Oliver Graf

Kristen Brandt

Lab 4

Ejercicio 1:

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

sudo stap profiler.stp:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application, table

Description automatically generated

* ¿Qué puede ver en el *output* cuando realiza estas acciones?

Lo que se puede ver en el output es que cuando uno realiza algún tipo de movimiento o click el ID del proceso es mucho mas alto.

* ¿Para qué sirve SystemTap?

SystemTap proporciona una infraestructura de software libre (GPL) para simplificar la recopilación de información sobre el sistema Linux en ejecución, lo que ayuda a diagnosticar un problema funcional o de rendimiento. SystemTap elimina la necesidad de que el desarrollador tenga que pasar por el tedioso y perturbador instrumento, recompilar, instalar y reiniciar la secuencia que de otro modo podría ser necesaria para recopilar datos.

* ¿Qué es una *probe*?

Un probe del sistema operativo es para supervisar los recursos del sistema, el uso de la CPU, la actividad del disco, el uso de la memoria física y el tráfico de la red. Supervisa la cantidad de tráfico de red en el servidor. El tipo de probe más simple es rastrear un evento.

* ¿Cómo funciona SystemTap?

SystemTap proporciona una interfaz de línea de comandos simple y un lenguaje de scripting para escribir instrumentación para un kernel en ejecución más aplicaciones de espacio de usuario. Estamos publicando muestras, así como ampliando la biblioteca de scripts "tapset" interna para ayudar a la reutilización y abstracción.

* ¿Qué es hacer *profiling* y qué tipo de *profiling* se hace en este ejercicio?

Hacer profiling es una forma de análisis dinámico de programas que mide el espacio (memoria) o la complejidad temporal de un programa, el uso de instrucciones particulares o la frecuencia y duración de las llamadas a funciones. El tipo de profiling que se hace en este ejercicio es

timer.profile es un temporizador que se activa periódicamente en cada CPU.

Ejercicio 2:

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

* ¿Cuál es la diferencia en C entre un método que no recibe parámetros y uno que recibe void?

La diferencia entre un método que no recibe parámetros y uno que recibe void es que el método que no toma ningún argumento. El método que no recibe parámetros es un método que puede tomar cualquier cosa por sus argumentos.

* ¿Qué diferencia hay entre printk y printf?

Printf: Escribe la cadena C apuntada por formato en la salida estándar (stdout). Si el formato incluye especificadores de formato (subsecuencias que comienzan con%), los argumentos adicionales que siguen al formato se formatean y se insertan en la cadena resultante reemplazando sus respectivos especificadores.

Printk: Función C de la interfaz del kernel de Linux que imprime mensajes en el registro del kernel. Printk acepta un parámetro de cadena llamado cadena de formato, que especifica un método para convertir un número arbitrario de parámetros de tipo de datos variados en una cadena, que luego se imprime en el registro del kernel.

* ¿Qué es y para qué sirve KERN\_INFO?

KERN\_INFO es el nivel de registro utilizado para mensajes informativos sobre la acción realizada por el kernel. Este sirve al ser utilizado dentro de un printk(KERN\_INFO "mensaje").

Graphical user interface, text, application

Description automatically generated

• ¿Qué es una ***goal definition*** o definición de meta en un *Makefile,* y qué se está haciendo con la definición de meta obj-m?

Las goal definition de meta en un makefile hacen que, en última instancia, conviene esforzarse por actualizar. De forma predeterminada, el objetivo es el primer objetivo en el archivo MAKE (sin contar los objetivos que comienzan con un punto). Por lo tanto, los archivos MAKE generalmente se escriben de manera que el primer objetivo sea compilar todo el programa o los programas que describen. Con la definición de meta obj-m se especifican archivos objeto que se crean como cargables módulos del kernel. Un módulo puede construirse a partir de un archivo fuente o de varias fuentes y/o archivos. En el caso de un archivo fuente, kbuild makefile

simplemente agrega el archivo a obj-m.

• ¿Qué función tienen las líneas all: y clean:?

Cuando make se ejecuta sin argumentos, el primer objetivo encontrado será construido. En el nivel superior Makefile el primer objetivo presente es all:.

Clean:, es una regla opcional. Esta permite escribir en la línea de comando para deshacerse de su objeto y archivos ejecutables. A veces, el compilador vinculará o compilará archivos incorrectamente y la única forma de comenzar de nuevo es eliminar todos los archivos de objetos y ejecutables.

