HW 6: System Calls, AY2023 Spring Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

100 points total

(30) 1. Why does the operating system provide *system calls*, as opposed to just allowing user applications unfettered access to the CPU, devices, and memory?

|  |
| --- |
|  |

(20) 2. Use ltrace to enumerate the library function calls when you execute trace-me-1 with your favorite year as the input argument (for example, ltrace ./trace-me-1 1992). Copy the output below:

|  |
| --- |
|  |

List each library call by name (without its arguments):

|  |
| --- |
|  |

(30) 3. Use strace to enumerate the system calls when you execute trace-me-2 with your favorite year as the input argument (for example, strace ./trace-me-2 1992). Look carefully at the output, and see if you can discover the secret message. It is a multi-step process to discover it.

(hint 1: use strace -f to include a trace of a forked child process)

(hint 2: use the touch command to easily create an empty file with a given filename)

Secret message:

|  |
| --- |
|  |

Steps needed to obtain secret message (what commands you entered):

|  |
| --- |
|  |

(20) 4. The following x86\_64 assembly program uses system calls to accomplish a specific task. Using the class notes about system calls as a reference, as well as [this link](http://blog.rchapman.org/posts/Linux_System_Call_Table_for_x86_64/), describe below exactly what the program does. Feel free to assemble, link, and strace the code, if that helps.

SECTION .data

stuff1: db 0x45,0x4E,0x44,0x43,0x4F,0x56,0x49,0x44,0x13,0x0A

stuff2: db 0x4E,0x4F,0x4D,0x41,0x53,0x4B,0x00

SECTION .text

global \_start

\_start:

mov rax,85

mov rdi,stuff2

mov esi,0644Q

syscall

mov rdx,11

mov rsi,stuff1

mov rdi,rax

mov rax,1

syscall

mov rdi,0

mov rax,60

syscall

|  |
| --- |
|  |