Introduction to Linux Init Scripts

In this session we will cover the Linux initialization process, run levels, how to change the run level and how to initialize a script on login.


July 2012
You are free:

- to Share — to copy, distribute and transmit the work
- to Remix — to adapt the work
- to make commercial use of the work

Under the following conditions:

- Attribution — You must give the original author(s) credit
- Share Alike — If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.

With the understanding that:

- Waiver — Any of the above conditions can be waived if you get permission from the copyright holder.
- Public Domain — Where the work or any of its elements is in the public domain under applicable law, that status is in no way affected by the license.
- Other Rights — In no way are any of the following rights affected by the license.
- Notice — For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.

CC BY-SA 3.0 License:
http://creativecommons.org/licenses/by-sa/3.0/us/legalcode
Pre-work Check List

- Installed and configured VMWare Player v4 or later
- Installed Ubuntu 10.04
- Installed the latest Sitara Linux SDK and CCSv5
- Within the Sitara Linux SDK, ran the setup.sh (to install required host packages)
- Using a Sitara EVM, followed the QSG to connect ethernet, serial cables, SD card and 5V power
- Booted the EVM and noticed the Matrix GUI application launcher on the LCD
- Pulled the ipaddr of your EVM and ran remote Matrix using a web browser
- Brought the USB to Serial cable you confirmed on your setup (preferable)
Agenda

• Linux Init Process
• Linux Run Levels
• Initialization Directory
• Init Process Console Log
• Changing Run Levels
• Init Script Defaults
• Running a Process on Login
Linux Init Process

- The last step of the Linux Kernel boot process is to call the user space initialization function “init”
  - This is one call of several made by the kernel looking for the user space init function -
    run_init_process("/sbin/init"); (in init/main.c)

- Is the first User Space application to run that setups the user environment and allows user interaction such as login. Below is a ps (process status) dump from a console, init typically has a PID of 1, first process started.

<table>
<thead>
<tr>
<th>PID</th>
<th>USER</th>
<th>VSZ</th>
<th>STAT</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>root</td>
<td>1616</td>
<td>S</td>
<td>init [5]</td>
</tr>
</tbody>
</table>

- /etc/inittab is the script that contains instructions for the init process that sets up and calls other scripts as part of the user space initialization
  - Sets the run level, this in turn sets which init scripts based on run level are run
  - Identifier:RunLevel:Action:Command
  - Based on sysv5init scheme
Init Process – inittab example

• The inittab file is parsed by Init

• The inittab file contains position dependent
  – Identifier: RunLevel: Action: Command
  – Colon is used as a delimiter and is used to comment the line if placed in the first character of the line entry.

• id:5:initdefault:
  – Indicates which run level that init is supposed to enter

```bash
#!/etc/inittab: init(8) configuration.
# $Id: inittab,v 1.91 2002/01/25 13:35:21 miquels Exp $

# The default runlevel.
id:5:initdefault:

# Boot-time system configuration/initialization script.
# This is run first except when booting in emergency (-b) mode.
s1::sysinit:/etc/init.d/rcS

# What to do in single-user mode.
ss::wait:/sbin/suologin

# /etc/init.d executes the S and K scripts upon change
# of runlevel.

# Runlevel 0 is halt.
# Runlevel 1 is single-user.
# Runlevels 2-5 are multi-user.
# Runlevel 6 is reboot.

l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
# Normally not reached, but fallthrough in case of emergency.
z6:6:respawn:/sbin/suologin
S:2345:respawn:/sbin/getty ttyS2
```
Additional initab resource links

# Linux Run-Levels

- From Wikipedia

## Typical Linux runlevels

Major Linux distributions agreed to define the following runlevels as part of the Linux Standard Base specification[1]:

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Init</td>
<td>Shuts down the system.</td>
</tr>
<tr>
<td>1</td>
<td>Single-User Mode</td>
<td>Mode for administrative tasks. [2][3]</td>
</tr>
<tr>
<td>2</td>
<td>Multi-User Mode</td>
<td>Does not configure network interfaces and does not export networks services. [4]</td>
</tr>
<tr>
<td>3</td>
<td>Multi-User Mode with Networking</td>
<td>Starts the system normally. [5]</td>
</tr>
<tr>
<td>4</td>
<td>Not used/User-definable</td>
<td>For special purposes.</td>
</tr>
<tr>
<td>5</td>
<td>Start the system normally with appropriate display manager. (with GUI)</td>
<td>As runlevel 3 + display manager.</td>
</tr>
<tr>
<td>6</td>
<td>Reboot</td>
<td>Reboots the system.</td>
</tr>
</tbody>
</table>

^ = The additional behavior of this runlevel varies greatly. All distributions provide at least one virtual terminal. Some distributions start a login shell as the superuser; some require correctly entering the superuser's password first; others provide a login prompt, allowing any user access.

