Tarea 1: Depuración por *trazas*

Sistema operativo en el que se realizo: openSUSE

Programa empleado para obtener la traza: strace

Programa objetivo: ifconfig, lo elegí porque tengo interés de como funciona la red y en general en como se identifica y conecta cualquier dispositivo externo(tarjeta de sonido, video, red, lector de SD..) al S.O. Intente con lspci pero eran muchas lineas.

Pasos:

Primero utilice el comando strace -c ifconfig para que me listara las llamadas al sistema, así entrar en contexto investigando un poco que hace cada llamada, también me da información de cuantas llamadas se hace, que errores hubo y en que llamada.

% time	seconds	usecs/call	calls	errors	syscall
0.00	0.000000	0	14		read
0.00	0.000000	0	25		write
0.00	0.000000	0	11	2	open
0.00	0.000000	0	10		close
0.00	0.000000	0	10		fstat
0.00	0.000000	0	16		mmap
0.00	0.000000	0	4		mprotect
0.00	0.000000	0	6		munmap
0.00	0.000000	0	3		brk
0.00	0.000000	0	31	1	ioctl
0.00	0.000000	0	9	6	access
0.00	0.000000	0	3		socket
0.00	0.000000	0	1		execve
0.00	0.000000	0	1		uname
0.00	0.000000	0	1		arch_prctl
100.00	0.000000		145	9	total

A continuación utilice strace -T ifconfig, solo por curiosidad para darme una idea de cuanto tiempo se tarda en realizar cada llamada. Después strace -r ifconfig para saber el tiempo total que se llevo para realizar todas las llamadas y terminar la ejecución.

```
     0.000064 open("/proc/net/if_inet6", O_RDONLY) = 6
     0.000070 fstat(6, {st_mode=S_IFREG|0444, st_size=0, ...}) = 0
     0.000055 mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f6b129bd000
     0.000053 read(6, "00000000000000000000000000000001"..., 1024) = 108
     0.000076 read(6, "", 1024)         = 0
     0.000072 write(1, "          inet6 addr: fe80::1e3e"..., 62         inet6 addr: fe80::1e3e:84ff:fe71:7fd7/64 Scope:Link
) = 62
     0.000061 read(6, "", 1024)         = 0
     0.000048 close(6)                  = 0
     0.000055 munmap(0x7f6b129bd000, 4096) = 0
     0.000060 write(1, "          UP BROADCAST RUNNING M"..., 61         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
) = 61
     0.000064 write(1, "          RX packets:52103 error"..., 65         RX packets:52103 errors:0 dropped:0 overruns:0 frame:0
) = 65
     0.000059 write(1, "          TX packets:38898 error"..., 67         TX packets:38898 errors:0 dropped:0 overruns:0 carrier:0
) = 67
     0.000059 write(1, "          collisions:0 txqueuele"..., 40         collisions:0 txqueuelen:1000
) = 40
     0.000061 write(1, "          RX bytes:51138510 (48."..., 65         RX bytes:51138510 (48.7 Mb)  TX bytes:5211448 (4.9 Mb)
) = 65
     0.000057 write(1, "\n", 1
)        = 1
     0.000057 close(5)                  = 0
     0.000062 exit_group(0)             = ?
     0.000170 +++ exited with 0 +++
```

Finalmente utilice strace -f ifconfig para saber los procesos hijos que llama el S.O., esto para entender mejor que sucede en las entrañas.

