Advanced Unix System Administration

Lecture 7
March 5, 2008

Steven Luo
<sluo+decal@OCF.Berkeley.EDU>

Resource Limits

- The ulimit facility con't
  - Linux/BSD limits
    - user's total number of processes (RLIMIT_NPROC)
    - physical memory used (RLIMIT_RSS)
    - locked memory (RLIMIT_MEMLOCK)
  - Setting limits
    - Processes can call setrlimit(2)
    - Use shell's ulimit from a shell script before running an application
    - PAM modules, etc. to set up limits for a user's login session

Startup and Shutdown

- Bootloader
  - Highly architecture-dependent behavior
- Kernel
  - Need to get enough loaded to find root partition, mount it, and launch userspace
  - Traditional fully monolithic kernels load all their code here
- init
  - Mounts filesystems, launches daemons, and brings up the system

Startup and Shutdown

- init(8)
  - PID 1, the “ultimate parent”
  - Spawns and respawns various children, according to configuration
  - Two traditional varieties: System V, BSD
- System V init binary
  - Used on Linux, Solaris, most commercial Unix
  - Configured via /etc/inittab
  - Uses “runlevels” to define the stages of boot and what should be running

Startup and Shutdown

- System V style runlevel handling
  - Usually performs actions when changing runlevels based on the contents of /etc/rcN.d, where N is the new runlevel
    - Scripts starting with S are run with argument “start”, scripts starting with K are run with argument “stop”
    - Two-digit number following S or K gives the ordering
  - Runlevel S is notionally invoked at the beginning of startup, 0 and 6 at halt or reboot

Startup and Shutdown

- BSD init binary
  - Used primarily on the BSDs
  - Launches a script /etc/rc when invoked, then spawns and respawns programs based on the contents of /etc/ttys
  - /etc/rc.shutdown is run on shutdown
- BSD style init handling
  - Only used by OpenBSD nowadays
  - /etc/rc and /etc/rc.shutdown do most/all of the work themselves

Startup and Shutdown

- Comparing BSD and SysV init handling
  - BSD-style init handling is simple and straightforward, but difficult to modify automatically
  - SysV init has more flexibility and modularity
  - With appropriate configuration of /etc/inittab or /etc/rc, SysV init binaries can be configured to behave BSD style (i.e. Slackware) and vice versa – or could behave entirely differently from either