# Understanding the Linux Boot Sequence

| | |
|---|---|
| File: | http://csplinux.saultc.on.ca/~fcarella/courses/cso201/boot/bootsequence.html |
| Date: | March 1999. |
| Description: | Describe what happens when Linux boots |
| Distributions: | **Slackware 3.6, Redhat** |
| Keywords: | init, inittab, run-levels, rc.d |
| References: | <1> Bootdisk-HOWTO<br><2> inittab man page (man inittab)<br><3> http://metalab.unc.edu/LDP/HOWTO/Unix-Internet-Fundamentals-HOWTO-3.html<br><4> http://www.linuxhq.com/guides/GS/node6.html (Linux Installation and getting started guide)<br><5> /etc/inittab<br><6> init man page (man init) |
| Glossary: | Go to it... |

## Overview

Ever wonder what's going on behind the scenes when you boot your Linux system and what's causing all those startup messages? Do you get those annoying cd-rom drive errors on boot because there's no cd in it and wonder what is trying to mount the cd and why? And how about those annoying "cant load so and so module errors"? Want to get rid of those. Serial ports; when are their baud rates set and otherwise initialized anyway? I wondered all those things so I did some hacking around and decided to share what I found with you through this document. Hope it helps you to understand the boot process a little better.

This document borrows heavily from pre-existing documents. It doesn't intend to replace them but to serve as a guide to and a summary of those documents. See references above...

### System Startup

<1>

- All PC systems start the boot process by executing code in ROM (specifically, the BIOS) to load the sector from sector 0, cylinder 0 of the boot drive.
  - The boot drive is usually the first floppy drive (designated A: in DOS and /dev/fd0 in Linux).
  - The BIOS then tries to execute this sector. On most bootable disks, sector 0, cylinder 0 contains either:
    - code from a boot loader such as LILO, which locates the kernel, loads it and executes it to start the boot proper.
    - the start of an operating system kernel, such as Linux.
  - Once the kernel is completely loaded, it goes through some basic device initialization. It then tries to load and mount a root filesystem from some device.
    - A root filesystem is simply a filesystem that is mounted as "/".
    - The kernel has to be told where to look for the root filesystem;
      - if it cannot find a loadable image there, it halts.
  - Once the root filesystem is loaded and mounted, you see a message like:

```
VFS: Mounted root (ext2 filesystem) readonly.
```

### Init

<1>

- At this point the system finds the init program on the root filesystem (in /bin or /sbin) and executes it. init reads its configuration file /etc/inittab, looks for a line designated sysinit, and executes the named script .

    - The sysinit script is usually something like /etc/rc or /etc/init.d/boot ***(in Slackware it's /etc/rc.d/rc.S)***. This script is a set of shell commands that set up basic system services, such as:
        - Running fsck on all the disks,
        - Loading necessary kernel modules,
        - Starting swapping,
        - Initializing the network,
        - Mounting disks mentioned in fstab.
    - This script (***/etc/rc.d/rc.S***) often invokes various other scripts to do modular initialization.
        - For example, in the common System V structure, the directory /etc/rc.d/ contains a complex structure of subdirectories whose files specify how to enable and shut down most system services.
- When the sysinit script finishes control returns to init, which then enters the default runlevel, specified in inittab with the initdefault keyword.
- The runlevel line usually specifies a program like getty, which is responsible for handling commununications through the console and ttys.
    - It is the getty program which prints the familiar `login:" prompt.
    - The getty program in turn invokes the login program to handle login validation and to set up user sessions.
- NOTE** Slackware uses a BSD style init structure while Redhat uses a System V init structure.

## Slackware 3.6 /etc/inittab

From the above discussion, we know that the program called "init" has a very important role in system startup since it starts up daemons, initializes the network, etc... "init" relies on /etc/inittab to tell it what to do.

