OPERATING SYSTEM DESIGN

S Thenmozhi

Department of Computer Applications

OPERATING SYSTEM DESIGN

OS Structures & Kernel Programming

S Thenmozhi
Department of Computer Applications

PES UNIVERSITY
CELEBRATING 50 YEARS

Module Commands

- modinfo – display information about a kernel module
  - Parameters, license, description and dependencies
  - Ex: modinfo module_name
- lsmod – list all the loaded modules
  - lsmod | grep <module_name> - list specific module
  - Check in /proc/modules
- Insmod – load the given kernel module
  - insmod <module_name>.ko
  - Full path is needed
- rmmod – unload loadable modules

Kernel Modules

First Kernel Module

- Install the necessary packages
- Create a kernel module code
- Create a Makefile
- Build the kernel module
- Load the kernel module
- Check the kernel log
- Unload the kernel module

OPERATING SYSTEMS DESIGN

Kernel Modules

- Install the necessary packages
  - sudo apt-get update
  - sudo apt-get install build-essential linux-headers-$(uname -r)
- Create a kernel module code
  - Write hello.c
- Create a make file
  - Write Makefile
- Build the kernel module
  - make
- Load the kernel module
  - sudo insmod hello_world.ko
- Check the kernel log
  - dmesg | tail
- Unload the kernel module
  - sudo rmmod hello_world

Kernel Modules

hello.c

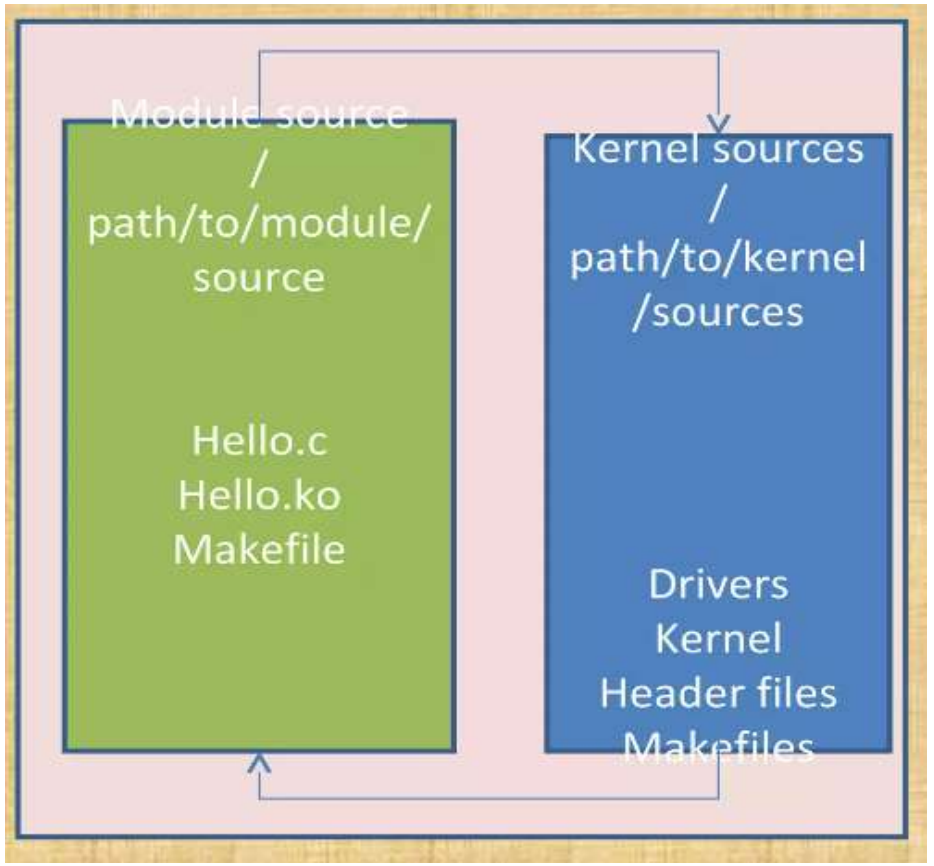
```
#include <linux/init.h>
#include <linux/module.h>
static int __init hello_init(void){
printk(“Hello:This is my first kernel module\n”);
return 0;
}
static void __exit hello_exit(void) {
 printk(“Bye: Module Unloaded\n”); }
module_init(hello_init);
module_exit(hello_exit);
MODULE_DESCRIPTION(“SAMPLE”);
MODULE_AUTHOR(“THENMOZHI”);
MODULE_LICENSE(“GPL”);
```

Kernel Modules

- Header specific to linux kernel <linux/xxxxx.h>
  - No access to usual c library
- An initialization function
  - called when the module is loaded using insmod/modprobe tool
  - Perform all the initialization functionality
  - Returns an error code
    - 0- success
    - negative value on failure, errors defined in header file
  - Declared by the module _init() macro
- A cleanup function
  - Called when the module is unloaded using rmmod tool
  - Perform all the clean-up functionality
  - Declared by the module_exit() macro
- Metadata Information
  - Module description, author, license

Compiling the module

- Kernel modules need to be compiled a bit differently from regular user space apps



PES UNIVERSITY
CELEBRATING 50 YEARS

Makefile

```
obj-m := hello.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
```

Kernel Modules

Open the terminal

```
$ make
```

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

To check the kernel log for the "Hello, World!" message, you can use:

```
$dmesg |tail
```

To remove the module, use:

```
$ sudo rmmod hello
```

THANK YOU

S Thenmozhi

Department of Computer Applications

thenmozhis@pes.edu

+91 80 6666 3333 Extn 393