Linux Device Driver

LKM - Exporting Symbols Between Modules

https://community.ruggedboard.com

LKM- Exporting Symbols Between Modules

Rugged Board

What is Symbol?

- A symbol is a name given to a space in the memory which stores
- data (Variables, For reading and writing)
- instructions (Functions, for executing)
- So symbol in the programming language is either a variable or function.

What is Symbol table?

- Data Structure created by compiler containing all the symbols used in the program.
- Every kernel image that you build has a symbol table with it.
- The Linux kernel symbol table contains names and addresses of all the kernel symbols.
- When you install the kernel it will be present in /boot/System.map-<linux_version>

LKM- Exporting Symbols Between Modules

How to Export your symbols?

- When you define a new function in your module, the default behavior of this function is local, only the module in which the function is defined can access it, cannot be accessed by other modules.
- To export this module we need to use EXPORT_SYMBOL or EXPORT_SYMBOL_GPL.
- Once you export them, they will be available to other modules to use.

Difference between EXPORT_SYMBOL and EXPORT_SYMBOL_GPL

EXPORT_SYMBOL: The exported symbol can be used by any kernel module

EXPORT_SYMBOL_GPL: The exported symbol can be used by only GPL licensed code.

LKM- Exporting Symbols Between Modules

1. Write module which exports a function myadd performing addition of two numbers.
2. Write another module which uses the exported function.

module1.c

```
#include <linux/kernel.h>
#include <linux/module.h>

MODULE_LICENSE("GPL");

int myadd(int a, int b)
{
   pr_info("%s: Adding %d with %d\t Result:%d\n", __func__, a, b, a+b);
   return a+b;
}

EXPORT_SYMBOL(myadd);
```

```
static int module1_init(void)
{
   pr_info("%s: In init\n", __func__);
   return 0;
}

static void module1_exit(void)
{
   pr_info("%s: In exit\n", __func__);
}

module_init(module1_init);
module_exit(module1_exit);
```

LKM- Exporting Symbols Between Modules

Rugged Board

module2.c

```
#include <linux/kernel.h>
#include <linux/module.h>

MODULE_LICENSE("GPL");

extern int myadd(int a, int b);
static int module2_init(void)
{
    pr_info("%s: In init\n", __func__);
    pr_info("%s: Add:%d\n", __func__, myadd(3, 5));
    return 0;
}
static void module2_exit(void)
{
    pr_info("%s: In exit\n", __func__);
}
module_init(module2_init);
module_exit(module2_exit);
```

Makefile:

```
obj-m := module2.o
obj-m += module1.o
all:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

LKM- Exporting Symbols Between Modules

When we try to insert module2.ko first before inserting module1.ko we will get error

Connect-us

Developer Wiki

Open Discussions