Linux System Initialization

Advanced Embedded Linux Development

with **Dan Walkes**

University of Colorado **Boulder**

Learning objectives:

Linux Initialization/Startup Manager Options

Basics of Linux Init Scripts

Initialization

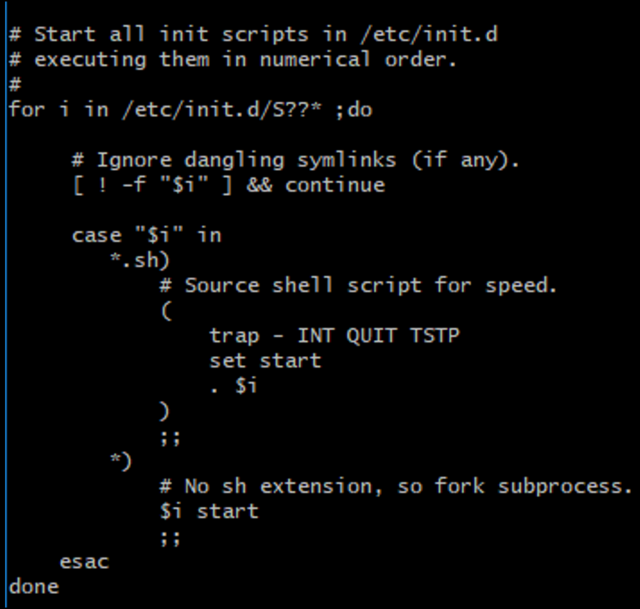
- Busybox Init
  - Default for buildroot projects
- System V (System 5) init
  - Default for Yocto projects
- systemd
  - More sophisticated startup manager

System V Init

- 1980s era Unix system
- Includes concept of runlevels
- S runlevel = startup tasks
- K runlevel = shutdown tasks
- Levels 1-5 are defined as single/multi user modes with or without graphical login support

Init Scripts

- In System V and Busybox cases, initialization is mostly driven with shell scripts
- Busybox
  - /etc/init.d/rcS
  - S = startup runlevel



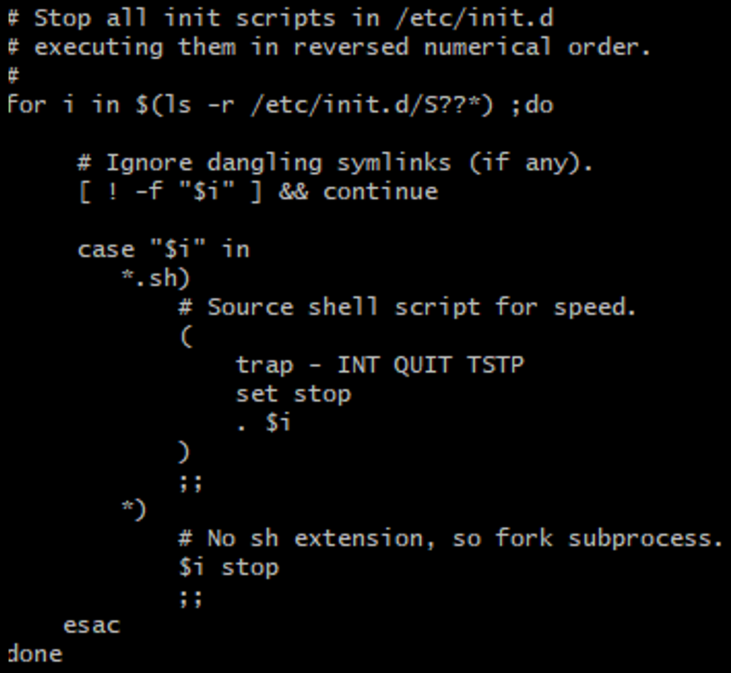
```
# Start all init scripts in /etc/init.d
# executing them in numerical order.
#
for i in /etc/init.d/S??* ;do

     # Ignore dangling symlinks (if any).
     [ ! -f "$i" ] && continue

     case "$i" in
        *.sh)
            # Source shell script for speed.
            (
                trap - INT QUIT TSTP
                set start
                . $i
            )
            ;;
        *)
            # No sh extension, so fork subprocess.
            $i start
            ;;
    esac
done
```

