

Lab 4

1. ¿Qué puede ver en el output cuando realiza estas acciones?
Cuando no se está haciendo ninguna acción el ID del proceso y el tipo de proceso no cambian, cuando movemos el mouse se ejecuta el proceso swapper con el proceso 0, y en algunas ocasiones tiene ID de 1523
2. ¿Para qué sirve SystemTap?
SystemTap sirve para extraer, filtrar y resumir datos para permitir el diagnóstico de problemas complejos de rendimiento o funcionales dentro del sistema linux. Con esta herramienta se evita tener que compilar, instalar y reiniciar que a veces sirve para recopilar información.
3. ¿Qué es una probe?
Es un set de herramientas para debuggear el kernel de un linux, también sirve para encontrar cuellos de botella en el performance del sistema. Con la información recolectada se puede utilizar para mejorar el sistema.
4. ¿Cómo funciona SystemTap?
Funciona traduciendo el archivo a C, compilando el archivo este crea un módulo en el kernel con la lógica del archivo. Cuando el módulo se carga este activa los eventos escritos conectándose con el kernel. A medida que los eventos van sucediendo se ejecutan los archivos compilados.
5. ¿Qué es hacer profiling y qué tipo de profiling se hace en este ejercicio?
Es la investigación de comportamiento de un programa de computador usando información reunida desde un análisis dinámico por parte del mismo. Basado en eventos.

Ejercicio 2

1. ¿Cuál es la diferencia en C entre un método que no recibe parámetros y uno que recibe void?
No hay diferencia, simplemente colocamos void como indicación que no recibe parámetros.
2. ¿Qué diferencia hay entre printk y printf?
printk es una función propia del kernel para debuggear, puede ser llamado en cualquier lugar del kernel. Mientras que printf se ejecuta en un archivo STD OUT.
3. ¿Qué es y para qué sirve KERN_INFO?
Es un nivel de log, indica que tan importante es el mensaje. El kernel decide donde debe ser desplegado dependiendo del nivel lo desplegará a una consola.
4. ¿Qué es una goal definition o definición de meta en un Makefile, y qué se está haciendo con la definición de meta obj-m?
Es para compilar cualquier programa compilable, cuando se ejecuta el programa se utiliza la versión más reciente y si hubieron cambios no es necesario volver a compilar.
meta obj-m es una instrucción para crear un módulo en el kernel con extensión .ko, después de crear los .o y .c del programa.
5. ¿Qué función tienen las líneas all: y clean:?

All: Ejecuta todos los sub procesos o comandos para poder hacer el build de lo que se requiere.

Clean: Como su nombre indica limpia, remueve todos los archivos creados incorrectamente por el compilador automáticamente. (No siempre sucede pero es para prevenir)

6. ¿Qué hace la opción -C en este Makefile?

Indica al compilador crear un archivo .o

7. ¿Qué hace la opción M en este Makefile?

Hace que el makefile regrese al directorio de origen del módulo antes de crear el destino de los módulos.

8. ¿Para qué sirve dmesg?

Hace un print del buffer del kernel. El output son drivers del sistema.

9. ¿Qué hace la función simple_init en su programa simple.c?

Carga el módulo

```
[ 13.168269] fuse init (API version 7.16)
[ 14.363469] Bluetooth: Core ver 2.16
[ 14.363517] NET: Registered protocol family 31
[ 14.363520] Bluetooth: HCI device and connection manager initialized
[ 14.363524] Bluetooth: HCI socket layer initialized
[ 14.373028] e1000: eth1 NIC Link is Up 1000 Mbps Full Duplex, Flow Control
X
[ 14.374616] ADDRCONF(NETDEV_UP): eth1: link is not ready
[ 14.374717] ADDRCONF(NETDEV_CHANGE): eth1: link becomes ready
[ 14.652140] lp: driver loaded but no devices found
[ 14.661880] ppdev: user-space parallel port driver
[ 24.448101] eth1: no IPv6 routers present
[ 84.919970] Clocksource tsc unstable (delta = 94976416 ns)
[ 84.920489] Switching to clocksource acpi_pm
[ 6878.049249] hrtimer: interrupt took 31495343 ns
[ 7703.499831] Loading ModuleOlaBiejaMeada
os@debian:~/Desktop/Lab4$ S
```

10. ¿Qué hace la función simple_exit en su programa simple.c?

Remueve el módulo o da por terminado la ejecución.

