APUNTES DSO

The kernel uses the major number at open time to dispatch execution to the appropriate driver.

The minor number is used only by the driver specified by the major number; other parts of the kernel don't use it, and merely pass it along to the driver. It is common for a driver to control several devices (as shown in the listing); the minor number provides a way for the driver to differentiate among them.

Version 2.4 of the kernel, though, introduced a new (optional) feature, the device file system or devfs. If this file system is used, management of device files is simplified and quite different; on the other hand, the new filesystem brings several user-visible incompatibilities, and as we are writing it has not yet been chosen as a default feature by system distributors. The previous description and the following instructions about adding a new driver and special file assume that devfs is not present. The gap is filled later in this chapter, in Section 3.10.

strace es una utilidad de línea de órdenes para comprobación de errores en el sistema operativo GNU/Linux. Permite vigilar las llamadas al sistema usadas por un determinado programa y todas las señales que éste recibe.1 Su funcionamiento es posible por una característica del núcleo linux llamada ptrace.

Uso de la herramienta **strace** para monitorizar las llamadas al sistema de un proceso.

dmesg (diagnostic message, mensajes de diagnóstico) es una instrucción presente en los sistemas operativos Unix que lista el buffer de mensajes del núcleo. Este búfer contiene una gran variedad de mensajes importantes generados durante el arranque del sistema, la detección del hardware, asignación de controladores (drivers) y durante la depuración de aplicaciones. La información ofrecida por dmesg puede guardarse en el disco duro mediante un demonio de registro, como syslog.

Con especial atención a setvfub() y las opciones de buffering (no buffer: _IONFB, line buffer; _IOLBF; y full buffer: _IOFBF).

Uso de **fseek()** para acceder de forma no secuencial a diferentes posiciones de un fichero.

Diferentes formas de acceder a la syscall write del kernel (printf/fprintf, write/fwrite, syscall(4,...), incluir código ensamblador para saltar a la rutina del kernel realizando el cambio de modo de ejecución).

The **fseek()** function sets the file position indicator for the stream pointed to by stream. The new position, measured in bytes, is obtained by adding offset bytes to the position specified by whence. If whence is set to SEEK_SET, SEEK_CUR, or SEEK_END, the offset is relative to the start of the file, the current position indicator, or end-of-file, respectively. A successful call to the fseek() function clears the end-of-file indicator for the stream and undoes any effects of the ungetc(3) function on the same stream.

What is **syscall** in Linux?

As the name suggests, syscalls are system calls, and they're the way that you can make requests from user space into the Linux kernel. The kernel does some work for you, like creating a process, then hands control back to user space.

En informática, un **módulo de kernel cargable (LKM)** es un archivo de objeto que contiene código para extender el kernel en ejecución, o el llamado kernel base, de un sistema operativo. Los LKM se utilizan normalmente para agregar soporte para nuevo hardware (como controladores de dispositivo) y / o sistemas de archivos, o para agregar llamadas al sistema.

Cuando la funcionalidad proporcionada por un LKM ya no es necesaria, se puede descargar para liberar memoria y otros recursos.

El sistema de archivos /**proc** contiene un sistema de archivos imaginario o virtual. Este no existe físicamente en disco, sino que el núcleo lo crea en memoria. Se utiliza para ofrecer información relacionada con el sistema (originalmente acerca de procesos, de aquí su nombre).

En sistemas operativos Unix-like, un **loop device**, vnd (vnode disk) o lofi (loop file interface) es un dispositivo virtual que hace que se pueda acceder a un fichero como un dispositivo de bloques.

En el contexto del desarrollo de software, **Make** es una herramienta de gestión de dependencias, típicamente, las que existen entre los archivos que componen el código fuente de un programa, para dirigir su recompilación o "generación" automáticamente. Si bien es cierto que su función básica consiste en determinar automáticamente qué partes de un programa requieren ser recompiladas y ejecutar los comandos necesarios para hacerlo, también lo es que Make puede usarse en cualquier escenario en el que se requiera, de alguna forma, actualizar automáticamente un conjunto de archivos a partir de otro, cada vez que este cambie.

Preguntas Test:

¿Con qué funciones se puede acceder a variables sencillas del espacio de usuario desde el kernel? Sol: get_user() y put_user().

Las llamadas al sistema (syscall) son: El interfaz para acceder a los servicios del kernel del SO.

Cuando queremos acceder desde el kernel a un buffer en el área de memoria de usuario: usaremos las funciones coy_from_user() y copy_to_user().

Cuando compilamos un LKM con Makefile, obtenemos un arhivo: .ko.

Al abrir un LKM debemos implementar: una función de inicialización y otra para finalización o descarga del modulo.

¿Qué commandos de linux hemos usado para cargar y descargar los módulos? Sol: insmod y rmmod.

Si se invoca a una llamada al Sistema que no está implementada, ¿qué valor retorna? Sol: -38.

La herramienta/comando make sirver para: ejecutar las reglas definidas en el fichero Makefile.

Tras hacer make, el fichero con la imagen del kernel se debe instalar en el directorio de la raspberry: Sol: /boot.