Init Scripts

- Busybox
  - /etc/init.d/rcK
  - K = kill or shutdown runlevel



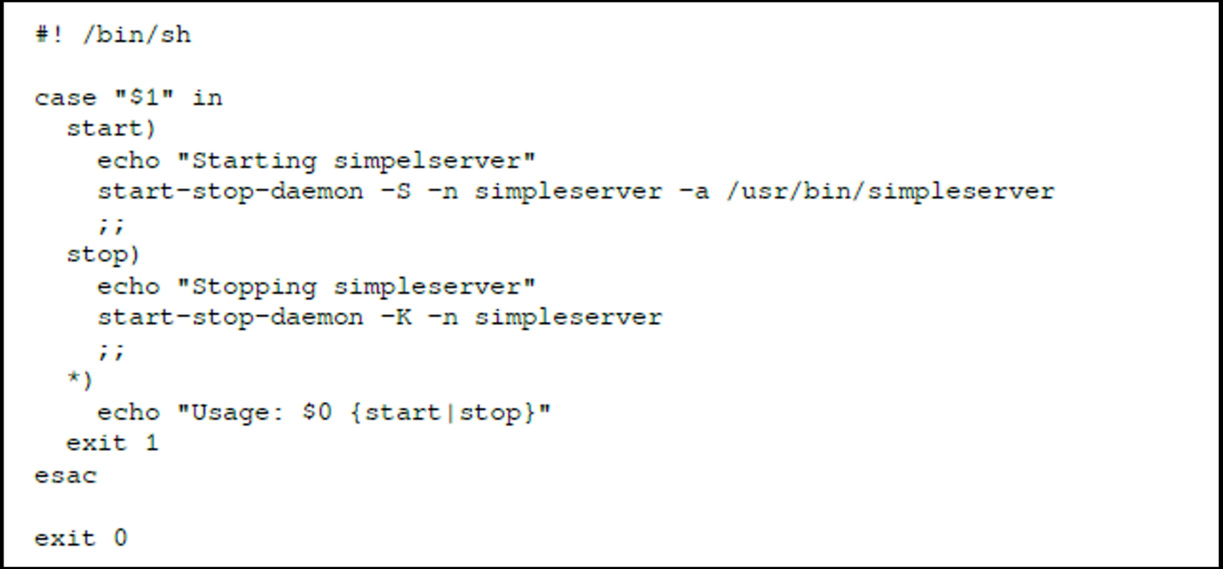
```
# Stop all init scripts in /etc/init.d
# executing them in reversed numerical order.
#
for i in $(ls -r /etc/init.d/S??*) ;do

     # Ignore dangling symlinks (if any).
     [ ! -f "$i" ] && continue

     case "$i" in
        *.sh)
            # Source shell script for speed.
            (
                trap - INT QUIT TSTP
                set stop
                . $i
            )
            ;;
        *)
            # No sh extension, so fork subprocess.
            $i stop
            ;;
    esac
done
```

Busybox /etc/init.d scripts

- Should support start and stop parameters
- Typically handle these in a case statement



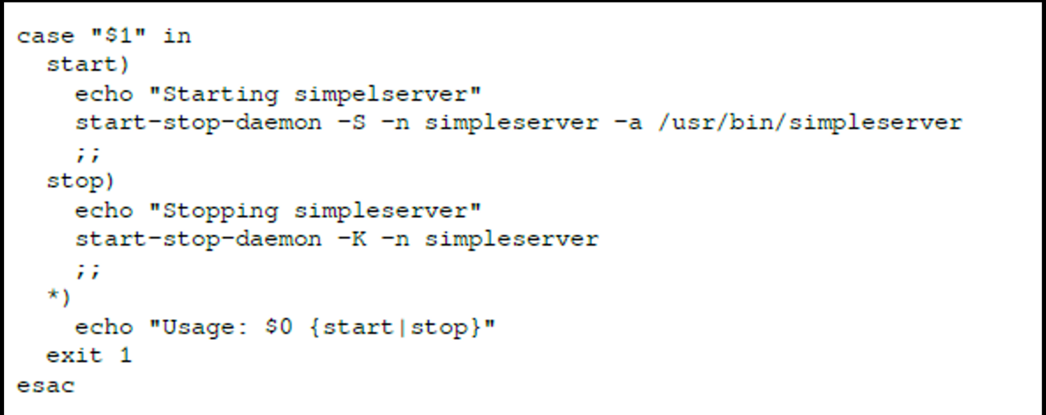
```
#! /bin/sh

case "$1" in
  start)
    echo "Starting simpelserver"
    start-stop-daemon -S -n simpleserver -a /usr/bin/simpleserver
    ;;
  stop)
    echo "Stopping simpleserver"
    start-stop-daemon -K -n simpleserver
    ;;
  *)
    echo "Usage: $0 {start|stop}"
  exit 1
esac

exit 0
```

