

Problem 2 Sample Runs

I. Running strace ./main

```
derek@derek-VirtualBox:~/OS/PSET #7/Program$ strace ./main
execve("./main", [ "./main" ], 0x7fffc4f6d610 /* 53 vars */) = 0
brk(NULL)                               = 0x55f7afdd4000
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
access("/etc/ld.so.preload", R_OK)     = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=77247, ...}) = 0
mmap(NULL, 77247, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fe18c6b7000
close(3)                                 = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0260\34\2\0\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=2030544, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fe18c6b5000
mmap(NULL, 4131552, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fe18c0b2000
mprotect(0x7fe18c299000, 2097152, PROT_NONE) = 0
mmap(0x7fe18c499000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7fe18c499000
mmap(0x7fe18c49f000, 15072, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fe18c49f000
close(3)                                 = 0
arch_prctl(ARCH_SET_FS, 0x7fe18c6b64c0) = 0
mprotect(0x7fe18c499000, 16384, PROT_READ) = 0
mprotect(0x55f7af80c000, 4096, PROT_READ) = 0
mprotect(0x7fe18c6ca000, 4096, PROT_READ) = 0
munmap(0x7fe18c6b7000, 77247)            = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 0), ...}) = 0
brk(NULL)                               = 0x55f7afdd4000
brk(0x55f7afdf5000)                     = 0x55f7afdf5000
write(1, "Hello World!\n", 13Hello World!
)                                         = 13
exit_group(0)                           = ?
+++ exited with 0 +++
```

Figure 1: The write system call returns 13, as expected. Afterwards, the `exit_group()` function is called. This is similar to the `exit()` function, but applies to all threads in the calling process.

II. Testing asm.S

```
derek@derek-VirtualBox:~/OS/PSET #7/Program$ make
gcc -o main main.c
as --64 -o asm.o asm.S
ld -m elf_x86_64 asm.o
derek@derek-VirtualBox:~/OS/PSET #7/Program$ strace ./a.out
execve("./a.out", ["./a.out"], 0x7ffc3461b5b0 /* 53 vars */) = 0
write(0, "Hello World!\n", 13Hello World!
)          = 13
exit(0)          = ?
+++ exited with 0 +++
derek@derek-VirtualBox:~/OS/PSET #7/Program$ ./a.out
Hello World!
derek@derek-VirtualBox:~/OS/PSET #7/Program$ echo $?
0
```

Figure 2: The write system call returns 13 and the exit code is 0, as expected.