#### Here is the /etc/inittab file for Slackware 3.6

```
#
# inittab       This file describes how the INIT process should set up
#               the system in a certain run-level.
#
# Version:      @(#)inittab             2.04    17/05/93        MvS
#                                       2.10    02/10/95        PV
#
# Author:       Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
# Modified by:  Patrick J. Volkerding, <volkerdi@ftp.cdrom.com>
#
# Default runlevel.
id:3:initdefault:

# System initialization (runs when system boots).
si:S:sysinit:/etc/rc.d/rc.S

# Script to run when going single user (runlevel 1).
su:1S:wait:/etc/rc.d/rc.K

# Script to run when going multi user.
rc:2345:wait:/etc/rc.d/rc.M

# What to do at the "Three Finger Salute".
ca::ctrlaltdel:/sbin/shutdown -t5 -rf now

# Runlevel 0 halts the system.
l0:0:wait:/etc/rc.d/rc.0

# Runlevel 6 reboots the system.
l6:6:wait:/etc/rc.d/rc.6
```

```
# What to do when power fails (shutdown to single user).
pf::powerfail:/sbin/shutdown -f +5 "THE POWER IS FAILING"

# If power is back before shutdown, cancel the running shutdown.
pg:0123456:powerokwait:/sbin/shutdown -c "THE POWER IS BACK"

# If power comes back in single user mode, return to multi user mode.
ps:S:powerokwait:/sbin/init 5

# The getties in multi user mode on consoles an serial lines.
#
# NOTE NOTE NOTE adjust this to your getty or you will not be
#                able to login !!
#
# Note: for 'agetty' you use linespeed, line.
# for 'getty_ps' you use line, linespeed and also use 'gettydefs'
c1:1235:respawn:/sbin/agetty 38400 tty1 linux
c2:1235:respawn:/sbin/agetty 38400 tty2 linux
c3:1235:respawn:/sbin/agetty 38400 tty3 linux
c4:1235:respawn:/sbin/agetty 38400 tty4 linux
c5:1235:respawn:/sbin/agetty 38400 tty5 linux
c6:12345:respawn:/sbin/agetty 38400 tty6 linux

# Serial lines
#s1:12345:respawn:/sbin/agetty 19200 ttyS0 vt100
#s2:12345:respawn:/sbin/agetty 19200 ttyS1 vt100

# Dialup lines
#d1:12345:respawn:/sbin/agetty -mt60 38400,19200,9600,2400,1200 ttyS0 vt100
#d2:12345:respawn:/sbin/agetty -mt60 38400,19200,9600,2400,1200 ttyS1 vt100

# Runlevel 4 used to be for an X-window only system, until we discovered
# that it throws init into a loop that keeps your load avg at least 1 all
# the time. Thus, there is now one getty opened on tty6. Hopefully no one
# will notice. ;^)
# It might not be bad to have one text console anyway, in case something
# happens to X.
x1:4:wait:/etc/rc.d/rc.4

# End of /etc/inittab
```

## Analysis of /etc/inittab

### First, some background on inittab from the inittab man page <2>

- An entry in the inittab file has the following format:

  id:runlevels:action:process
- Lines beginning with `#' are ignored.

Field descriptions (from the man page)

| | |
|---|---|
| id | is a unique sequence of 1-4 characters which identifies an entry in inittab (for versions of sysvinit compiled with libraries < 5.2.18 or a.out libraries the limit is 2 characters).<br>Note: For gettys or other login processes, the id field should be the tty suffix of the corresponding tty, e.g. 1 for tty1. Otherwise, the login accounting might not work correctly. |
| runlevels | lists the runlevels for which the specified action should be taken. |
| action | describes which action should be taken.<br><br>Valid actions for the action field are: |

| | |
|---|---|
| respawn | The process will be restarted whenever it terminates (e.g. getty). |
| wait | The process will be started once when the specified runlevel is entered and init will wait for its termination. |
| once | The process will be executed once when the specified runlevel is entered. |
| boot | The process will be executed during system boot. The runlevels field is ignored. |
| bootwait | The process will be executed during system boot, while init waits for its termination (e.g. /etc/rc). The runlevels field is ignored. |
| off | This does nothing. |
| ondemand | A process marked with an ondemand runlevel will be executed whenever the specified ondemand runlevel is called. However, no runlevel change will occur (ondemand runlevels are `a', `b', and `c'). |
| initdefault | An initdefault entry specifies the runlevel which should be entered after system boot. If none exists, init will ask for a runlevel on the console. The process field is ignored. |
| sysinit | The process will be executed during system boot. It will be executed before any boot or bootwait entries. The runlevels field is ignored. |
| powerwait | The process will be executed when init receives the SIGPWR signal, indicating that there is something wrong with the power. Init will wait for the process to finish before continuing. |
| powerfail | As for powerwait, except that init does not wait for the process's completion. |
| powerokwait | The process will be executed when init receives the SIGPWR signal, provided there is a file called /etc/powerstatus containing the word OK. This means that the power has come back again. |
| ctrlaltdel | The process will be executed when init receives the SIGINT signal. This means that someone on the system console has pressed the CTRL-ALT-DEL key combination. Typically one wants to execute some sort of shutdown either to get into single-user level or to reboot the machine. |
| kbrequest | The process will be executed when init receives a signal from the keyboard handler that a special key combination was pressed on the console keyboard. The documentation for this function is not complete yet; more documentation can be found in the kbdx.xx packages (most recent was kbd-0.94 at the time of this writing). Basically you want to map some keyboard combination to the "KeyboardSignal" action. For example, to map Alt-Uparrow for this purpose use the following in your keymaps file:<br>alt keycode 103 = KeyboardSignal |