^ = In some cases, runlevels 2 and 3 function identically; offering a Multi-User Mode with Networking.
Initialization Directory (SysV)

- Initialization Directory /etc/init.d in the root file system
- All System initialization files are typically stored in this directory.

```
root@am180x-evm:/etc# ls init.d
alignment.sh    modutil.sh    sendsigs
alsa-state      mountall.sh   single
banner          mountnfs.sh   sysfs.sh
bootmisc.sh     networking    syslog
checkroot       populate-volatile.sh  sycleg.busybox
dbus-1          psqlash       telnetd
devpts.sh       ramdisk       tftpd
finish.sh       rc            udev
functions       rc3           udev-cache
halt            reboot         umountfs
hostname.sh     remologin     umountfs.sh
hwclock.sh      .. save-etc.sh urandom
```

- Initialization Directories for the different possible run levels in /etc/ in the root file system. Each run level has its own directory and contains scripts to be executed for that particular run-level.

```
-rw-r--r-- 2 root root  4096 Jul 14  2011 rc0.d
-rw-r--r-- 2 root root  4096 Jul 14  2011 rc1.d
-rw-r--r-- 2 root root  4096 Jul 14  2011 rc2.d
-rw-r--r-- 2 root root  4096 Jul 14  2011 rc3.d
-rw-r--r-- 2 root root  4096 Jul 14  2011 rc4.d
-rw-r--r-- 2 root root  4096 Jul 14  2011 rc5.d
-rw-r--r-- 2 root root  4096 Jul 14  2011 rc6.d
-rw-r--r-- 2 root root  4096 Jul 14  2011 rcS.d
```
Init process console log

- Init process console log
- Running in Kernel context before this line

```bash
INIT: version 2.06 booting
Please wait: booting...
Starting udev
udev: /proc/84/oom_adj is deprecated, please use /proc/84/oom_score_adj instead.
udev: starting version 141
Mounting root file system...
Caching udev daemon...
Populating dev cache
EXT3-fs: barriers not enabled
kjournal starting. Commit interval 5 seconds
EXT3-fs (mmcblk0p3): using internal journal
EXT3-fs (mmcblk0p3): mounted filesystem with writeback data node
mv: cannot rename '/cnp/devices': No such file or directory
NET: Registered protocol family 10
logger: mount: mount point /proc/bus/usb does not exist
ALSA: Restoring mixer settings...
Configuring network interfaces... davinci_mdio davinci_mdio.0: resetting idled controller
net eth0: attached PHY driver [SMSC LAN8710/LAN8720] (mii_bus:phy_addr=0:00, id=0e0f1)
ADDRCMD(255) ADDRCONF(255) ADDRDEF(255) ADDRDEV(255) ADDRDELETE(255)
eth0: link is not ready
udhcpc (v1.13.2) started
Sending discover...
Sending discover...
Sending discover...
No lease, forking to background
done.
Setting up IP spoofing protection: rp_filter.
Thu Jul 14 15:05:00 UTC 2011
INIT: Entering runlevel: 5
Starting system message bus: dbus.
Starting telnet daemon.
Starting syslogd/klogd: done
Starting thttpd.
```

•rcS.d

•rc5.d – starts here
Init Process – rcS.d

- rcS.d Base Initialization Directory contains several that all are symlinked to initialization directory /etc/init.d

- These are all executed in the order displayed, if you want to change the order they are executed modify Sxx pre-fix number to the string
Run Level Init Directory Files

- Run Level 5 Init Directory

```
root@am180x-evm:/etc# ls -la rc5.d/
```

```
-rwxr-xr-x 1 root root 4096 Jan 1 2000 .
-rwxr-xr-x 24 root root 4096 Jul 6 09:11 ..
```

- Adding files to this directory are called by init process

- Alphabetical is the order the scripts are executed in, S10 before S99

- S prefix is intended to mean startup,

- K prefix is intended to mean stoping a service or process, a script used to reset a possible running service. Typically called in Run Level 0 and 6 (halt and reboot)

- An ls command will list K<script>s first and be executed first. So if the developer wants to make sure that a previous process is stopped before starting a new process using the same resources.
Changing Run Levels

• The default run level can be changed by either:
  – Passing the desired run level in the kernel bootargs
    
    ```
    setenv bootargs \${bootargs_def} 3
    ```
  – Changing the default run level line in /etc/inittab
    
    ```
    id:3:initdefault:
    ```

• When the system is running you can also change run levels using the “init” command. For example:
  – To change from run level 3 to run level 5
    
    ```
    init 5
    ```
  – To reboot the system
    
    ```
    init 6
    ```

• NOTE: When switching run levels the rc utility will check if there is a Start (or S) script in the previous run level and there is no Kill (or K) script in the current level. To simplify, if you add a start script and the process was already started then it will not be run again
Init script defaults

• Many init scripts use files in the /etc/default directory to provide default settings.

• For example the /etc/default/rcS file contains settings used by the scripts in the /etc/rcS.d directory. Some examples are:
  – SULOGIN - whether to prompt for a login before moving to the default run-level.
  – UTC - whether the system clock is set to UTC time
  – ENABLE_ROOTFS_FSCK - whether to run fsck on the root file system on boot
  – FSCKFIX - whether to fix file system errors by default

• This is useful if you have a series of scripts needing access to the same defaults to avoid duplication of settings
Running a Process on Login

• The set of commands and processes to when a user logs into the system are in the “/etc/profile” file.

• This file usually sets default values for the PATH and other variables

• By adding commands to the /etc/profile file you can change what actions are performed on a successful login. Some common examples:
  – Adding additional directories to the path
  – Exporting environment variables

• Likewise additional files ending in .sh in the “/etc/profile.d” directory will be sourced.
  – This allows abstracting setting and functions to run to individual files rather than lumping everything into the /etc/profile file.
  – When creating packages you can add any custom setting used for login without having to tell the user how to modify their profile. i.e. TSLIB_TSDEVICE exported to the user’s environment
Linux Init Process Links

Thank you