Q: When I run gdb (Release 3) or kgdb (Release 2) on my loadable kernel server, the slave system doesn't stop at the breakpoints I've set. What am I doing wrong?

Q: The symbols in my kernel server seem to have addresses that are relative to the origin of the kernel server, not the origin of the kernel. What am I doing wrong? Are these related?

A: When you compile a kernel server with kl_ld, it produces a **relocatable** object module. This module has not been linked against the kernel, and the addresses of its symbols start at 0. When you invoke kl_util -a, it edits the server's links against the currently running kernel, and produces a **loadable** module, with addresses that reflect its actual location in the kernel.

Normally, the loadable module is simply loaded into the kernel, and not copied into the file system. However, if you invoke kl_ld with the **-d** option, then when you later execute kl_util -a, the loader will leave behind a copy of the loadable object file (-d option in bold face below):

```
mysystem% kl_ld -n myserver -l Load_commands -u \
                    Unload_Commands -i instance_variables \
                    -d  myserver_loadable \
                    -o myserver_reloc \
                    myserver_object_file.o other_object.o
mysystem% ls *reloc *loadable
                    myserver_reloc
mysystem%


mysystem% kl_util -a myserver_reloc
mysystem% ls *reloc *loadable
                    myserver_reloc myserver_loadable
mysystem%
```

(Note that you must specify the _loadable suffix as part of <name>.   The system doesn't do that for you.)

It is the _loadable module against which you must execute gdb(kgdb), not the _reloc object.

If you run gdb(kgdb) against the relocatable module, or against the wrong loadable module, the addresses seen are incorrect, and breakpoints do not work.

If you have any question as to whether the module your using is the right one, you can look at its addresses with the nm command, and compare them to the addresses of the same symbols in the kernel you're debugging.   They should be the same.

QA722

Valid for 2.0, 3.0