| | |
|---|---|
| process | specifies the process to be executed. If the process field starts with a `+' character, init will not do utmp and wtmp accounting for that process. This is needed for gettys that insist on doing their own utmp/wtmp housekeeping. This is also a historic bug. |

**Line by Line Analysis of inittab**

| | |
|---|---|
| `# System initialization (runs when system boots).`<br>`si:S:sysinit:/etc/rc.d/rc.S`<br>`#` | init processes this line first. It tells init to execute the script rc.S |
| `# Default runlevel.`<br>`id:3:initdefault:`<br>`#` | after executing rc.S, control returns back to init which then enters the run level specified by this line ***(run level 3 in Slackware 3.6)***. |

| | |
|---|---|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

inittab --> rc.S

| | |
|---|---|
| #!/bin/sh<br>#<br># /etc/rc.d/rc.S:  System initialization script.<br>#<br># Mostly written by:  Patrick J. Volkerding, <volkerdi@ftp.cdrom.com><br># | |
| **PATH=/sbin:/usr/sbin:/bin:/usr/bin** | • set path |
| # enable swapping<br>**/sbin/swapon -a** | • turn swap on |
| # Start update.<br>**/sbin/update &** | • a daemon that periodically flushes the system buffers. |
| # Automatic module loading.  To load and unload kernel modules<br># automatically as needed, uncomment the lines below to run kerneld.<br># In some cases, you'll need to create aliases to load the correct<br># module. For more information, see the docs in /usr/doc/modules.<br># NOTE: This is commented out by default, since running kerneld has<br># caused some experimental kernels to hang during boot.<br>#if [ -x /sbin/kerneld ]; then<br>**#  /sbin/kerneld**<br>#fi | • turn on kerneld if it exists.<br>• kerneld will load and unload modules automatically |
| # Test to see if the root partition is read-only, like it ought to be.<br>READWRITE=no<br>**if echo -n >> "Testing filesystem status"; then**<br>**rm -f "Testing filesystem status"**<br>**READWRITE=yes**<br>**fi**<br><br># Check the integrity of all filesystems<br>if [ ! $READWRITE = yes ]; then<br>**/sbin/fsck -A -a**<br># If there was a failure, drop into single-user mode.<br>if [ $? -gt 1 ] ; then<br>echo<br>echo<br>echo "**********************************"<br>echo "fsck returned error code - REBOOT NOW!"<br>echo "**********************************"<br>echo | |

```
  echo
  /bin/login
 fi
 # Remount the root filesystem in read-write mode
 echo "Remounting root device with read-write enabled."
 /sbin/mount -w -n -o remount /
 if [ $? -gt 0 ] ; then
  echo
  echo "Attempt to remount root device as read-write failed!  This is
going to"
  echo "cause serious problems...  "
  echo
  echo "If you're using the UMSDOS filesystem, you **MUST** mount the
root partition"
  echo "read-write!  You can make sure the root filesystem is getting
mounted "
  echo "read-write with the 'rw' flag to Loadlin:"
  echo
  echo "loadlin vmlinuz root=/dev/hda1 rw   (replace /dev/hda1 with
your root device)"
  echo
  echo "Normal bootdisks can be made to mount a system read-write with
the rdev command:"
  echo
  echo "rdev -R /dev/fd0 0"
  echo
  echo "You can also get into your system by using a bootkernel disk
with a command"
  echo "like this on the LILO prompt line:  (change the root partition
name as needed)"
  echo
  echo "LILO: mount root=/dev/hda1 rw"
  echo
  echo "Please press ENTER to continue, then reboot and use one of the
above methods to"
  echo -n "get into your machine and start looking for the problem. "
  read junk;
 fi
else
 echo "Testing filesystem status: read-write filesystem"
 if cat /etc/fstab | grep ' / ' | grep umsdos 1> /dev/null 2>
/dev/null ; then
  ROOTTYPE="umsdos"
 fi
 if [ ! "$ROOTTYPE" = "umsdos" ]; then # no warn for UMSDOS
  cat << EOF

*** ERROR: Root partition has already been mounted read-write. Cannot
check!

For filesystem checking to work properly, your system must initially
mount
the root partition as read only. Please modify your kernel with 'rdev'
so that
it does this. If you're booting with LILO, add a line:

   read-only

to the Linux section in your /etc/lilo.conf and type 'lilo' to
reinstall it.
```