Análisis.

```
linux-89ul:/tmp # strace -f ifconfig
execve("/sbin/ifconfig", ["ifconfig"], [/* 65 vars */]) = 0
brk(0)                                  = 0xd9a000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fe0a3509000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=151478, ...}) = 0
mmap(NULL, 151478, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fe0a34e4000
close(3)                                = 0
open("/lib64/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0000\34\2\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=1979344, ...}) = 0
mmap(NULL, 3832352, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fe0a2f42000
mprotect(0x7fe0a30e0000, 2097152, PROT_NONE) = 0
mmap(0x7fe0a32e0000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1
9e000) = 0x7fe0a32e0000
mmap(0x7fe0a32e6000, 14880, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0)
 = 0x7fe0a32e6000
close(3)                                = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fe0a34e3000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fe0a34e2000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fe0a34e1000
arch_prctl(ARCH_SET_FS, 0x7fe0a34e2700) = 0
mprotect(0x7fe0a32e0000, 16384, PROT_READ) = 0
mprotect(0x60d000, 4096, PROT_READ)     = 0
mprotect(0x7fe0a350a000, 4096, PROT_READ) = 0
munmap(0x7fe0a34e4000, 151478)          = 0
brk(0)                                  = 0xd9a000
brk(0xdbb000)                           = 0xdbb000
open("/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or direc
tory)
open("/usr/share/locale/locale.alias", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=2434, ...}) = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fe0a3508000
read(3, "# Locale name alias data base.\n#"..., 4096) = 2434
read(3, "", 4096)                       = 0
close(3)                                = 0
```

En esta primera parte entiendo que primero strace llama a ejecutar ifconfig, crea mapas de memoria, comprueba si se puede acceder a algunos archivos, lee parte de esos archivos y los deja en una dirección de memoria.

```
open("/usr/lib/locale/en_US.UTF-8/LC_CTYPE", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or
 directory)
open("/usr/lib/locale/en_US.utf8/LC_CTYPE", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=256420, ...}) = 0
mmap(NULL, 256420, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fe0a34a2000
close(3)                                = 0
open("/usr/lib64/gconv/gconv-modules.cache", O_RDONLY) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=26244, ...}) = 0
mmap(NULL, 26244, PROT_READ, MAP_SHARED, 3, 0) = 0x7fe0a3502000
close(3)                                = 0
```

Por estas lineas de arriba entiendo que lo que carga en memoria son librerías necesarias para resolver el ifconfig, pienso que son de idioma.

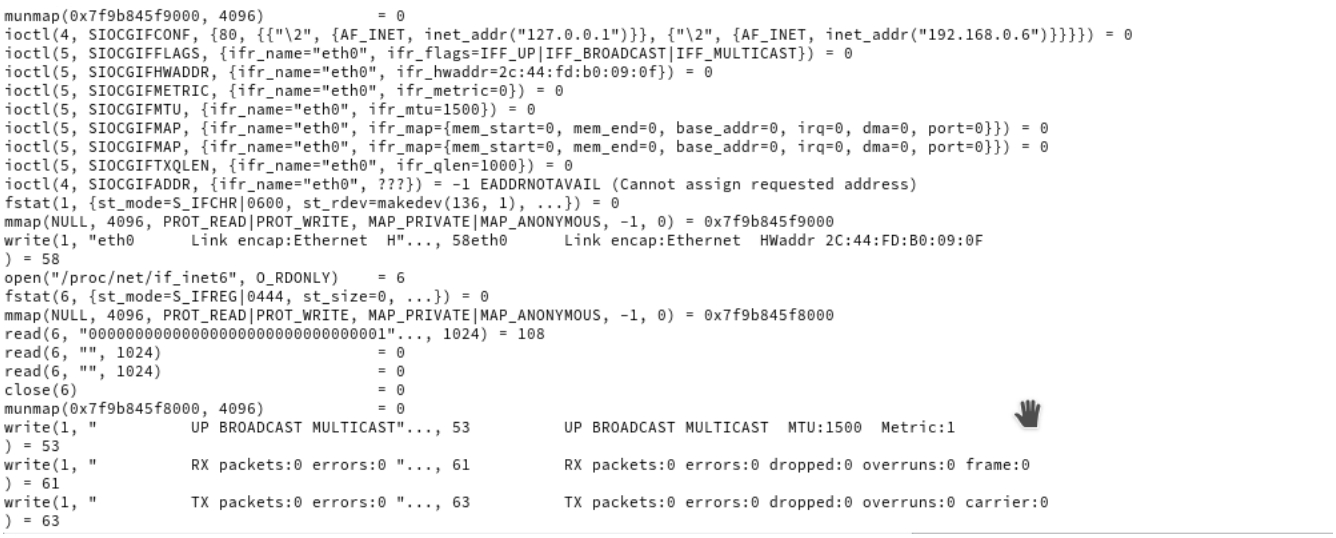
```
uname({sysname="Linux", nodename="linux-89ul.suse", ...}) = 0
```

