Operating Systems
COMS W4118
Lecture 18

Alexander Roth

2015 – 04 – 02

1 The Linux Kernel

1.1 Header Files

- It's impossible to know everything about the linux kernel at any given time.
- It's just too big. *Phrasing.*
- You are not going to be able to become a code ninja over night.

1.2 Source Code

- The documentation directory holds all necessary documentation.
- The `kernel` file is the top level directory.

1.3 Processes or Threads

- The generic term in linux is `task`
- In theoretical operating systems, the names are usually process descriptor, process control block, task descriptor, etc.
- These things are some structure in the kernel memory that maintain a link list that represents running processes.
- If you have multiple threads in a process, they all share these resources.
- `task struct` maintain pointers to all the necessary values in the operating system.
- When a user program calls read, the system jumps into kernel mode.
- When we are in kernel mode, there is a special kernel stack that is created.

- The kernel cannot trust addresses provided by the user.
- It is not a privileged operation to load things into a register.
- The kernel is in complete control and only one copy of the kernel is running at a time, so you only need one kernel stack.
- Once you are in kernel mode, you are never going to switch processes.
- When linux had to support multiple CPU's, a locking mechanism was needed.
- A pre-empted kernel means that the kernel is safe to run on multiple CPU's at the same time.
- Every process has two stacks
  1. User stack
  2. Small part of the kernel stack (typically 4KB)
- Memory is counted in chunks or pages. One page is typically 4KB.

1.4 Process Descriptor

- `task_struct` is a huge structure.
- If you are running something in `sudo`, the effective user becomes `boot`, while the standard user is yourself.
- Effective user determines what you can and cannot do within a system.
- Linux uses part of a task's kernel-stack page-frame to store thread information.
- Do not use the word current because it's a defined macro.
- Processes are all interlinked in an embedded linked list.