

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Sistemas Operativos 1
Vacaciones 2020
Ing. Sergio Méndez
Aux. Brayan Ordoñez
Elmer Edgardo Alay Yupe
201212945

Creación de Módulos

RAM

creamos un archivo con nombre memo_carnet.c y colocamos el siguiente código

```
#include <linux/proc_fs.h>
#include <linux/seq_file.h>
#include <asm/uaccess.h>
#include <linux/hugetlb.h>
#include <linux/module.h>
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/fs.h>
```

En esta parte se incluyeron todas las librerías que se necesitan para ejecutar los comandos

```
#define BUFESIZE 150
MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("Escribir informacion de la memoria ram");
MODULE_AUTHOR("Elmer Edgardo Alay Yupe - 201212945");
```

Colocamos una descripción opcional del módulo

```
struct sysinfo inf;
```

Esta estructura es la que nos ayudará a obtener la memoria

```

static int escribir_archivo(struct seq_file * archivo, void *v){
    si_meminfo(&inf);
    seq_printf(archivo, " _____ \n");
    seq_printf(archivo, "|                |\n");
    seq_printf(archivo, "| 201212945      |\n");
    seq_printf(archivo, "| Elmer Edgardo Alay Yupe      |\n");
    seq_printf(archivo, "|_____|\n");
    seq_printf(archivo, "\n");
    seq_printf(archivo, " RAM Total: \t%li MB\n", (inf.totalram*4)/(1024));
    seq_printf(archivo, " RAM Libre: \t%li MB\n", (inf.freeram*4)/(1024));
    seq_printf(archivo, " % RAM en uso: \t%li %%\n", ((inf.totalram-inf.freeram)*100)/inf.totalram);

    return 0;
}

```

Este es el método que escribe un archivo con la información que le decimos, en la parte roja es donde obtenemos los datos de la memoria ram que nos interesa, la dividimos en 1024 para obtener su valor en MB

```

static int al_abrir(struct inode *inode, struct file *file){
    return single_open(file, escribir_archivo, NULL);
}

```

Este método escribirá el archivo cada vez que se le llame

```

static struct file_operations operaciones =
{
    .open = al_abrir,
    .read = seq_read
};

```

operaciones que se realizarán al abrir el archivo

```

static int iniciar(void){
    proc_create("memo_201212945", 0, NULL, &operaciones);
    printk(KERN_INFO "Carnet: 201212945\n");
}

```

```
return 0;  
}
```

Este método se ejecutará al crear el módulo e imprimirá un mensaje en consola

```
static void salir(void){  
    remove_proc_entry("memo_201212945", NULL);  
    printk(KERN_INFO "Curso: Sistemas Operativos 1\n");  
}
```

Este método se ejecutará al eliminar el módulo e imprimirá un mensaje en consola

```
module_init(iniciar);  
module_exit(salir);
```

Agregamos los 2 métodos anteriores a los métodos de crear y salir de los módulos

Ahora creamos un archivo de nombre Makefile y colocamos lo siguiente

```
bj-m += memo_201212945.o
```

Con este comando le damos un nombre al módulo

all:

```
make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
```

Con este comando creamos el módulo

clean:

```
make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

Eliminamos todos los archivos creados

Abrimos una terminal en la ubicación donde están los 2 archivos y ejecutamos el comando Make all

```
tutosea@tutosea-pc: ~/Documents/sopes1p1/modulos/memoria
File Edit View Search Terminal Help
tutosea@tutosea-pc:~/Documents/sopes1p1/modulos/memoria$ make all
make -C /lib/modules/4.15.0-101-generic/build M=/home/elmer/Documents/sopes1p1/modulos/memoria modules
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-101-generic'
  CC [M]  /home/elmer/Documents/sopes1p1/modulos/memoria/memo_201212945.o
/home/elmer/Documents/sopes1p1/modulos/memoria/memo_201212945.c: In function 'escribir_archivo':
/home/elmer/Documents/sopes1p1/modulos/memoria/memo_201212945.c:29:29: warning:
unknown conversion type character 'R' in format [-Wformat=]
    seq_printf(archivo, " % RAM en uso: \t%li %%\n",((inf.totalram-inf.freeram)
*100)/inf.totalram);
                                ^
Building modules, stage 2.
MODPOST 1 modules
  CC      /home/elmer/Documents/sopes1p1/modulos/memoria/memo_201212945.mod.o
  LD [M]  /home/elmer/Documents/sopes1p1/modulos/memoria/memo_201212945.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-101-generic'
tutosea@tutosea-pc:~/Documents/sopes1p1/modulos/memoria$
```

Ahora se han creado los siguientes archivos

```
tutosea@tutosea-pc: ~/Documents/sopes1p1/modulos/memoria
File Edit View Search Terminal Help
tutosea@tutosea-pc:~/Documents/sopes1p1/modulos/memoria$ ls
Makefile          memo_201212945.ko      memo_201212945.mod.o  modules.order
memo_201212945.c  memo_201212945.mod.c  memo_201212945.o      Module.symvers
tutosea@tutosea-pc:~/Documents/sopes1p1/modulos/memoria$
```

El que nos interesa es el archivo memo_201212945.ko, ahora ejecutamos el siguiente comando para insertar el módulo