Esta es una de las partes que me parece mas interesante, uname devuelve información de una estructura, quizá sea en este punto que comienza a traer información de la tarjeta de red.

```
access("/proc/net", R_OK)               = 0
access("/proc/net/unix", R_OK)          = 0
socket(PF_LOCAL, SOCK_DGRAM, 0)         = 3
socket(PF_INET, SOCK_DGRAM, IPPROTO_IP) = 4
access("/proc/net/if_inet6", R_OK)      = 0
socket(PF_INET6, SOCK_DGRAM, IPPROTO_IP) = 5
access("/proc/net/ax25", R_OK)          = -1 ENOENT (No such file or directory)
access("/proc/net/nr", R_OK)            = -1 ENOENT (No such file or directory)
access("/proc/net/ipx", R_OK)           = -1 ENOENT (No such file or directory)
access("/proc/net/appletalk", R_OK)     = -1 ENOENT (No such file or directory)
access("/proc/net/x25", R_OK)           = -1 ENOENT (No such file or directory)
open("/proc/net/dev", O_RDONLY)         = 6
fstat(6, {st_mode=S_IFREG|0444, st_size=0, ...}) = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f9b845f9000
read(6, "Inter-|   Receive                "..., 1024) = 570
read(6, "", 1024)                       = 0
close(6)                                = 0
```

Aquí, por tratarse de proc, lo que intenta revisar es si tiene acceso a algunas partes del kernel que tienen que ver con la red, y también se nota algo muy interesante, returns con -1, veo 5, estas las estaba esperando porque el comando strace -c ifconfig me alerto que buscara 6 errores relacionado con "access", lo que entiendo que sucede es que no tiene permitido este proceso acceder a esos archivos especiales.

También encuentro ya algo relacionado con leer del buffer, primero abre con "open" algún archivo especial relacionado con net, con "fstat" comprueba información del el y con "read" trae información del el(1024 bytes), desde el archivo 6 o eso pienso porque se repite en "fstat" también, al final cierra el archivo.



En este punto abre un espacio de memoria, y comienza a hacer peticiones por medio la "ioctl", a los archivos 4,5 que son archivos especiales que se comunican con los drivers, y se ve un eth0 que se refiere a una interface de red ethernet quizá pregunte si ese archivo tiene que ver con eth0.

Casualmente abajo escribe la primera linea que aparece al ejecutar ifconfig. Aquí hay algo escondido que no comprendo bien, de donde trajo esa cadena de caracteres que contiene hasta la dirección mac y como la cargo en otra llamada al sistema, quizá mas arriba cuando leyó los 1024 bytes trajo esa información, y cuando ejecuto "ioctl" comprobó algo de la mac porque pasa un parámetro ifr_hwaddr=…..

A continuación abre del archivo if_inet6, hace sus comprobaciones, genera un espacio de memoria, lee de el archivo "6", lo cierra, y comienza a escribir el resto de la información de la interface eth0.

También encuentro que siempre escribe a 1, quizá sea STDOUT.

Repite el proceso(desde "ioctl") con la interface lo y wlan0.

Y al final ejecuta "exit_group(0)", quiza signifique termina el grupo de procesos y subprocesos abiertos.

Por ultimo se me olvido comentar otro error esperando gracias al comando "strace -c", que es muy claro, intento abrir un archivo y no se encontró en el directorio indicado, eso causo otra llamada en seguida a open con los mismos parámetros, solo en otro path.

```
open("/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or direc
tory)
open("/usr/share/locale/locale.alias", O_RDONLY|O_CLOEXEC) = 3
```

Esquema de la traza.