Nicholas Singh
12/16/22
OS

Problem Set 7

```
nicholassingh72@DESKTOP-3V6E6AD:~/projects/ps7$ strace ./hello_noexit
execve("./hello_noexit", ["./hello_noexit"], 0x7fffdd73dc70 /* 20 vars */) = 0
write(1, "Hello world\n", 12Hello world
)           = 12
--- SIGSEGV {si_signo=SIGSEGV, si_code=SI_KERNEL, si_addr=NULL} ---
+++ killed by SIGSEGV (core dumped) +++
Segmentation fault (core dumped)
nicholassingh72@DESKTOP-3V6E6AD:~/projects/ps7$
```

Strace of program with no _exit system call

```
nicholassingh72@DESKTOP-3V6E6AD:~/projects/ps7$ as hello_exit.s -o hello_exit.o --64
nicholassingh72@DESKTOP-3V6E6AD:~/projects/ps7$ ld hello_exit.o -o hello_exit -m elf_x86_64
nicholassingh72@DESKTOP-3V6E6AD:~/projects/ps7$ ./hello_exit
Hello world
nicholassingh72@DESKTOP-3V6E6AD:~/projects/ps7$ strace ./hello_exit
execve("./hello_exit", ["./hello_exit"], 0x7fffcd0c7750 /* 20 vars */) = 0
write(1, "Hello world\n", 12Hello world
)           = 12
exit(13)                                = ?
+++ exited with 13 +++
nicholassingh72@DESKTOP-3V6E6AD:~/projects/ps7$ echo $?
13
nicholassingh72@DESKTOP-3V6E6AD:~/projects/ps7$
```

Assemble/link build process, result, strace, and $? result of program with _exit system call

```
nicholassingh72@DESKTOP-3V6E6AD:~/projects/ps7$ as hello_error.s -o hello_error.o --64
nicholassingh72@DESKTOP-3V6E6AD:~/projects/ps7$ ld hello_error.o -o hello_error -m elf_x86_64
nicholassingh72@DESKTOP-3V6E6AD:~/projects/ps7$ ./hello_error
nicholassingh72@DESKTOP-3V6E6AD:~/projects/ps7$ strace ./hello_error
execve("./hello_error", ["./hello_error"], 0x7fffdc84f450 /* 20 vars */) = 0
write(1, 0x4d2, 12)                     = -1 EFAULT (Bad address)
exit(13)                                = ?
+++ exited with 13 +++
```

Assemble/link build process, result, and strace of program with invalid parameter in write system call

Write Up

Problem 3: The result and strace output of the program without the _exit system call indicates a SIGSEGV being delivered to it. This is likely because there is no _exit, but more specifically, the kernel is trying to find something to execute but nothing is left in memory or it’s just random, invalid instructions so it’s left in a loop that requires a SIGSEGV.

Adding the _exit system call worked as predicted and passed the non-zero return integer specified. In this case, it was 13 and that was seen in both the strace output and the $? return value.

Problem 4: The program has an invalid parameter passed into write. $msg was replaced with $1234 which strace recognizes as an EFAULT. This is because the system call is expecting a string or const void (as the man page states) in order for proper output. Instead, it receives 1234 in hexadecimal which is clearly not allowed since it's an integer but the program still exits with the value 13.