```
tutosea@tutosea-pc: ~/Documents/sopes1p1/modulos/memoria
File Edit View Search Terminal Help
tutosea@tutosea-pc:~/Documents/sopes1p1/modulos/memoria$ ls
Makefile          memo_201212945.ko      memo_201212945.mod.o  modules.order
memo_201212945.c  memo_201212945.mod.c  memo_201212945.o      Module.symvers
tutosea@tutosea-pc:~/Documents/sopes1p1/modulos/memoria$ sudo insmod memo_201212945.ko
[sudo] password for tutosea:
tutosea@tutosea-pc:~/Documents/sopes1p1/modulos/memoria$
```

si colocamos el siguiente comando “dmesg” veremos la salida que colocamos en el archivo

```
tutosea@tutosea-pc: ~/Documents/sopes1p1/modulos/memoria
File Edit View Search Terminal Help
[24209.341037] audit: type=1400 audit(1592192161.135:46): apparmor="ALLOWED" operation="open" profile="libreoffice-soffice" name="/usr/share/drirc.d/00-mesa-defaults.conf" pid=18481 comm="soffice.bin" requested_mask="r" denied_mask="r" fsuid=1000 ouid=0
[24209.342496] audit: type=1400 audit(1592192161.135:47): apparmor="ALLOWED" operation="open" profile="libreoffice-soffice" name="/usr/share/drirc.d/00-mesa-defaults.conf" pid=18481 comm="soffice.bin" requested_mask="r" denied_mask="r" fsuid=1000 ouid=0
[24209.349503] audit: type=1400 audit(1592192161.143:48): apparmor="ALLOWED" operation="open" profile="libreoffice-soffice" name="/usr/share/drirc.d/00-mesa-defaults.conf" pid=18481 comm="soffice.bin" requested_mask="r" denied_mask="r" fsuid=1000 ouid=0
[24209.349938] audit: type=1400 audit(1592192161.143:49): apparmor="ALLOWED" operation="open" profile="libreoffice-soffice" name="/usr/share/drirc.d/00-mesa-defaults.conf" pid=18481 comm="soffice.bin" requested_mask="r" denied_mask="r" fsuid=1000 ouid=0
[24209.354166] audit: type=1400 audit(1592192161.147:50): apparmor="ALLOWED" operation="open" profile="libreoffice-soffice" name="/home/elmer/.cache/mesa_shader_cache/index" pid=18481 comm="soffice.bin" requested_mask="wrc" denied_mask="wrc" fsuid=1000 ouid=1000
[24209.355149] audit: type=1400 audit(1592192161.147:51): apparmor="ALLOWED" operation="open" profile="libreoffice-soffice" name="/usr/share/drirc.d/00-mesa-defaults.conf" pid=18481 comm="soffice.bin" requested_mask="r" denied_mask="r" fsuid=1000 ouid=0
[24605.138178] acpi INT3400:00: Unsupported event [0x86]
[25519.660487] memo_201212945: loading out-of-tree module taints kernel.
[25519.660533] memo_201212945: module verification failed: signature and/or required key missing - tainting kernel
[25519.660690] Carnet: 201212945
tutosea@tutosea-pc:~/Documents/sopes1p1/modulos/memoria$
```

Como podemos ver al final aparece nuestro mensaje. Ahora en otra terminal nos colocamos en la carpeta proc/ y colocamos el siguiente comando

```
tutosea@tutosea-pc: /proc
File Edit View Search Terminal Tabs Help
tutosea@tutosea-pc: ~/Documents/sopes1p1/modulos/memoria x tutosea@tutosea-pc: /proc
tutosea@tutosea-pc: /proc$ cat memo_201212945

201212945
Elmer Edgardo Alay Yupe

RAM Total:      15860 MB
RAM Libre:      6044 MB
tutosea@tutosea-pc: /proc$
```

Listo!!! para eliminar el módulo colocamos lo siguiente

```
tutosea@tutosea-pc: /proc
File Edit View Search Terminal Tabs Help
tutosea@tutosea-pc: ~/Documents/sopes1p1/modulos/memoria x tutosea@tutosea-pc: /proc
tutosea@tutosea-pc:/proc$ sudo rmmmod memo_201212945
[sudo] password for tutosea:
tutosea@tutosea-pc:/proc$
```

Volvemos a colocar “dmesg” y obtendremos lo siguiente

