Mark Loiseau

"Stand a little less between me and the sun."

-Diogenes

“Hello World” Loadable Kernel Module

Posted on **April 21, 2012**

A Loadable Kernel Module (LKM) allows modification or extension of a Unix-like operating system’s kernel without the need to recompile or reboot the machine. LKM functionality has been available to Linux users since 2.6, and similar functionality is available in BSD, OSX and most Unix variants.

The following tutorial outlines the process of writing, compiling and inserting a “Hello World” LKM into your running kernel. An archive of the source files for this post are available here. I performed the following steps on computers running Ubuntu 11.10 and 12.10, but the same process should work (with minor modifications) on any Linux box.

How to Create an LKM:

1. Install and prepare module-assistant.
2. Create hello.c and the Makefile (get both here).
3. Compile the code.
4. Insert the compiled module into the running kernel.
5. Remove the module when you’re finished.

1. Install and prepare the kernel headers

Debian and Ubuntu both provide provide module-assistant, a convenient package that contains all you need to write your own LKM. Install and configure it with the following command:

```
$ sudo -i
# apt-get install module-assistant
# m-a prepare
```

module-assistant doesn’t do anything *too* fancy. It’s just a front-end that manages kernel source packages. The following command would also install the packages we need:

```
$  sudo apt-get install build-essential linux-headers-$(uname -r)
```

2. Create the source and the Makefile for your LKM

Here is the C source code for the hello world LKM, hello.c:

```
// Defining __KERNEL__ and MODULE allows us to access kernel-level code not usual
#undef __KERNEL__
#define __KERNEL__

#undef MODULE
#define MODULE

// Linux Kernel/LKM headers: module.h is needed by all modules and kernel.h is ne
#include <linux/module.h>    // included for all kernel modules
#include <linux/kernel.h>     // included for KERN_INFO
#include <linux/init.h>         // included for __init and __exit macros

static int __init hello_init(void)
{
    printk(KERN_INFO "Hello world!\n");
    return 0;    // Non-zero return means that the module couldn't be loaded.
}

static void __exit hello_cleanup(void)
{
    printk(KERN_INFO "Cleaning up module.\n");
}

module_init(hello_init);
module_exit(hello_cleanup);
```

The function given as an argument to module_init runs when the LKM is inserted into the kernel. Since it is given as the argument to module_exit, hello_cleanup() runs when the LKM is removed.

Here is the source of the Makefile:

```
obj-m := hello.o
KDIR := /lib/modules/$(shell uname -r)/build
PWD := $(shell pwd)

all:
$(MAKE) -C $(KDIR) M=$(PWD) modules

clean:
$(MAKE) -C $(KDIR) M=$(PWD) clean
```

2. Compile the code

To compile the code, go to your project directory (available as this archive/the one containing hello.c and the Makefile) and type “make.”

```
$ make
```

```
make -C /lib/modules/3.0.0-17-generic/build M=/var/www/lkm modules
make[1]: Entering directory `/usr/src/linux-headers-3.0.0-17-generic'
CC [M] /var/www/lkm/hello.o
Building modules, stage 2.
MODPOST 1 modules
CC /var/www/lkm/hello.mod.o
LD [M] /var/www/lkm/hello.ko
```

4. Insert the compiled module into the running kernel.

To insert your LKM into the running kernel, use insmod.

```
$ sudo insmod hello.ko
```

Our LKM uses printk(), which prints to the syslog. To see our message, tail the syslog:

```
$ tail /var/log/syslog
<snip>
Apr 20 16:27:39 laptop kernel: [19486.347191] Hello world!
```

On some systems, kernel messages are logged to a different file (i.e. /var/log/messages).

5. Remove the module when you’re finished.

Use rmmod to remove the LKM:

```
$ sudo rmmod hello
```

To verify that the LKM has been removed successfully, tail the syslog again:

```
$ tail /var/log/syslog
<snip>
Apr 20 16:29:23 laptop kernel: [19486.347191] Cleaning up module.
```

Congratulations! You’ve just created and compiled a Loadable Kernel Module.

Further Reading

The Linux Kernel Module Programming Guide
Anatomy of Linux loadable kernel modules
Loadable Kernel Module Programming and System Call Interception
(nearly) Complete Linux Loadable Kernel Modules

Incoming search terms:

- LINUX KERNEL MODULE TUTORIAL
- KERNEL MODULE EXAMPLE
- HELLO WORLD KERNEL MODULE
- KERNEL MODULE TUTORIAL
- LOADABLE KERNEL MODULE TUTORIAL
- LOADABLE KERNEL MODULE
- UBUNTU KERNEL MODULE COMPILE
- KERNEL MODULE MAKEFILE
- INSERTING SIMPLE LKM
- WRITING A KERNEL MODULE

This entry was filed under **Linux**, **Programming** and tagged **C**, **hello world**, **kernel**, **linux**, **lkm**, **loadable kernel module**, **tutorial**, **ubuntu** by **mark**. Bookmark the **permalink [http://blog.markloiseau.com/2012/04/hello-world-loadable-kernel-module-tutorial/]** .

7 thoughts on "“Hello World” Loadable Kernel Module"

David
on **December 9, 2012 at 5:07 am** said:

This guide is soooooo much better than the other ones I’ve found online so far. Plus you’ve got some great references that I didn’t see until now, except for the first one, The Linux Kernel Module Programming Guide. (By the way, there is a version that is 2 years newer at http://www.tldp.org/LDP/lkmpg/2.6/html/lkmpg.html)

I wish this were the one that came up at the top, or at least top 5, when I Googled “Linux kernel module”. As it is I found my way here because someone mentioned it in a superuser.com thread.

> smd
> on **July 12, 2013 at 8:40 am** said:
>
> This should be the first Google result for LKM.

Esteban

on **January 29, 2013 at 9:18 pm** said:

Hey, thank for the guide, very useful and up to date compared to the rest of the doc!

raghu

on **March 10, 2013 at 7:40 pm** said:

Great one man plain and simple keep going

Sandeep K Rai

on **April 5, 2013 at 1:27 pm** said:

Great work Man!! Keep doing………Good for kernel 3+ version where it can easily run… Unlike other hello world modules what other persons wrote on other link…

rupesh jain

on **July 3, 2013 at 12:57 pm** said:

i am learning embedded linux my self your tutorial is best for me.thanks you very much

smd

on **July 12, 2013 at 8:39 am** said:

Saved me days! Thank you! Thumbs up for module assistant