. For these reasons, the use of this feature is only minimally supported and recommended to be used only as a kernel debugger of "last resort".

NOTES

platform-name can be found using the **–i** option of **uname**(1)

obpsym is supported only on architectures that support OpenBoot firmware.

On some systems, OpenBoot must be completely RAM resident so the **obpsym** symbol callback support can be added to the firmware, if the firmware doesn’t include support for the symbol callbacks. On these systems, **obpsym** may complain that it requires that "you must use ramforth to use this module".

See the **OpenBoot Command Reference** manual for details on how to use the *ramforth* command, how to place the command into *nvramrc*, and how to set *use-nvramrc?* to *true*. On systems with version 1.x OpenBoot firmware, *nvramrc* doesn’t exist, and the *ramforth* command must be typed manually after each reset, in order to use this module.

Once installed, the symbol table callbacks can be disabled by using the following OpenBoot firmware command:

0 0 set-symbol-lookup