



Universidad Nacional Autónoma de México
Facultad de Ingeniería



Asignatura: Sistemas Operativos
Profesor: Wolf Iszaevich Gunnar Eyal
Grupo: 06

Nombre del Alumno: Padilla Herrera Carlos Ignacio
Fecha de entrega: 31/08/2017
<Práctica 2: Depuración por trazas>

Introducción

Strace es usado para ver que llamadas al sistema son usadas principalmente

Comando de Linux ls

Lista el contenido de los directorios (por defecto ordena la salida alfabéticamente)

Llamadas al sistema

Read	Lee bytes del descriptor de archivos (FD, File descriptor) (Archivo, socket)
Write	Escribe bytes de un FD
Open	Abre un archivo (regresa un FD)
Close	Cierra un FD
Fork	Crea un nuevo proceso (el proceso actual está bifurcado)
Exec	Ejecuta un nuevo programa
Connect	Conecta a un nuevo huésped
Accept	Acepta una nueva conexión de red
Stat	Lee archivos estadísticos
lctl	Configura las propiedades E/S, o las funciones miscelaneas
Mmap	Maapea un archivo al espacio de dirección del proceso de memoria
Brk	Extiende el puntero de cabeza (head pointer)

Desarrollo

Sistema Operativo

Usando el simulador de Linux, me dio el siguiente output al introducir en la línea de comandos strace ls -l:

<https://bellard.org/jslinux/vm.html?url=https://bellard.org/jslinux/buildroot-x86.cfg>

Elaborado por: Padilla Herrera Carlos Ignacio

Se utilizó la terminal y el commando strace

Resultados

```
[root@localhost ~]# strace ls -l
execve("/bin/ls", ["ls", "-l"], [/* 10 vars */) = 0
readlinkat(AT_FDCWD, "/proc/self/exe", "/bin/busybox", 4096) = 12
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_UNINITIALIZED, -1, 0) = 0xb7793000
open("/lib/libc.so.1", O_RDONLY) = 3
fstat(3, {st_mode=S_IFREG|0755, st_size=349804, ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_UNINITIALIZED, -1, 0) = 0xb7792000
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0 \233\0\0004\0\0\0"...
, 4096) = 4096
mmap2(NULL, 368640, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7738000
mmap2(0xb7738000, 346240, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED, 3, 0) = 0xb7738000
mmap2(0xb778d000, 4790, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED, 3, 0x54000) = 0xb778d000
mmap2(0xb778f000, 9792, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0xb778f000
close(3) = 0
munmap(0xb7792000, 4096) = 0
stat("/lib/ld-uClibc.so.0", {st_mode=S_IFREG|0755, st_size=28696, ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_UNINITIALIZED, -1, 0) = 0xb7792000
set_thread_area({entry_number:-1, base_addr:0xb7792690, limit:1048575, seg_32bit:1, contents:0, read_exec_only:0, limit_in_pages:1, seg_not_present:0, useable:1}) = 0 (entry_number:6)
mprotect(0xb778d000, 4096, PROT_READ) = 0
mprotect(0xb779e000, 4096, PROT_READ) = 0
ioctl(0, TCGETS, {B38400 opost isig icanon echo ...}) = 0
ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0
getuid32() = 0
time([1504219221]) = 1504219221
ioctl(0, TIOCGWINSZ, {ws_row=30, ws_col=80, ws_xpixel=0, ws_ypixel=0}) = 0
ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0
brk(NULL) = 0x80d2000
```