```
tutosea@tutosea-pc: /proc
File Edit View Search Terminal Tabs Help
tutosea@tutosea-pc: ~/Documents/sopes1p1/modulos/memoria x tutosea@tutosea-pc: /proc
[25654.142286] Call Trace:
[25654.142299] vsnprintf+0x66/0x510
[25654.142308] seq_vprintf+0x35/0x50
[25654.142314] seq_printf+0x4e/0x70
[25654.142322] escribir_archivo+0xeb/0xf0 [memo_201212945]
[25654.142328] seq_read+0xe5/0x430
[25654.142336] proc_reg_read+0x45/0x70
[25654.142342] __vfs_read+0x1b/0x40
[25654.142347] vfs_read+0x8e/0x130
[25654.142352] Sys_read+0x5c/0xe0
[25654.142360] do_syscall_64+0x73/0x130
[25654.142368] entry_SYSCALL_64_after_hwframe+0x3d/0xa2
[25654.142372] RIP: 0033:0x7fd5feef081
[25654.142375] RSP: 002b:00007ffcee58cbc8 EFLAGS: 00000246 ORIG_RAX: 0000000000000000
[25654.142379] RAX: ffffffffda RBX: 0000000000200000 RCX: 00007fd5feef081
[25654.142381] RDX: 0000000000200000 RSI: 00007fd603bd000 RDI: 0000000000000003
[25654.142383] RBP: 0000000000200000 R08: 00000000ffffff R09: 0000000000000000
[25654.142385] R10: 0000000000000022 R11: 0000000000000246 R12: 00007fd603bd000
[25654.142388] R13: 0000000000000003 R14: 00007fd603bd00f R15: 0000000000200000
[25654.142391] Code: fd ff ff 48 8d 42 02 b9 4c 00 00 00 48 89 45 e8 e9 cd fc ff ff 0f be f2 48 c7 c7 30 60 3a a
0 c6 05 86 76 c6 00 01 e8 4c 53 6e ff <0f> 0b 48 8b 45 e8 e9 ff fe ff ff 90 89 f0 c1 e0 08 c1 f8 08 89
[25654.142442] ---[ end trace 4e5d0057d48c106d ]---
[25703.846608] Curso: Sistemas Operativos 1
tutosea@tutosea-pc:/proc$
```

Ahora nos sale el otro mensaje.

CPU

Creamos un archivo cpu_201212945.c y colocamos lo siguiente

```
cpu_201212945.c
~/Documents/sopes1p1/modules/cpu

#include <linux/module.h>
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/sched/signal.h>
#include <linux/tty.h>
#include <linux/version.h>
#include <linux/kthread.h>
#include <linux/proc_fs.h>
#include <linux/seq_file.h>
#include <asm/uaccess.h>
#include <linux/hugetlb.h>
#include <linux/fs.h>

#define MAX_BUF_SIZE 1024

MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("Escribir informacion de los procesos del cpu");
MODULE_AUTHOR("Elmer Edgardo Alay Yupe - 201212945");

struct list_head *list;
struct task_struct *task_child;
struct task_struct *iter;

static int escribir_archivo(struct seq_file * archivo, void *v)
{
    seq_printf(archivo, "
    seq_printf(archivo, "
    seq_printf(archivo, " 201212945
    seq_printf(archivo, " Elmer Edgardo Alay Yupe
    seq_printf(archivo, "
    seq_printf(archivo, "\n");

    int cnt = 0;
    for_each_process(iter)
    {
        seq_printf(archivo, "Padre PID:%d - NOMBRE:%s - Estado:%ld\n", iter->pid, iter->comm, iter->state);

        list_for_each(list, &iter->children)
        {
            task_child = list_entry(list, struct task_struct, sibling);
            seq_printf(archivo, " -Hijo de %s[%d] PID:%d - NOMBRE:%s - Estado:%ld\n", iter->comm, iter->pid, task_child->pid, task_child->comm, task_child->state);
        }

        cnt++;
    }
}
```

```
cpu_201212945.c
~/Documents/sopes1p1/modules/cpu

int cnt = 0;
for_each_process(iter)
{
    seq_printf(archivo, "Padre PID:%d - NOMBRE:%s - Estado:%ld\n", iter->pid, iter->comm, iter->state);

    list_for_each(list, &iter->children)
    {
        task_child = list_entry(list, struct task_struct, sibling);
        seq_printf(archivo, " -Hijo de %s[%d] PID:%d - NOMBRE:%s - Estado:%ld\n", iter->comm, iter->pid, task_child->pid, task_child->comm, task_child->state);
    }

    cnt++;
}

seq_printf(archivo, "Número de procesos: %d\n", cnt);
return 0;

static int al_abrir(struct inode *inode, struct file *file){
    return single_open(file, escribir_archivo, NULL);
}

static struct file_operations operaciones =
{
    .open = al_abrir,
    .read = seq_read
};

static int iniciar(void)
{
    proc_create("cpu_201212945", 0, NULL, &operaciones);
    printk(KERN_INFO "Nombre: Elmer Edgardo Alay Yupe\n");
    return 0;
}

static void salir(void)
{
    remove_proc_entry("cpu_201212945", NULL);
    printk(KERN_INFO "Curso: Sistemas Operativos 1\n");
}

module_init(iniciar);
module_exit(salir);
```

y ahora creamos el archivo Makefile con lo siguiente


```
obj-m += cpu_201212945.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
```

Makefile ▾ Tab Width: 4 ▾ Ln 7, Col 64 ▾ INS

Ahora los pasos son los mismos para crear el módulo de la RAM, sólo seguimos los pasos