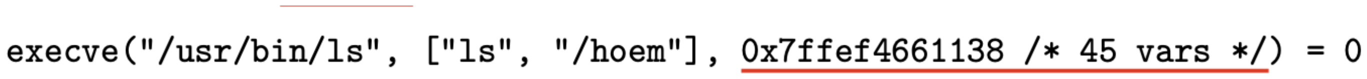
Week 4 Debugging

strace

`strace` is a diagnostic, debugging, and instructional userspace utility for Linux. It is used to monitor and tamper with interactions between processes and the Linux kernel, which includes ==*system calls, signal deliveries, and changes of process state*. ==

- The utility works by **tracing the system calls executed by a program**. Each line of the `strace` output represents a system call made by the program. The output includes
  the name of the system call
  its arguments
  its return value
  any errors that occur

e.g.



```
execve("/usr/bin/ls", ["ls", "/hoem"], 0x7ffef4661138 /* 45 vars */) = 0
```

The output includes the name of the

1. system call —> `execve`
2. its arguments —> an array of strings representing **the command-line arguments** that are passed to the program being executed
   1. `“/usr/bin/ls”`, the path of the program that is being executed
   2. `["ls", "/hoem"]`
      "/hoem", is an actual argument to the ls command, indicating the directory that ls is supposed to list. It appears to be a typo for "/home".
   3.

ltrace

`ltrace` is a debugging utility in Unix-like operating systems that is used to **intercept and record dynamic library calls** which are called by an executed process and the signals received by that process.

It can be used to

- *trace any dynamic library function calls, system calls made by a program, and signals received by a program.*

The primary use of `ltrace` is for debugging purposes when developing software. It helps programmers find bugs in their programs by tracking the library function calls that are made during the execution of a program.

- This can be particularly useful when trying to understand **how a program interacts with the system's libraries**, and it can also be used to troubleshoot problems related to library functions and to verify if the correct functions are being called.