

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Laboratorio de Sistemas Operativos I
Auxiliar de cátedra: Bryan Otoniel Ordoñez Morales

DOCUMENTACIÓN
Módulos de Kernel

Luis Javier Yela	201314891
Danilo Urías Coc	201314565

Contenido

Conocimientos previos.....	1
Kernel de Linux.....	1
Módulos de Kernel.....	2
Proceso de creación del módulo de memoria	2
Archivos necesarios.....	2
Implementación del Módulo de memoria.....	3
Implementación del Módulo CPU	5

Conocimientos previos

Kernel de Linux

El kernel de Linux es el componente principal de un sistema operativo de Linux y es la interfaz principal entre el hardware de una computadora y sus procesos. Se comunica entre ambos, gestionando los recursos de la manera más eficiente posible.

El núcleo se llama así porque, existe dentro del sistema operativo y controla todas las funciones principales del hardware, ya sea un teléfono, computadora portátil, servidor o cualquier otro tipo de computadora.

El Kernel tiene 4 trabajos:

- **Gestión de la memoria:** Realiza un seguimiento de la cantidad de memoria utilizada para almacenar datos.
- **Gestión de procesos:** Determina cuales procesos pueden usar la unidad central de procesamiento (CPU), cuándo y durante cuánto tiempo.
- **Controladores de dispositivo:** Actúa como mediador / intérprete entre el hardware y los procesos.
- **Llamadas del sistema y seguridad:** Recibe solicitudes de servicio de los procesos.

El kernel trabaja con una analogía similar a la de un asistente personal ocupado para un ejecutivo poderoso (el hardware). El trabajo del asistente es transmitir mensajes y solicitudes (procesos) de los empleados y el público (usuarios) al ejecutivo, recordar qué se almacena dónde (memoria) y determinar quién tiene acceso al ejecutivo en un momento dado y cómo largo.

Módulos de Kernel

Los módulos son piezas de código que se pueden cargar y descargar en el núcleo en el momento que sea necesario. Extienden la funcionalidad del núcleo sin la necesidad de reiniciar el sistema. Por ejemplo, un tipo de módulo es el controlador del dispositivo, que permite que el núcleo acceda al hardware conectado al sistema. Sin módulos, Se tendría que construir núcleos monolíticos y agregar nuevas funcionalidades directamente a la imagen del núcleo. Además de provocar núcleos más grandes, esto tiene la desventaja de requerir reconstruir y reiniciar el núcleo cada vez que se necesita una nueva funcionalidad.

Proceso de creación del módulo de memoria

Archivos necesarios

- **Archivo Makefile:** Los archivos makefile facilitan la escritura de conjuntos complejo de reglas sobre cómo compilar un programa y luego simplemente compilarlo escribiendo make en la línea de comandos.

Ejemplo de un make file que contiene el archivo.o resultante al compilar el código y los parámetros para las opciones all y clean.

```
obj-m += cpu_201314565_201314891.o
KDIR ?= /lib/modules/$(shell uname -r)/build

all:
    make -C $(KDIR) M=$(PWD) modules

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

- **Archivo de implementación:** Para implementar el módulo es necesario escribir todas las instrucciones de su implementación en lenguaje C, en este archivo se especifica todo lo necesario para la creación, carga y descarga del módulo.

Implementación del Módulo de memoria

El primer módulo solicitado en el proyecto debía tener las acciones necesarias para poder mostrar el estado de la memoria RAM, además de mostrar un mensaje al ser cargado y otro mensaje al ser descargado.

Paso 1: Se escribían los archivos makefile y .c con sus instrucciones correspondientes para la creación, carga y descarga del módulo.

Para poder acceder a la información de la memoria fue utilizado el encabezado *linux/kernel.h* para poder acceder a los valores *totalram* y *freeram* del struct *sysinfo*. Estos valores poseían la información necesaria de memoria total del equipo y la memoria libre.

Paso 2: Ingresar como super usuario en una terminal.

```
sudo su
```

Paso 3: Navegar hasta el directorio en donde se encuentra el archivo makefile y .c, al estar dentro de la carpeta ejecutar el siguiente comando para compilar el archivo .c.

```
make all
```

Paso 4: Si se consiguió una compilación exitosa en el paso anterior se debe ingresar el siguiente comando que insertará el módulo.

```
sudo insmod memo_201314565_201314891.ko
```

Paso 6: En el método “al_cargar” del archivo .c se especificaban datos que debían mostrarse luego de cargar el módulo. Con el siguiente comando se debe visualizar el mensaje.

```
dmesg
```

Paso 7: El módulo al ejecutarse debía crear un archivo de texto con la información de la memoria y la información de los estudiantes. Para poder visualizar el contenido del archivo se debe navegar al directorio /proc/

```
cd /proc/
```

Paso 8: Desde este directorio es posible mostrar el contenido del archivo utilizando el comando cat.

```
cat memo_201314565_201314891
```

Otros

Descargar el módulo: Para descargar el módulo y dejarlo totalmente inhabilitado se puede utilizar el siguiente comando.

```
rmmod memo_201314565_201314891
```

Implementación del Módulo CPU

La finalidad del segundo modulo era listar los procesos que con sus atributos como el PID, nombre, estado y sus procesos hijos.

Para la implementación de este módulo fue necesaria la utilización de *task_struct*, que permite iterar todos los structs de los procesos incluyendo los hijos de cada proceso, pudiendo acceder a la información solicitada.

Paso 1: Se escribían los archivos makefile y .c con sus instrucciones correspondientes para la creación, carga y descarga del módulo.

Paso 2: Ingresar como super usuario en una terminal.

```
sudo su
```

Paso 3: Navegar hasta el directorio en donde se encuentra el archivo makefile y .c, al estar dentro de la carpeta ejecutar el siguiente comando para compilar el archivo .c.

```
make all
```

Paso 4: Si se consiguió una compilación exitosa en el paso anterior se debe ingresar el siguiente comando que insertará el módulo.

```
sudo insmod cpu_201314565_201314891.ko
```

Paso 6: En el método “al_cargar” del archivo .c se especificaban datos que debían mostrarse luego de cargar el módulo. Con el siguiente comando se debe visualizar el mensaje.

```
dmesg
```

Paso 7: El módulo al ejecutarse debía crear un archivo de texto con la información de los procesos y la identificación de los estudiantes. Para poder visualizar el contenido del archivo se debe navegar al directorio /proc/

```
cd /proc/
```

Paso 8: Desde este directorio es posible mostrar el contenido del archivo utilizando el comando cat.

```
cat cpu_201314565_201314891
```

Otros

Descargar el módulo: Para descargar el módulo y dejarlo totalmente inhabilitado se puede utilizar el siguiente comando.

```
rmmod cpu_201314565_201314891
```