Elaborado por: Padilla Herrera Carlos Ignacio

```

brk(0x80d3000) = 0x80d3000
lstat64(".", {st_mode=S_IFDIR|0700, st_size=135, ...}) = 0
open(".", O_RDONLY|O_NONBLOCK|O_DIRECTORY|O_CLOEXEC) = 3
fstat64(3, {st_mode=S_IFDIR|0700, st_size=135, ...}) = 0
brk(0x80d4000) = 0x80d4000
getdents64(3, /* 6 entries */, 4096) = 168
brk(0x80d5000) = 0x80d5000
lstat64("./dos", {st_mode=S_IFDIR|0755, st_size=163, ...}) = 0
lstat64("./hello.c", {st_mode=S_IFREG|0644, st_size=242, ...}) = 0
getdents64(3, /* 0 entries */, 4096) = 0
close(3) = 0
write(1, "total 8\n", 8total 8
) = 8
open("/etc/group", O_RDONLY) = 3
close(3) = 0
open("/etc/passwd", O_RDONLY) = 3
ioctl(3, TCGETS, 0xbfcff50) = -
1 ENOTTY (Inappropriate ioctl for device)
read(3, "root:x:0:0:root:/root:/bin/sh\nda"..., 4096) = 334
close(3) = 0
write(1, "drwxr-xr-x  3 root  root  "..., 72drwxr-xr-x
x  3 root  root
      163 Aug 20  2011 dos
) = 72
write(1, "-rw-r--r--  1 root  root  "..., 75-rw-r--r-
-  1 root  root
      242 Jul 15 08:55 hello.c
) = 75
exit_group(0) = ?
+++ exited with 0 +++
[root@localhost ~]#
[root@localhost ~]#

```

Análisis

Comentando las líneas

```
execve("/bin/ls", ["ls", "-l"], [/ * 10 vars * /]) = 0
```

Es una llamada al Sistema que corre la función /bin/ls y los argumentos son ["ls", "-l"]

```

readlinkat(AT_FDCWD, "/proc/self/exe", "/bin/busybox", 4096) = 12
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS|MAP
UNINITIALI

```

Lee el valor de un vincula simbólico relative al FD (descriptor de archivos).

```

fstat(3, {st_mode=S_IFREG|0755, st_size=349804, ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS|MAP
UNINITIALI

```

devuelve la información del archivo

```
open("/lib/libc.so.1", O_RDONLY) = 3
```

Estos Access y open abren dependencias y librerías que ls usa.

```
mmap2(NULL, 368640, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -
1, 0) = 0xb7738000
```

Es asignación de memoria para ls

```
stat("/lib/ld-
uClibc.so.0", {st_mode=S_IFREG|0755, st_size=28696, ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS|MAP
UNINITIALI
```

Muestra el estado de la librería

```
close(3) = 0
```

Cierra una dependencia o librería.

```
ioctl(0, TCGETS, {B38400 opost isig icanon echo ...}) = 0
```

Configura las entradas/salidas

```
open("/etc/group", O_RDONLY) = 3
```

Abre el archivo con permiso de solo lectura de el argumento /etc/group

```
read(3, "root:x:0:0:root:/root:/bin/sh\nda"... , 4096) = 334
```

Lee algo de raíz, como entradas supongo

```
munmap(0xb7792000, 4096) = 0
```

remueve la memoria mapeada, un pedazo de memoria direccionada. Se mueve la ubicación de la interrupción

```
brk(NULL) = 0x80d2000
```

Lee el final del actual segmento de datos

```
open("/etc/passwd", O_RDONLY) = 3
```

Abre el argumento /etc/passwd de modo solo lectura

```
exit_group(0) = ?
```

Todas las llamadas al sistema pueden devolver algo, porque todas ellas pueden, al menos hipotéticamente, fallar, y te tienen que decir si lo hacen. Si no hay valor para ellas aparte de una indicación de éxito/error, regresan un entero que es cero para éxito y -1 para fallo, y strace imprime ese verbatim.

```
mprotect(0xb778d000, 4096, PROT_READ) = 0
```

Protege un conjunto sobre la región de memoria.

Conclusiones

Strace es un comando fácil de usar, y solo muestra las llamadas al Sistema. Es de fácil acceso y uso, pero puede causar problemas de rendimiento graves, causando que la aplicación objetivo se aliente hasta 100 veces más.

Referencias

<http://francisconi.org/linux/comandos/ls>

<https://www.youtube.com/watch?v=EG0ihttnEJI>

<http://www.brendangregg.com/blog/2014-05-11/strace-wow-much-syscall.html>

Elaborado por: Padilla Herrera Carlos Ignacio