- mount the root device
- and perform an fsck if necessary

| | |
|---|---|
| If you boot from a kernel on a floppy disk, put it in the drive and type:<br>  rdev -R /dev/fd0 1<br><br>If you boot from a bootkernel disk, or with Loadlin, you can add the 'ro' flag.<br><br>This will fix the problem *AND* eliminate this annoying message. :^)<br><br>EOF<br>  echo -n "Press ENTER to continue. "<br>  read junk;<br> fi<br>fi | |
| # remove /etc/mtab* so that mount will create it with a root entry<br>**/bin/rm -f /etc/mtab* /etc/nologin /etc/shutdownpid** | • get rid of old /etc/mtab |
| # mount file systems in fstab (and create an entry for /)<br># but not NFS because TCP/IP is not yet configured<br>**/sbin/mount -avt nonfs** | • mount the file systems in /etc/fstab |
| # Clean up temporary files on the /var volume:<br>**/bin/rm -f /var/run/utmp /var/run/*.pid** | • get rid of stale files |
| # Looks like we have to create this.<br>**cat /dev/null > /var/run/utmp** | |
| # Configure the system clock.<br># This can be changed if your system keeps GMT.<br>if [ -x /sbin/clock ]; then<br>  **/sbin/clock -s**<br>fi<br><br>if [ "$ROOTTYPE" = "umsdos" ]; then # we need to update any files added in DOS:<br> echo "Synchronizing UMSDOS directory structure:"<br> echo "  umssync -r99 -v- /"<br> umssync -r99 -v- /<br>fi | • set clock |
| # Setup the /etc/issue and /etc/motd to reflect the current kernel level:<br># THESE WIPE ANY CHANGES YOU MAKE TO /ETC/ISSUE AND /ETC/MOTD WITH EACH<br># BOOT. COMMENT THEM OUT IF YOU WANT TO MAKE CUSTOM VERSIONS.<br>**echo > /etc/issue**<br>**echo Welcome to Linux `/bin/uname -a \| /bin/cut -d\  -f3`. >> /etc/issue**<br>**echo >> /etc/issue**<br>**echo "`/bin/uname -a \| /bin/cut -d\  -f1,3`." > /etc/motd** | • create /etc/issue.<br>• gets printed at the login prompt |
| # This loads any kernel modules that are needed.  These might be required to<br># use your CD-ROM drive, bus mouse, ethernet card, or other optional hardware.<br>**if [ -x /etc/rc.d/rc.modules ]; then**<br>**  . /etc/rc.d/rc.modules**<br>**fi** | • start /etc/rc.d/rc.modules<br>• this script loads up modules |

```
# Initialize PCMCIA devices:
#
# NOTE: This had been closer to the top of this script so that PCMCIA
devices
# could be fsck'ed along with the other drives.  This had some
unfortunate
# side effects, however, since root isn't yet read-write, and /var
might not
# even be mounted the .pid files can't be correctly written in
/var/run and
# the pcmcia system can't be correctly shut down.  If you want some
PCMCIA
# partition to be mounted at boot (or when the card is inserted) then
add
# the appropriate lines to /etc/pcmcia/scsi.opts.
#
if [ -x /etc/rc.d/rc.pcmcia ] ; then
  . /etc/rc.d/rc.pcmcia start
fi

# Run serial port setup script:
# (CAREFUL! This can make some systems hang if the rc.serial script
isn't
# set up correctly. If this happens, you may have to edit the file
from a
# boot disk)
#
# . /etc/rc.d/rc.serial
```

