

Práctica dos de sistemas operativos

Se trabajó en el sistema operativo: Porteus linux de 32 bits

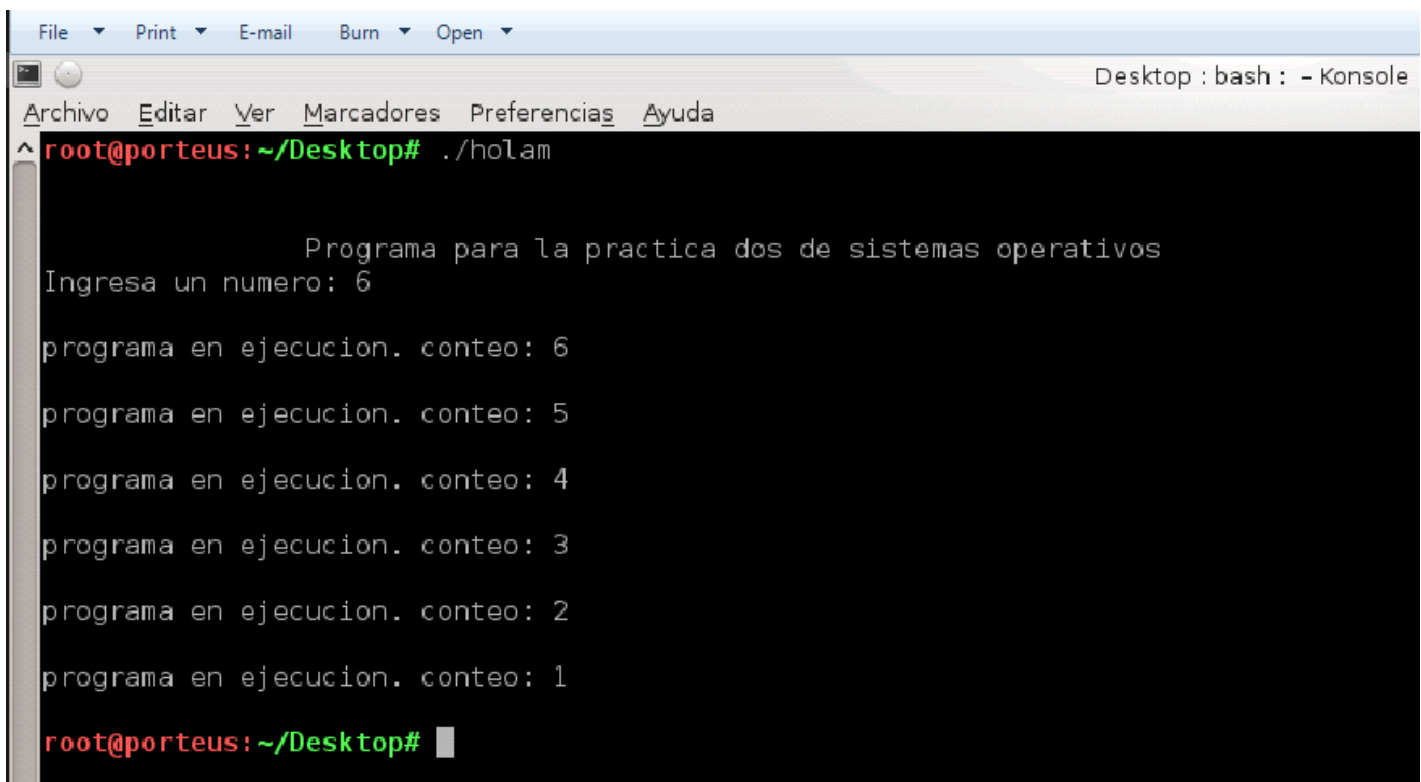
Se empleó strace para obtener la traza del programa.

Se eligió el programa “holam” porque es un programa sencillo que recibe un dato desde la terminal y muestra un mensaje en pantalla. “holam” es un programa de mi autoría.

