Operating Systems
COMS W4118
Lecture 17

Alexander Roth

2015 – 03 – 31

1 Linux System Call Dispatch

- Suppose we wrote a hello world application that uses `printf`.
- `printf` is a library code that is run within your executable.
- At the end of the day, `printf` will call the system call `write`.
- During the `write` call, the system will enter the `kernel`.
- `write` is a C function; `write` is a library function but we regard it as a system call because the library function is pretty much the same as the system call.
- `write` stores the number for a write system call into a register.
- The instruction `int 0x80` calls a system interrupt to the OS.
- There are many types of software interrupts. `0x80` is reserved for invoking system calls.
- When you call `int 0x80`, we jump to the kernel to the function called `system_call`.
- `system_call` runs a function that points to a table called `sys_call_table`, which contains the `sys_write` function.
- Every system call has to have a corresponding C level library wrapper.
- System calls just invoke an interrupt with a number in a register.
- Arguments for system calls go into different registers.
- The act of invoking a system call should not be privileged because there needs to be a method to get you into the privileged mode.

2 System Call Parameter Passing

- In linux, when there are fewer than 6 parameters, they are passed into register.
- All linux system calls are either integers or pointer parameters.
- Linux has 300+ system calls.
- If you are writing a system call, you have to make sure that whatever parameter is passed in, must be checked.
- You cannot trust random pointer values that the user gave you.
- At the system call level, everything is privileged. Whatever happens will happen.
- If you blindly start adding values to a pointer, you could accidentally corrupt the kernel.
- You must make sure that whatever something points to points to the legitimate part of the program.
- If user passes a parameter, you must call a helper function to pull the information from the userspace.
- These helper functions can block becuase you need to allocate memory for the system.
- Memory allocation simply means a lot of things can happen in the background while you are locating a section of memory to access and backup.

3 Tracing System Calls in Linux

- `strace` used to trace system calls when a progarm is run.
- `strace` is useful for debugging a program.
- It's impossible to monitor a program without changing the state of the program.
- The kernel has to fundamental support tracing in the code in order to allow for debugging.
- You can give an environment variable to the main method.
- `brk` is a system call that allocates items onto the heap.
- `malloc` increases the heap region by some fixed amount and manages the size by some number of bytes.

- Unless you force it, gcc will not link the C library statically.
- The C library code is actually loaded dynamically at runtime.
- Thus, the newest version of the C code can be loaded in at once, and there can be one copy of the library code.
- All memory is virtual memory, which map to the physical space underneath.