System Initialization

init

- **`/sbin/init`** (usually just called **init**) is the first user-level process (or task) that runs on the system and continues to run until the system is shutdown
- Traditionally it has been considered the parent of all user processes, although technically that is not true as some processes are started directly by the kernel
- init coordinates the later stages of the boot process, configures all aspects of the environment and starts the processes needed for logging into the system; it also works closely with the kernel in cleaning up after processes when they terminate

SysVinit

- Traditionally, nearly all distributions based the init process on UNIX's venerable **SysVinit**
- This scheme was developed decades ago under rather different circumstances:
  - The target was multi-user mainframe systems (not personal computers, laptops, and other devices)
  - The target was a single processor system
  - Startup (and shutdown) time was not an important matter; it was far less important than getting things right
    - Startup was viewed as a serial process, divided into a series of sequential stages; each stage required completion before the next could proceed, so it didn't easily take advantage of the parallel processing that could be done on multiple processors or cores
    - Shutdown/reboot was seen as a relatively rare event and exactly how long it took was not considered important

New Methods of Controlling System Startup

To deal with these intrinsic limitations in SysVinit, new methods of controlling system startup were developed and have replaced it in all new systems. Two main schemes were adopted by Linux distributors:

- Upstart
    - Developed by Ubuntu and first included in the 6.10 release in 2006, and made the default in the 9.10 release in 2009
    - Adopted in Fedora 9 (in 2008) and in RHEL 6 and its clones, such as CentOS, Scientific Linux and Oracle Linux, and in openSUSE it was offered as an option since version 11.3
    - It has also been used in various embedded and mobile devices
- systemd is more recent and Fedora was the first major distribution to adopt it in 2011

systemd

- RHEL 7 is based on **systemd** and every other major Linux distribution has adopted it and has made it the default or announced plans to do so; even Ubuntu phased out Upstart and is now systemd-based
- We will discuss systemd commands, in particular the use of **systemctl**, when we discuss controlling system services; starting and stopping, enabling and disabling at boot, showing status, etc.
- Migration to systemd was non-trivial and missing features can be very disabling, so there are essential required compatibility layers
- Wrappers will ensure one can use SysVinit utilities and methods for quite some time, even if under the hood things are quite different
- However, eventually they will atrophy and disappear, so learning systemd is the best investment of your learning efforts

System Runlevels

As a SysVinit system starts, is passes through a sequence of runlevels which define different system states; they are numbered from 0 to 6:

- Runlevel 0 is reserved for the **system halt** state
- Runlevel 1 for **single-user mode**
- Runlevel 6 for **system reboot**
- The other runlevels are used to define what services are running for a normal system and different distributions define them somewhat differently
    - Example: On Red Hat-based systems, runlevel 2 is defined as a running system without networking or X, runlevel 3 includes networking, and runlevel 5 includes networking and X