• ¿Qué hace la opción –C en este *Makefile*?

La opción -c en Makefile es cambiar al directorio dir antes de leer los archivos MAKE. Si se especifican varias opciones '-C', cada una se interpreta en relación con la anterior '-C '.

• ¿Qué hace la opción M en este *Makefile*?

La opción M en el Makefile es para especificar en que dirección está presente el módulo.

Graphical user interface, text, application, email

Description automatically generated

sudo insmod simple.ko, demsg:

Text

Description automatically generated

* ¿Para qué sirve dmesg?

La función dmesg sirve para examinar o controlar el kernel ring buffer. Esta función ayuda a imprimir los mensajes de bootup de la máquina/módulo.

* ¿Qué hace la función simple\_init en su programa simple.c?

Esta función inicializa nuestro módulo e imprime el proceso.

sudo rmmod simple, demsg:

Graphical user interface, text, application

Description automatically generated

A screenshot of a cell phone

Description automatically generated with medium confidence

* ¿Qué hace la función simple\_exit en su programa simple.c?

La función finaliza nuestro módulo y lo desinstala.

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

Trabajar con módulos permite reutilizar código para evitar la redundancia en nuestros programas. Además, programar de manera modular permite tener código mucho más limpio y elegante, porque abstrae funciones que tenga que cumplir el programa.

Ejercicio 3:

Graphical user interface, application

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

* ¿Qué es y para qué sirve el archivo fstab?

El archivo fstab contiene descripciones de los filesystems que uno (la computadora) monta frecuentemente. Este archivo sirve para montar filesystems automáticamente cuande se hace un system boot.

* ¿Qué almacena el directorio /etc? ¿En Windows, quién (hasta cierto punto) funge como /etc?

Es una carpeta que contiene todos los archivos de configuración del sistema. En Windows hay un archivo llamado hosts que se encuentra dentro de la carpeta etc. Este archivo funge como /etc.

* ¿Qué se almacena en /dev y en /dev/disk?

En el directorio /dev están todos los archivos de dispositivo especiales para todos los dispositivos. En /dev/disk se almacena información adicional sobre las particiones del sistema.

Graphical user interface, text

Description automatically generated

* ¿Por qué se usa <la dirección completa del link hacia sda> en lugar de sólo /dev/sda, y cuál es el papel que el programa udev cumple en todo esto?

Se utiliza el link completo porque debe poder ser accedido desde cualquier directorio y es una dirección específica. Udev se encarga de proporcionar eventos de dispositivo, detectando los dispositivos que están conectados a la computadora.

* ¿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? Investigue y explique los conceptos de *Master Boot Record* (MBR) y *Volume Boot Rercord* (VBR), y su relación con UEFI.

Un block device es un archivo que hace referencia a un dispositivo. Estos archivos se distinguen de los archivos de caracteres porque proporcionan acceso al dispositivo sin mostrar las características del hardware del dispositivo. El MBR es la primera partición de la memoria secundaria que le dice a la computadora cómo cargar el sistema operativo. Mientras que el VBR está ubicado al principio de una partición de la memoria secundaria y ayuda en el boot similarmente a cómo funciona el MBR.

* ¿Qué es hacer *chain loading*?

Es reemplazar un programa que se está ejecutando con otro nuevo, utilizando un espacio en la memoria compartido para trasladar datos del programa actual al programa nuevo.

* ¿Qué se está indicando con la configuración root=”<el file system anotado>”?

Es para especificar cual sección del disco utilizar para cargar el kernel/SO.

Text

Description automatically generated

Graphical user interface

Description automatically generated with low confidence

Text

Description automatically generated

* ¿Qué es vmlinuz?

Es el ejecutable del kernel de Linux. Éste carga el sistema operativo en la memoria principal. Vmlinuz realiza una rutina que configura el hardware al hacer el boot del sistema.

Text

Description automatically generated

Text

Description automatically generated

Con LILO:

Graphical user interface, text

Description automatically generated

* Mencione tres diferencias funcionales entre GRUB y LILO.
  + GRUB puede utilizarse con varios sistemas operativos, mientras que LILO solo soporta Linux.
  + LILO es más fácil y sencillo de utilizar que GRUB.
  + GRUB ofrece una interfaz gráfica mientras que LILO no.