



# Práctica 1: Módulos

---

LIN - Curso 2014-2015



# Contenido

---



## 1 Introducción

## 2 Ejercicios

## 3 Práctica



# Práctica 1: Módulos



## Objetivos

- 1 Familiarizarse con las siguientes abstracciones de Linux:
  - Módulos
  - Sistema de ficheros /proc
  - Listas enlazadas en el kernel
  - Protección de memoria kernel/usuario
  - Gestión básica de memoria dinámica en el kernel
- 2 Afrontar las dificultades de la programación en espacio de kernel



# Contenido

---



**1** Introducción

**2** Ejercicios

**3** Práctica



# Ejercicios I



## Ejercicio 1

- `printk()` es un mecanismo de logging. 8 niveles de prioridad (`<linux/kernel.h>`)
  - ¿Qué diferencia encuentras entre `KERN_INFO` y `KERN_ALERT`?

## Ejercicio 2

- La función de carga de los módulos de ejemplo devuelve 0 ¿qué ocurre cuando se devuelve un número negativo?



# Ejercicios II



## Ejercicio 3

- Consultar uso de las macros `__init` y `__exit` `<linux/init.h>`
  - ¿Qué diferencias existen en el módulo generado al usar estas macros en la declaración de las funciones del módulo?

## Ejercicio 4

- En `<linux/module.h>` se describen los distintos tipos de licencias
  - ¿Qué diferencias existen en el módulo generado?

## Ejercicio 5

- Paso de parámetros `<linux/moduleparam.h>`
  - Ejemplo 2.7 de *Linux Kernel Module Programming Guide* (módulo `hello-5`)





## Ejercicio 6

- Estudiar la implementación del módulo 'clipboard', que exporta una entrada /proc
  - Al cargar/descargar el módulo se creará/eliminará una entrada **clipboard** en el sistema de ficheros virtual /proc
  - La entrada **clipboard** puede emplearse como un portapapeles (*clipboard*) del sistema



# Contenido

---



**1** Introducción

**2** Ejercicios

**3** Práctica







# Especificación de la práctica

- Crear un módulo modlist que gestione una lista enlazada de enteros

```
struct list_head mylist; /* Lista enlazada */  
  
/* Nodos de la lista */  
typedef struct {  
    int data;  
    struct list_head links;  
}list_item_t;
```

- El módulo permitirá al usuario insertar/eliminar elementos de la lista mediante la entrada `/proc/modlist`
  - Cuando el módulo se cargue/descargue se creará/eliminará dicha entrada
- La memoria asociada a los nodos de la lista debe gestionarse de forma dinámica empleando `vmalloc()` y `vfree()`
  - Al descargar el módulo → liberar memoria si lista no vacía



# Especificación de la práctica

## Operaciones soportadas por el módulo

### 1 Inserción al final de la lista

- `echo add 10 > /proc/modlist`

### 2 Eliminación de la lista

- `echo remove 10 > /proc/modlist`
- *Borra todas las ocurrencias de ese elemento en la lista*

### 3 Impresión por pantalla de la lista

- `cat /proc/modlist`

### 4 Borrado de todos los elementos de la lista

- `echo cleanup > /proc/modlist`

- Se aconseja utilizar `sscanf()` para procesar los comandos del usuario



# Ejemplo de ejecución

terminal

```
kernel@debian$ sudo insmod modlist.ko
kernel@debian$ cat /proc/modlist
kernel@debian$ echo add 10 > /proc/modlist
kernel@debian$ cat /proc/modlist
10
kernel@debian$ echo add 4 > /proc/modlist
kernel@debian$ echo add 4 > /proc/modlist
kernel@debian$ cat /proc/modlist
10
4
4
kernel@debian$ echo add 2 > /proc/modlist
kernel@debian$ echo add 5 > /proc/modlist
kernel@debian$ cat /proc/modlist
10
4
4
2
5
```





## Ejemplo de ejecución (cont..)

terminal

```
kernel@debian$ echo remove 4 > /proc/modlist
kernel@debian$ cat /proc/modlist
10
2
5
kernel@debian$ echo cleanup > /proc/modlist
kernel@debian$ cat /proc/modlist
kernel@debian$
```





# Partes opcionales

---

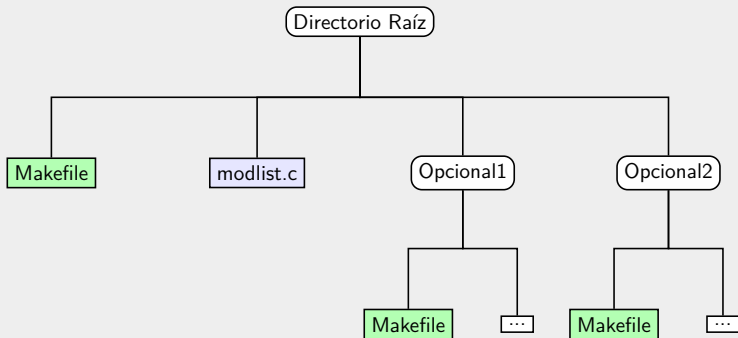
- 1 Implementar la operacion 'sort' de la lista
  - `echo sort > /proc/modlist`
  - La lista se ordenará en orden ascendente
- 2 Codificar un módulo donde la lista gestionada sea de cadenas de caracteres de longitud acotada (ej: máx 20 caracteres por nodo).



# Entrega de la práctica

- A través del Campus Virtual
  - Hasta el 22 de octubre
- Aconsejable mostrar el funcionamiento antes de hacer la entrega

## Estructura entrega (en un fichero comprimido .tar.gz o .zip)





## LIN - Práctica 1: Módulos Versión 0.1

©J.C. Sáez, M. Prieto

*This work is licensed under the Creative Commons **Attribution-Share Alike 3.0 Spain License**. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/es/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.*

*Esta obra está bajo una licencia **Reconocimiento-Compartir Bajo La Misma Licencia 3.0 España de Creative Commons**. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-sa/3.0/es/> o envíe una carta a Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.*

Este documento (o uno muy similar) está disponible en <https://cv4.ucm.es/moodle/course/view.php?id=49276>