Código fuente de “holam”:

```
holam.c
1 //programa para la practica dos de sistemas operativos
2 #include <stdio.h>//biblioteca de entrada y salida estandar
3
4 int main()
5 {
6     int conta, numero, conteo;//declaracion de variables
7     printf("\n\n\t\tPrograma para la practica dos de sistemas operativos");
8     printf("\nIngresa un numero: ");
9     scanf("%d", &numero);
10    //se muestra un mensaje en pantalla conforme al numero ingresado por el usuario
11    for(conta=0; conta < numero; conta++)
12    {
13        conteo=numero-conta;
14        printf("\nprograma en ejecucion. conteo: %d\n", conteo);
15    }
16
17    getchar();//cacha el enter del dato ingresado
18    getchar();//mantiene estatica la pantalla
19    return 0;
20 }
```

Ejemplo de ejecución de holam:



```
File Print E-mail Burn Open
Desktop : bash : - Konsole
Archivo Editar Ver Marcadores Preferencias Ayuda
^root@porteus: ~/Desktop# ./holam

        Programa para la practica dos de sistemas operativos
Ingresa un numero: 6

programa en ejecucion. conteo: 6
programa en ejecucion. conteo: 5
programa en ejecucion. conteo: 4
programa en ejecucion. conteo: 3
programa en ejecucion. conteo: 2
programa en ejecucion. conteo: 1
root@porteus: ~/Desktop#
```

```

root@porteus: ~/Desktop# strace ./holam
execve("./holam", ["/holam"], [/* 59 vars */]) = 0
brk(0) = 0x8197000
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7724000
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=78818, ...}) = 0
mmap2(NULL, 78818, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7710000
close(3) = 0
open("/lib/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0\320\230\1\0004\0\0\0"... , 512) = 512
fstat64(3, {st_mode=S_IFREG|0755, st_size=1611200, ...}) = 0
mmap2(NULL, 1620508, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0xb7584000
mmap2(0xb770a000, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x186000) = 0xb770a000
mmap2(0xb770d000, 10780, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0xb770d000
close(3) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7583000
set_thread_area({entry_number:-1 -> 6, base_addr:0xb75836c0, limit:1048575, seg_32bit:1, contents:0, read_exec_only:0, limit_in_
ent:0, useable:1}) = 0
mprotect(0xb770a000, 8192, PROT_READ) = 0
mprotect(0xb7749000, 4096, PROT_READ) = 0
munmap(0xb7710000, 78818) = 0
fstat64(1, {st_mode=S_IFCHR|0600, st_rdev=makedev(136, 1), ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7723000
write(1, "\n", 1
) = 1
write(1, "\n", 1
) = 1
write(1, "\t\tPrograma para la practica dos "..., 55          Programa para la practica dos de sistemas operativos
) = 55
fstat64(0, {st_mode=S_IFCHR|0600, st_rdev=makedev(136, 1), ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7722000
write(1, "Ingresa un numero: ", 19Ingresa un numero: ) = 19
read(0, 6
"6\n", 1024) = 2
write(1, "\n", 1
) = 1
write(1, "programa en ejecucion. conteo: 6"... , 33programa en ejecucion. conteo: 6
) = 33
write(1, "\n", 1
) = 1
write(1, "programa en ejecucion. conteo: 5"... , 33programa en ejecucion. conteo: 5
) = 33
write(1, "\n", 1
) = 1
write(1, "programa en ejecucion. conteo: 4"... , 33programa en ejecucion. conteo: 4
) = 33
write(1, "\n", 1
) = 1
write(1, "programa en ejecucion. conteo: 3"... , 33programa en ejecucion. conteo: 3
) = 33
write(1, "\n", 1
) = 1
write(1, "programa en ejecucion. conteo: 2"... , 33programa en ejecucion. conteo: 2
) = 33
write(1, "\n", 1
) = 1
write(1, "programa en ejecucion. conteo: 1"... , 33programa en ejecucion. conteo: 1
) = 33
read(0,
"\n", 1024) = 1
exit_group(0) = ?
+++ exited with 0 +++
root@porteus: ~/Desktop#

```

Analizando lo obtenido por strace:

Al ejecutarse holam hizo llamadas a otros programas contenidos en el sistema operativo.

*fstat envía información de estado sobre el archivo asociado.

*set_thread_area establece una entrada tls en el GDT (table de descriptors globales)

*mprotect establece protección de una region de memoria

*write escribe en un descriptor de archivo

*read leer desde un descriptor de archivo

Esas son todas las llamadas que logre identificar en la ejecución de holam. Sinceramente hay muchas que no son simples de entender debido a que no hay tanta familiaridad con analizar a este nivel lo que hace un programa respecto a las llamadas que realiza al sistema operativo. Sin embargo es muy interesante poder observar todo lo que conlleva la ejecución de un programa y tratar de entenderlo en la medida de lo posible.