NAME

kld - dynamic kernel linker facility

DESCRIPTION

The LKM (Loadable Kernel Modules) facility has been deprecated in FreeBSD 3.0 and above in favor of the **kld** interface. This interface, like its predecessor, allows the system administrator to dynamically add and remove functionality from a running system. This ability also helps software developers to develop new parts of the kernel without constantly rebooting to test their changes.

Various types of modules can be loaded into the system. There are several defined module types, listed below, which can be added to the system in a predefined way. In addition, there is a generic type, for which the module itself handles loading and unloading.

The FreeBSD system makes extensive use of loadable kernel modules, and provides loadable versions of most file systems, the NFS client and server, all the screen-savers, and the Linux emulator. **kld** modules are placed by default in the */boot/kernel* directory along with their matching kernel.

The **kld** interface is used through the kldload(8), kldunload(8) and kldstat(8) programs.

The kldload(8) program can load either a.out(5) or ELF formatted loadable modules. The kldunload(8) program unloads any given loaded module, if no other module is dependent upon the given module. The kldstat(8) program is used to check the status of the modules currently loaded into the system.

Kernel modules may only be loaded or unloaded if the system security level *kern.securelevel* is less than one.

MODULE TYPES

Device Driver modules

New block and character device drivers may be loaded into the system with **kld**. Device nodes for the loaded drivers are automatically created when a module is loaded and destroyed when it is unloaded by devfs(5). You can specify userland programs that will run when new devices become available as a result of loading modules, or existing devices go away when modules are unloaded, by configuring devd(8).

FILES

/boot/kernel	directory containing module binaries built for the kernel also residing in the directory.
/usr/include/sys/module.h	file containing definitions required to compile a **kld** module
/usr/share/examples/kld	example source code implementing a sample kld module

SEE ALSO

kldfind(2), kldfirstmod(2), kldload(2), kldnext(2), kldstat(2), kldunload(2), devfs(5), devd(8), kldload(8), kldstat(8), kldunload(8), sysctl(8)

HISTORY

The **kld** facility appeared in FreeBSD 3.0 and was designed as a replacement for the **lkm** facility, which was similar in functionality to the loadable kernel modules facility provided by SunOS 4.1.3.

AUTHORS

The **kld** facility was originally implemented by Doug Rabson <*dfr@FreeBSD.org*>.

BUGS

If a module B, is dependent on another module A, but is not compiled with module A as a dependency, then kldload(8) fails to load module B, even if module A is already present in the system.

If multiple modules are dependent on module A, and are compiled with module A as a dependency, then kldload(8) loads an instance of module A when any of the modules are loaded.

If a custom entry point is used for a module, and the module is compiled as an ‘ELF’ binary, then kldload(8) fails to execute the entry point.

kldload(8) points the user to read dmesg(8) for any error encountered while loading a module.

When system internal interfaces change, old modules often cannot detect this, and such modules when loaded will often cause crashes or mysterious failures.