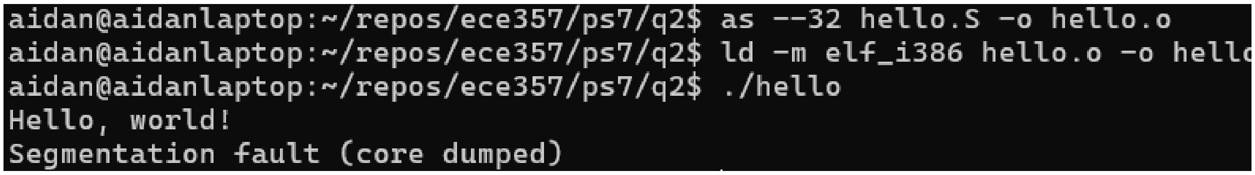
Aidan Cusa and Kristof Jablonowski

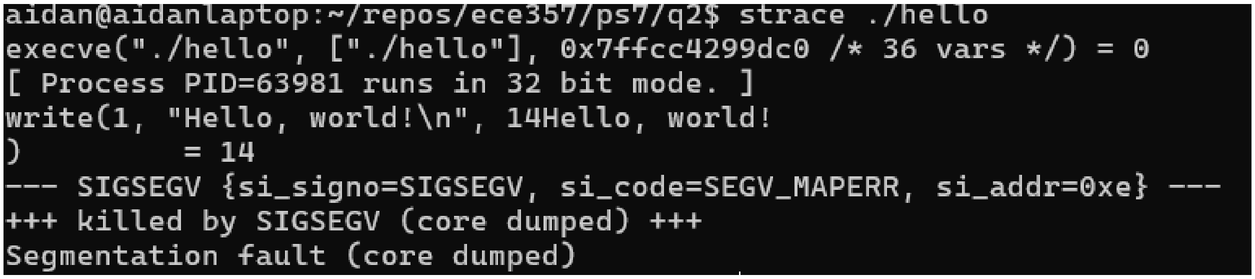
Problem 2B)

1. Assemble/link build process and 3. Output from program showing that message was written to stdout



```
aidan@aidanlaptop:~/repos/ece357/ps7/q2$ as --32 hello.S -o hello.o
aidan@aidanlaptop:~/repos/ece357/ps7/q2$ ld -m elf_i386 hello.o -o hell
aidan@aidanlaptop:~/repos/ece357/ps7/q2$ ./hello
Hello, world!
Segmentation fault (core dumped)
```

2. Strace output with write and exit syscalls

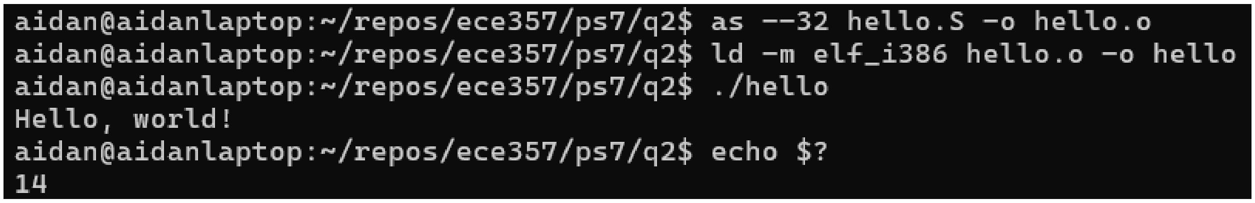


```
aidan@aidanlaptop:~/repos/ece357/ps7/q2$ strace ./hello
execve("./hello", ["./hello"], 0x7ffcc4299dc0 /* 36 vars */) = 0
[ Process PID=63981 runs in 32 bit mode. ]
write(1, "Hello, world!\n", 14Hello, world!
)         = 14
--- SIGSEGV {si_signo=SIGSEGV, si_code=SEGV_MAPERR, si_addr=0xe} ---
+++ killed by SIGSEGV (core dumped) +++
Segmentation fault (core dumped)
```

I think that it causes a seg fault and core dump as without exiting, the CPU continues to execute whatever follows _start in memory which might not be valid instructions and leads to illegal memory access.

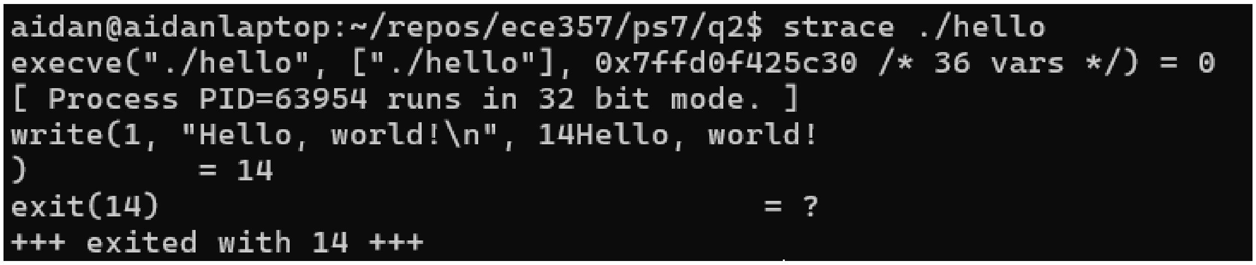
Problem 2C)

1. Assemble, link, execute, and echo $?



```
aidan@aidanlaptop:~/repos/ece357/ps7/q2$ as --32 hello.S -o hello.o
aidan@aidanlaptop:~/repos/ece357/ps7/q2$ ld -m elf_i386 hello.o -o hello
aidan@aidanlaptop:~/repos/ece357/ps7/q2$ ./hello
Hello, world!
aidan@aidanlaptop:~/repos/ece357/ps7/q2$ echo $?
14
```

2. strace results



```
aidan@aidanlaptop:~/repos/ece357/ps7/q2$ strace ./hello
execve("./hello", ["./hello"], 0x7ffd0f425c30 /* 36 vars */) = 0
[ Process PID=63954 runs in 32 bit mode. ]
write(1, "Hello, world!\n", 14Hello, world!
)         = 14
exit(14)                                = ?
+++ exited with 14 +++
```