11. Usted ha logrado crear, cargar y descargar un módulo de Linux. ¿Qué poder otorga el ejecutar código de esta forma?

Puede extender la funcionalidad, normalmente son fragmentos de código que pueden ser creados y eliminados en el núcleo bajo demanda. Performance

```
[ 14.374616] ADDRCONF(NETDEV_UP): eth1: link is not ready
[ 14.374717] ADDRCONF(NETDEV_CHANGE): eth1: link becomes ready
[ 14.652140] lp: driver loaded but no devices found
[ 14.661880] ppdev: user-space parallel port driver
[ 24.448101] eth1: no IPv6 routers present
[ 84.919970] Clocksource tsc unstable (delta = 94976416 ns)
[ 84.920489] Switching to clocksource acpi_pm
[ 6878.049249] hrtimer: interrupt took 31495343 ns
[ 7703.499831] Loading ModuleOlaBiejaMeada
[ 7834.786499] Loading ModuleAdioBiejaMeada
os@debian:~/Desktop/Lab4$ S
```

Ejercicio 3

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
proc /proc proc defaults 0 0
# / was on /dev/sda1 during installation
UUID=c6c1c4e3-4e42-4b7a-89e2-69140f21f585 / ext3 errors=remoun$
# swap was on /dev/sda5 during installation
UUID=e2976f5e-f45d-4d31-8a8f-ac19d4f04620 none swap sw $
/dev/scd0 /media/cdrom0 udf,iso9660 user,noauto 0 0
```

1. ¿Qué es y para qué sirve el archivo fstab?
Es una tabla de configuración diseñada para facilitar la carga de montar y desmontar sistemas de archivos de una máquina.
2. ¿Qué almacena el directorio /etc? ¿En Windows, quién (hasta cierto punto) funge como /etc?
Almacena todos los archivos de configuración del sistema.
System32
3. ¿Qué se almacena en /dev y en /dev/disk?
/dev: Es donde se almacenan archivos o directorios especiales, y es donde vemos que el sistema de linux se basa en eso, archivos y directorios.

/dev/disk: organiza los diferentes discos conocidos y conectados a la computadora, sean externos o internos con nombres más user-friendly.
4. ¿Por qué se usa en lugar de sólo /dev/sda, y cuál es el papel que el programa udev cumple en todo esto?
Porque dev/sda es el disco principal de almacenamiento
Su función es controlar los ficheros de dispositivo en /dev
5. ¿Qué es un block device y qué significado tiene sdxN, donde x es una letra y N es un número, en direcciones como /dev/sdb?
Block device es un dispositivo de almacenamiento que se encarga de mover datos en secuencias de bytes o bits.
6. Investigue y explique los conceptos de Master Boot Record (MBR) y Volume Boot Record (VBR), y su relación con UEFI.

MBR: es el primer lugar físico de un portador de datos, este sirve para iniciar las computadoras

VBR: es el primer sector de un dispositivo de almacenamiento que no está particionado o bien el primer sector particionado.

UEFI: es la interfaz en la que se opera automáticamente a partir de entornos operativos previamente iniciados del sistema operativo, por lo que la relación con los otros dos conceptos es que previamente se ejecuta MBR o VBR para luego dar inicio al UEFI.

7. ¿Qué es hacer chain loading?

Es reemplazar el programa actual que se está ejecutando por uno nuevo, esto es posible a través de un área común de data para pasar información del viejo al nuevo.

8. ¿Qué se está indicando con la configuración root=""?

Es el ID del disco de memoria, con este identificador podemos cambiar el orden del disco en la tarjeta madre sin afectar el punto de inicio que tendrán.

9. ¿Qué es vmlinuz?

Es la imagen comprimida del kernel, esta es descomprimida, cargada en la memoria y se ejecuta al bootear.

```
ps@debian:/boot$ sudo dpkg-reconfigure linux-image-2.6.32-5-686
Running depmod.
Running update-initramfs.
update-initramfs: Generating /boot/initrd.img-2.6.32-5-686
Examining /etc/kernel/postinst.d.
run-parts: executing /etc/kernel/postinst.d/dkms 2.6.32-5-686 /boot/vmlinuz-2.6.32-5-686
dkms: running auto installation service for kernel 2.6.32-5-686:
    virtualbox-ose-guest (3.2.10)...done.
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 2.6.32-5-686 /boot/vmlinuz-2.6.32-5-686
run-parts: executing /etc/kernel/postinst.d/pm-utils 2.6.32-5-686 /boot/vmlinuz-2.6.32-5-686
run-parts: executing /etc/kernel/postinst.d/update-notifier 2.6.32-5-686 /boot/vmlinuz-2.6.32-5-686
run-parts: executing /etc/kernel/postinst.d/zz-lilo 2.6.32-5-686 /boot/vmlinuz-2.6.32-5-686
/proc/misc: No entry for device-mapper found
Is device-mapper driver missing from kernel?
Added Linux *
Skipping /boot/vmlinuz.old
ps@debian:/boot$
```

10. Mencione tres diferencias funcionales entre GRUB y LILO

Lilo solo soporta un unico sistema operativo mientras que GRUB soporta varios

Lilo solo soporta Linux, mientras que GRUB soporta windows, unix, macOs, BSD, Solaris

Lilo es un boot loader fácil y simple de usar, en cambio GRUB es considerado difícil y complejo.

```
LILO 22.8 Loading Linux._
```

```
GRUB loading.  
Welcome to GRUB!  
error: file not found.  
Entering rescue mode...  
grub rescue> _
```