start-stop-daemon

- init scripts should use start-stop-daemon
  - Starts with -S (if the daemon doesn’t exist)
  - Sends SIGTERM with -K
- See Week 4 content for notes about how to write the daemon

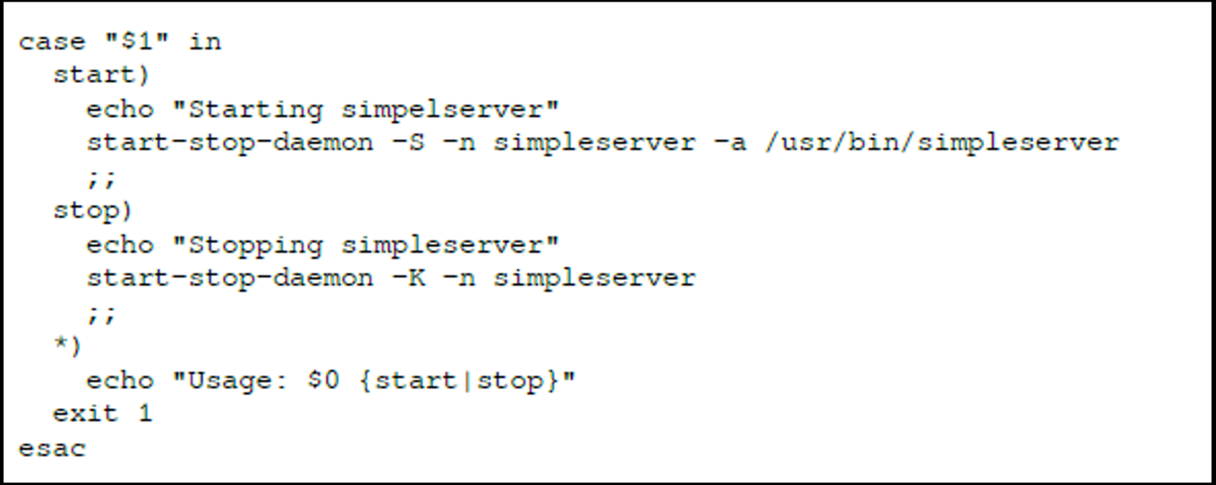


```
case "$1" in
  start)
    echo "Starting simpelserver"
    start-stop-daemon -S -n simpleserver -a /usr/bin/simpleserver
    ;;
  stop)
    echo "Stopping simpleserver"
    start-stop-daemon -K -n simpleserver
    ;;
  *)
    echo "Usage: $0 {start|stop}"
  exit 1
esac
```

Mastering Embedded Linux Programming Chapter 10
http://man7.org/linux/man-pages/man8/start-stop-daemon.8.html

Manually start and stop your script

- /etc/init.d/Smydaemon start
- /etc/init.d/Smydaemon stop



```
case "$1" in
  start)
    echo "Starting simpelserver"
    start-stop-daemon -S -n simpleserver -a /usr/bin/simpleserver
    ;;
  stop)
    echo "Stopping simpleserver"
    start-stop-daemon -K -n simpleserver
    ;;
  *)
    echo "Usage: $0 {start|stop}"
  exit 1
esac
```

Mastering Embedded Linux Programming Chapter 10
http://man7.org/linux/man-pages/man8/start-stop-daemon.8.html