Loadable Kernel Module:
A Brief Tutorial

Some Basic Commands - 1

- **`lsmod`**
  - Lists all the module presently running on kernel

- ***Module   Size   Used by***

- Pipeline with **`more`** or **`less`**

Some Basic Commands - 2

- **`modinfo <module_name>`**

- Filename
- License
- Author
- Description
- *etc.*

Loading / Unloading

- **`insmod`** ***`<module_name>`***

- **`rmmod`** ***`<module_name>`***

Checking Kernel Messages

- **`dmesg`**

- Pipeline with
  - More
  - Less
  - Tail
  - Tail -1

Checking Kernel Messages

- Messages are stored in
- **`/var/log/syslog`**

- See logs with
  - **`cat`**
  - **`tail`**
  - **`tail -f`**

Other importants

- **Uname**
  - **Prints system information**
- **Options**
  - **-a (kernel all info)**
  - **-s (kernel name)**
  - **-r (kernel-release)**
  - **-n -v -m -p -i -o**
- **apt-get install build-essential linux-headers-$(uname -r)**

Module Basics

- Headers
- **#include<linux/init.h>**
- **#include<linux/module.h>**
- **#include<linux/kernel.h>**

- Two modules are necessary
  - **Init**
  - **Exit**

Module Basics

- Register both functions through
- **module_init** (*init_name*);
- **module_exit** (*exit_name*);

- Printing messages
  - Through **PRINTK**
- `printk(KERN_ALERT "Some message\n", __FUNCTION__);`

Tainted module?

- MODULE_LICENSE("GPL");
- MODULE_AUTHOR("name");
- MODULE_DESCRIPTION("Simple");

Makefile

```
obj-m = sim_mod_1.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
```