Linux kernel Module Development (1)

Target

1. Write a c/c++ program
2. To implement a Linux kernel module
3. GCC

Tools

Install GCC Software Collection

```
sudo apt-get install build-essential
```

How to use GCC

- gcc and make

An example of Linux Kernel Module

```
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/sched/task.h>
#include <linux/sched/signal.h>

/* init function  */
static int
hellokmodule_init(void)
{
    printk(KERN_ALERT "ALAL:simple module initialized\n");
    printk("ALAL:Jiffies value: %lu\n", jiffies);
    return 0;
}

/* exit function - logs that the module is being removed */
static void
hellokmodule_exit(void)
{
    printk(KERN_ALERT "ALAL:simple module is being unloaded\n");
    print("ALAL:Jiffies value: %lu\n", jiffies);
}

module_init(hellokmodule_init);
module_exit(hellokmodule_exit);

MODULE_LICENSE ("GPL");
MODULE_AUTHOR ("LKD");
MODULE_DESCRIPTION ("Simple Kernel Module");
MODULE_VERSION("1.01");
```

Compile the above Linux kernel module

1. 创建Makefile文件:

```
ModuleName=hellokmodule
obj-m +=${ModuleName}.o
all:${ModuleName}.ko
${ModuleName}.ko:${ModuleName}.c
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
testload:${ModuleName}.ko
    sudo dmesg -C
    sudo insmod ${ModuleName}.ko
    sudo dmesg | grep ALAL
testunload:${ModuleName}.ko
    sudo dmesg -C
    sudo rmmod ${ModuleName}.ko
    sudo dmesg | grep ALAL
```

2. 编译:

```
make
```

3. 执行（插入模块）：

```
make testload
```

3. 执行（卸载模块）：

```
make testunload
```

Target & how to do

write a c program to implement to list all processes using Linux kernel module mechanism

- 需要列出每个进程的名字(comm)、进程ID号(pid)、父进程ID号、进程状态、学号姓名等
- 统计共有多少个进程

```
   struct task_struct *p=&init_task;
   for_each_process(p)
    {
        printk("%s[%d]\n",p->comm, p->pid1);
    }
```