Laborator 2

Module Kernel și Debugging

Sisteme de Operare 2

4 martie 2015

Outline

Module kernel - .ko

- The cornerstone of ~~any nutritious breakfast~~ modern kernels
- built-in
- loadable

Outline

Structura unui modul

Headere

Informații modul

Entry point

Exit point

Specificare entry/exit points

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

MODULE_DESCRIPTION("My kernel module");
MODULE_AUTHOR("Me");
MODULE_LICENSE("GPL");

static int dummy_init(void)
{
        printk( KERN_DEBUG "Hi\n" );
        return 0;
}

static void dummy_exit(void)
{
        printk( KERN_DEBUG "Bye\n" );
}

module_init(dummy_init);
module_exit(dummy_exit);
```

Compilarea unui modul

Makefile

```
KDIR=/lib/modules/`uname -r`/build

kbuild:
        make -C $(KDIR) M=`pwd`

clean:
        make -C $(KDIR) M=`pwd` clean
```

Kbuild

```
EXTRA_CFLAGS=-g

obj-m        = modul.o
```

Lucrul cu module

listare module

```
# lsmod
```

inserare modul

```
# insmod nume_modul.ko
```

oprire modul

```
# rmmod nume_modul
```

informații modul

```
# modinfo nume_modul
```

Outline

- kernel oops/panic
- objdump
- addr2line
- netconsole
- printk, dyndbg
- KDB

Kernel Oops

```
BUG: unable to handle kernel paging request at 00001234
IP: [<c89d4005>] my_oops_init+0x5/0x20 [oops]
Oops: 0002 [#1] PREEMPT DEBUG_PAGEALLOC
last sysfs file: /sys/devices/virtual/net/lo/operstate
Modules linked in: oops(+) netconsole ide_cd_mod pcnet32

Pid: 4157, comm: insmod Not tainted (2.6.28.4 #2)
EIP: 0060:[<c89d4005>] EFLAGS: 00010246 CPU: 0
EIP is at my_oops_init+0x5/0x20 [oops]
EAX: 00000000 EBX: fffffffc ECX: c89d4300 EDX: 00000001
ESI: c89d4000 EDI: 00000000 EBP: c5799e24 ESP: c5799e24
 DS: 007b ES: 007b FS: 0000 GS: 0033 SS: 0068
Process insmod (pid: 4157, ti=c5799000 task=c665c780)
Stack:
 c5799f8c c010102d c72b51d8 0000000c c5799e58
 c89d4300 c5799e58 c724f448 00000001 c89d4300
Call Trace:
 [<c010102d>] ? _stext+0x2d/0x170
 [<c01708e4>] ? __vunmap+0xa4/0xf0
 [<c0170981>] ? vfree+0x21/0x30
```

Printk

- console_loglevel
- /proc/sys/kernel/printk

- KERN_EMERG - n = 0
- KERN_ALERT - n = 1
- KERN_CRIT - n = 2
- KERN_ERR - n = 3
- KERN_WARNING - n = 4
- KERN_NOTICE - n = 5
- KERN_INFO - n = 6
- KERN_DEBUG - n = 7

Dyndbg

- debugfs: /debug/dynamic_debug/control
- echo 'file sock.c line 16 +p' > /debug/dynamic_debug/control
- flags: p f l m t _

Filtre

- func
- file
- module
- format
- line

KDB

Live Kernel Debugger

- lsmod, ps, kill, dmesg, env, bt (backtrace)
- dump trace logs
- utilizare hardware breakpoints sau modificare memorie

Tracing

- Kprobes
  - adresă instrucțiune + handlere
  - înainte/după instrucțiune
- Jprobes
  - entry-point funcție
  - adresa simbol + handler
  - aceeași semnatură
  - jprobe_return()
- Kretprobes
  - return funcție
  - adresă simbol + handler

Outline

Keywords

- built-in module
- loadable module
- module_init
- module_exit
- Kbuild
- insmod,rmmod
- printk, dyndbg
- objdump
- addr2line
- netconsole
- KDB
- Kprobes
- Jprobes
- Kretprobes