- start rc.pcmcia-->configure pcmcia devices
- start rc.serial-->configure serial ports

inittab-->rc.S-->rc.M

```
#!/bin/sh
#
# rc.M          This file is executed by init(8) when the system is being
#               initialized for one of the "multi user" run levels (i.e.
#               levels 1 through 6).  It usually does mounting of file
#               systems et al.
#
# Version:      @(#)/etc/rc.d/rc.M      2.02    02/26/93
#
# Author:       Fred N. van Kempen, <waltje@uwalt.nl.mugnet.org>
#               Heavily modified by Patrick Volkerding <volkerdi@ftp.cdrom.com>
#


# Tell the viewers what's going to happen...
echo "Going multiuser..."

# Screen blanks after 15 minutes idle time.
/bin/setterm -blank 15

# Look for a CD-ROM in a CD-ROM drive, and if one is found,
# mount it under /cdrom.  This must happen before any of the
# binaries on the CD are needed.
#
# If you don't have a CD-ROM and want to disable this, set the
# /etc/rc.d/rc.cdrom permissions to non-executable: chmod 644 /etc/rc.d/rc.cdrom
#
if [ -x /etc/rc.d/rc.cdrom ]; then
  . /etc/rc.d/rc.cdrom
```

```
fi

# If there's no /etc/HOSTNAME, fall back on this default:
if [ ! -r /etc/HOSTNAME ]; then
 echo "darkstar.frop.org" > /etc/HOSTNAME
fi

# Set the hostname.  This might not work correctly if TCP/IP is not
# compiled in the kernel.
/bin/hostname `cat /etc/HOSTNAME | cut -f1 -d .`

# Initialize the NET subsystem.
if [ -x /etc/rc.d/rc.inet1 ]; then
  . /etc/rc.d/rc.inet1
  . /etc/rc.d/rc.inet2
else
  if [ -x /usr/sbin/syslogd ]; then
    /usr/sbin/syslogd
    sleep 1 # Prevents a race condition with SMP kernels
    /usr/sbin/klogd
  fi
  if [ -x /usr/sbin/lpd ]; then
    /usr/sbin/lpd
  fi
fi

# Start netatalk. (a file/print server for Macs using Appletalk)
#if [ -x /etc/rc.d/rc.atalk ]; then
#  /etc/rc.d/rc.atalk
#fi              # Start crond (Dillon's crond):
# If you want cron to actually log activity to /var/adm/cron, then change
# -l10 to -l8 to increase the logging level.
/usr/sbin/crond -l10 >>/var/adm/cron 2>&1

# Remove stale locks and junk files (must be done after mount -a!)
/bin/rm -f /var/spool/locks/* /var/lock/* /var/spool/uucp/LCK..* /tmp/.X*lock /tmp/core /core 1> /dev/null 2>
/dev/null

# Remove stale hunt sockets so the game can start.
if [ -r /tmp/hunt -o -r /tmp/hunt.stats ]; then
  echo "Removing your stale hunt sockets from /tmp..."
  /bin/rm -f /tmp/hunt*
fi

# Ensure basic filesystem permissions sanity.
chmod 755 /
chmod 1777 /tmp /var/tmp

# Update all the shared library links automatically
/sbin/ldconfig

# Start the sendmail daemon:
if [ -x /usr/sbin/sendmail ]; then
  echo "Starting sendmail daemon (/usr/sbin/sendmail -bd -q15m)..."
  /usr/sbin/sendmail -bd -q15m
fi

# Start the APM daemon if APM is enabled in the kernel:
```

```
if [ -x /usr/sbin/apmd ]; then
  if cat /proc/apm 1> /dev/null 2> /dev/null ; then
    echo "Starting APM daemon..."
    /usr/sbin/apmd
  fi
fi

# Load a custom screen font if the user has an rc.font script.
if [ -x /etc/rc.d/rc.font ]; then
  . /etc/rc.d/rc.font
fi

# iBCS Emulation for Linux
# The Intel Binary Compatibility Specification, or iBCS, specifies the
# interfaces between application programs and the surrounding operating
# system environment for i386 based systems. There are however several
# flavours of iBCS in use - SVR4, SVR3 plus several vendor specific
# extensions to SVR3 which are slightly different and incompatible. The
# iBCS emulator for Linux supports all flavours known so far.
if [ -x /etc/rc.d/rc.ibcs2 ]; then
  . /etc/rc.d/rc.ibcs2
fi

# Start Web server:
if [ -x /etc/rc.d/rc.httpd ]; then
  . /etc/rc.d/rc.httpd
fi

# Start Samba (a file/print server for Win95/NT machines):
if [ -x /etc/rc.d/rc.samba ]; then
  . /etc/rc.d/rc.samba
# Load a custom keymap if the user has an rc.keymap script.
if [ -x /etc/rc.d/rc.keymap ]; then
  . /etc/rc.d/rc.keymap
fi

# Start the local setup procedure.
if [ -x /etc/rc.d/rc.local ]; then
  . /etc/rc.d/rc.local
fi

# All done.
```

