

Sistema operativo Elementary OS (loki)

Programa empleado strace

Programa objetivo ps

Elegí este programa porque es el que tiene una mayor variedad de llamadas que los que originalmente había elegido. Los que me resultaban interesantes era **fuser** y **cron**, sin embargo me encontraba con llamadas muy poco variadas a mi parecer, por eso terminé eligiendo **ps**.

Resultados:

```
chicoterry@chicoterry:~|⇒ strace ps
..
. *Se ejecutan otras lineas previamente*
..
/* aquí se establece un apuntador en un hilo o posible proceso, el cual nos devuelve unicamente el ID de dicho hilo */
set_tid_address(0x7fc3af50fb50)      = 9009

/* esta llamada al sistema nos regresa el head de todos los futex (operaciones que se encuentran a la espera) de los hilos cuyo
ID se encuentren especificados en un determinado id de proceso (pid)*/
set_robust_list(0x7fc3af50fb60, 24)   = 0

rt_sigaction(SIGRTMIN, {0x7fc3adf93b50, [], SA_RESTORER|SA_SIGINFO, 0x7fc3adf9f390}, NULL, 8) = 0
rt_sigaction(SIGRT_1, {0x7fc3adf93be0, [], SA_RESTORER|SA_RESTART|SA_SIGINFO, 0x7fc3adf9f390}, NULL, 8) = 0

/* a grandes rasgos aqui se desbloquea una señal que quizás estuvimos manipulando en las llamadas anteriores*/
rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0

/* aqui se obtienen los limites de memoria ? establecidos para los procesos 'convencionales', refiriendome a aquellos que no
tienen privilegios */
getrlimit(RLIMIT_STACK, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0

/* aquí el sistema determina una accion de una señal que pareciera, es para cuando un proceso va a finalizar */
rt_sigaction(SIGHUP, {0x403220, ~[RTMIN RT_1], SA_RESTORER, 0x7fc05f3d34b0}, NULL, 8) = 0

/* naturalmente se abre un archivo que se encuentra en el directorio de procesos */
open("/proc/self/stat", O_RDONLY)    = 3

read(3, "8447 (ps) R 8445 8445 6803 34824"... , 1024) = 308
close(3)                               = 0

/* aquí se setean ciertos parámetros para archivos especiales, me atrevería a decir que especifica una cierta longitud para las
filas y columnas que se mostrarán en la tabla que muestra ps en la termina */
ioctl(1, TIOCGWINSZ, {ws_row=44, ws_col=170, ws_xpixel=0, ws_ypixel=0}) = 0

ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0
```

```

ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0

/*aquí se devuelve el usuario que se encuentra en uso del proceso, es decir el "efectivo" */
geteuid() = 1000
open("/proc/uptime", O_RDONLY) = 3

/* aquí vamos a reposicionar el offset del archivo que tiene el descriptor de archivo 3, para posteriormente seguir leyendolo */
lseek(3, 0, SEEK_SET) = 0
read(3, "559457.49 180622.11\n", 8191) = 20
open("/usr/share/locale/locale.alias", O_RDONLY|O_CLOEXEC) = 4

/* en esta parte se obtiene el estatus de un archivo el cual devuelve informacion importante debería ser relacionada a procesos,
tal el id del usuario y algunos metadatos como fechas de acceso y modificacion de dicho archivo*/
fstat(4, {st_mode=S_IFREG|0644, st_size=2995, ...}) = 0
read(4, "# Locale name alias data base.\n#", 4096) = 2995
read(4, "", 4096) = 0
close(4) = 0
open("/usr/share/locale/es_MX/LC_MESSAGES/procps-ng.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/usr/share/locale/es/LC_MESSAGES/procps-ng.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/usr/share/locale-langpack/es_MX/LC_MESSAGES/procps-ng.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/usr/share/locale-langpack/es/LC_MESSAGES/procps-ng.mo", O_RDONLY) = 4
fstat(4, {st_mode=S_IFREG|0644, st_size=50964, ...}) = 0
mmap(NULL, 50964, PROT_READ, MAP_PRIVATE, 4, 0) = 0x7fc05fdae000
close(4) = 0
open("/usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache", O_RDONLY) = 4
fstat(4, {st_mode=S_IFREG|0644, st_size=26258, ...}) = 0
mmap(NULL, 26258, PROT_READ, MAP_SHARED, 4, 0) = 0x7fc05fd16000
close(4) = 0

/* en este punto parece establecerse una operacion que estará a la espera, cuando se tenga una determinada condición, quizás
espera por la información referente a algun proceso */
futex(0x7fc05f7638c8, FUTEX_WAKE_PRIVATE, 2147483647) = 0
open("/proc/sys/kernel/pid_max", O_RDONLY) = 4

/* se obtiene información referente a la memoria */
open("/proc/meminfo", O_RDONLY) = 4
lseek(4, 0, SEEK_SET) = 0
read(4, "MemTotal:      11740312 kB\nMemF", 8191) = 1279
stat("/proc/self/task", {st_mode=S_IFDIR|0555, st_size=0, ...}) = 0
open("/proc", O_RDONLY|O_NONBLOCK|O_DIRECTORY|O_CLOEXEC) = 5
fstat(5, {st_mode=S_IFDIR|0555, st_size=0, ...}) = 0

```