System V Init

System Administration DeCal
Spring 2010
Michael Gasidlo

Administrivia

- You may have noticed the course website was down yesterday for several hours. If you were unable to complete the lab because of this, don't panic. You will have until next week to finish the lab
- I spent most of yesterday frantically trying to get said website (and the server that it's hosted on) back up rather than preparing for this class, so y'all have no lab this week

Semester Recap

- So far we've learned about the shell, shell scripting, network services, power tools, users and groups, compiling software, and some basic network services
- This is a good foundation to learn about pretty much anything else
- With simple tools, you can make advanced tools; with advanced tools, you can make machinery; with machinery, you can make anything

Today

- But all that don't mean diddly squat if your basic tools don't work
- How does Linux go from booting the kernel to presenting you with a working OS?
- Lots of stuff needs to happen before it can get to a working state- load drivers, system event loggers, network servers, X-Windows session...

System V Init

- Remnant of long ago (System V ~ 1983)
- Relatively unchanged in the intervening 27 years- ”If it ain't broke, don't fix it”
- A very simple system for running things at startup and shutdown
- The first thing the Linux kernel runs when it boots, and init starts everything else

Init(8) blow-by-blow

- System boots.
- init reads /etc/inittab which contains the list of run levels.
- For each run level, you can have a different set of startup and shutdown tasks.
- Runs all scripts in /etc/rcX.d (where X is the runlevel) sequentially.

Anatomy of an init script

- All init-compatible scripts conform to a standard of recognized arguments.
- You can run each rc.d script with “start”, “stop”, “restart”, or “reload”.
- Ex: /etc/rc2.d/S19mysql start
- Most init scripts are stored in /etc/init.d/, and symlinked where needed
- When booting up, all scripts are run with “start” argument, etc.

Script filenames

- Scripts start with either an S or a K
- If it starts with S, call “start” on it
- If it starts with K, call “stop” on it
- Starting runlevel could mean starting certain things and stopping others.
- For the most part, starting up will have mostly ”S” scripts, while shutdown will have mostly ”K” scripts
- Numbers matter, e.g. S19mysql – service is started in numerical order. S18 will start first
- This is more or less how dependencies are taken care of

DEMO!

Special runlevels

- Default run level - varies by distribution.
- Often 2 or 3
- 1 - single user maintenance mode (no SSH!)
- 0 or 5 - shut down system and power off
- 6 - shut down system and reboot
- Vservers don’t have init, sorry! Try it on a local Ubuntu install.

init shortcomings

- All scripts are run sequentially - they must terminate before the next one is run
- e.g. while waiting for one script, run all other non-dependent ones
- Also, (classic) init has no dependency system. You can’t tell what depends on e.g. the network being started
- Solution: new init system, such as Upstart (used by Ubuntu)

init = Serious Business

- Why is speeding up boot process important for e.g. Ubuntu?
- For most people, the faster you get to a desktop, the faster the OS seems to be.

1. Faster boot = Better (perceived) OS = more money
2. ???
3. Profit!!

Final Projects

- Make sure that you're working on your projects. Remember that your projects count for 60% of your grade.
- If you need additional resources (more vm's, ports opened, etc.) make sure to let me know
- As a general guideline, you should be compiling software that is directly related to your project, but you may apt-get ”peripheral” software (libraries, python, perl, etc)

GO VOTE!