Loadable Kernel Module

1 Sample Loadable Kernel Module

A module that prints "Hello World"

A new file `lkm_1.c` was created, with the following contents

Include Header Files

`linux/module.h` // for module related functions
`linux/kernel.h` // for the kernel priority macros available in <LINUX_SRC>/include/linux/kern_levels.h
`linux/init.h` // for using module_init & module_exit functions, so that new names can be given to functions init_module() and cleanup_module()

Set Up modinfo

```
MODULE_LICENCE("GPL");           // for making the module licence as GPL
MODULE_AUTHOR("name, details");  // giving author details to the module
```

Initialize Module

```
static int __init myinit(void)   // __init macro: denote the initializer
{
    printk(KERN_ALERT "Hello World\n");       // print function, logs output
                                              // priority KERN_ALERT = <1>
    return 0; // successful module initialization
}
```

Cleanup Module

```
static void __exit myexit(void)   // __exit macro: denote the cleanup function
{
    printk(KERN_ALERT "Bye World\n");
}
```

Assigning the init and exit functions

```
module_init(myinit);
module_exit(myexit);
```

2 Extended module to accept command line params

Program prints TRUE if argument is 1, FALSE if 0

A new file `lkm_2.c` was created, with the following contents additional to `lkm_1.c`

Additional header file

`linux/moduleparam.h` // for handling command line parameters

Set up parameter 'choice' (short int)

```
static short int choice = -1;

/* specifying the parameter:
 * module_param(param_name, data_type, permission_bits);
 * will expose the parameters in sysfs if the permission_bits are non-zero
 */

module_param(choice, short, S_IRUSR | S_IWUSR | S_IRGRP | S_IWGRP);
MODULE_PARM_DESC(choice, "A boolean choice variable"); //parameter description
```

Initialization and Cleanup

```
static int __init myinit(void)
{
    printk(KERN_ALERT "Hello Param World\n");
    if(choice == 0)
        printk(KERN_ALERT "Choice FALSE\n");
    else if(choice == 1)
        printk(KERN_ALERT "Choice TRUE\n");
    else if(choice == -1)
        printk(KERN_ALERT "Choice at default value\n");
    return 0;
}

static void __exit myexit(void)
{
    printk(KERN_ALERT "Bye Param World\n");
}
```

3 Compiling, inserting and removing LKM

Makefile

```
obj-m += lkm_1.o
obj-m += lkm_2.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
```

Compiling

```
 command "make"
```

Inserting the module

```
1. command:       sudo insmod lkm_1.ko
   syslog output: Hello World

2. command:       sudo insmod lkm_2.ko             // without a command line argument
   syslog output: Hello Param World
                  Choice at default value

3. command:       sudo insmod lkm_2.ko choice = 0  //with command line argument 0
   syslog output: Hello Param World
                  Choice TRUE

4. command:       sudo insmod lkm_2.ko choice = 1  //with command line argument 1
   syslog output: Hello Param World
                  Choice FALSE
```

Removing the module

```
1. command:       sudo rmmod lkm_1
   syslog output: Bye World

2. command:       sudo rmmod lkm_2
   syslog output: Bye Param World
```

4 Viewing the output

Use `dmesg`
or `tail -f /var/log/syslog`