Other startup scripts for Slack 3.6

rc.inet1...
rc.inet2...
rc.modules...
rc.0...
rc.4...
rc.6...
rc.K...
rc.M...
rc.S...
rc.cdrom...
rc.httpd...
rc.local...

rc.pcmcia...
rc.serial...

Order of execution of the rc.* scripts

- I modified the rc.* scripts to echo their names when executed. This shows the order their executed on boot.
- the following output shows the bootstrap messages

```
Loading.....................................................................
Uncompressing Linux...done.
Now booting the kernel
Console: 16 point font, 400 scans
Console: colour VGA+ 80x25, 1 virtual console (max 63)
pci_init: no BIOS32 detected
Calibrating delay loop.. ok - 49.87 BogoMIPS
Memory: 22304k/24576k available (1052k kernel code, 384k reserved, 836k
data)
This processor honours the WP bit even when in supervisor mode. Good.
Swansea University Computer Society NET3.035 for Linux 2.0
NET3: Unix domain sockets 0.13 for Linux NET3.035.
Swansea University Computer Society TCP/IP for NET3.034
IP Protocols: IGMP, ICMP, UDP, TCP
VFS: Diskquotas version dquot_5.6.0 initialized
Checking 386/387 coupling... Ok, fpu using exception 16 error reporting.
Checking 'hlt' instruction... Ok.
Linux version 2.0.35 (root@darkstar) (gcc version 2.7.2.3) #13 Mon Oct 26
22:12:45 CST 1998
Starting kswapd v 1.4.2.2
Serial driver version 4.13 with no serial options enabled
tty00 at 0x03f8 (irq = 4) is a 16450
tty01 at 0x02f8 (irq = 3) is a 16450
Real Time Clock Driver v1.09
Ramdisk driver initialized : 16 ramdisks of 4096K size
hda: NEC Corporation D3725, 515MB w/88kB Cache, CHS=524/32/63
hdb: 625A, ATAPI CDROM drive
ide0 at 0x1f0-0x1f7,0x3f6 on irq 14
Floppy drive(s): fd0 is 1.44M
FDC 0 is an 8272A
md driver 0.36.3 MAX_MD_DEV=4, MAX_REAL=8
linear personality registered
raid0 personality registered
scsi : 0 hosts.
scsi : detected total.
PPP: version 2.2.0 (dynamic channel allocation)
TCP compression code copyright 1989 Regents of the University of California
PPP Dynamic channel allocation code copyright 1995 Caldera, Inc.
PPP line discipline registered.
SLIP: version 0.8.4-NET3.019-NEWTTY (dynamic channels, max=256).
CSLIP: code copyright 1989 Regents of the University of California.
eth0: 3c509 at 0x300 tag 1, 10baseT port, address 00 20 af 3e 0f 0b, IRQ
10.
3c509.c:1.12 6/4/97 becker@cesdis.gsfc.nasa.gov
arcnet.c: v2.56 96/10/18 Avery Pennarun <apenwarr@foxnet.net>
arc0: Stage 3: No ARCnet cards found.
Partition check:
hda: hda1 hda2
```

```
VFS: Mounted root (ext2 filesystem) readonly.
INIT: version 2.73 booting
----------------------------------------------------
INITTAB --> rc.S
----------------------------------------------------
Adding Swap: 34268k swap-space (priority -1)
/etc/rc.d/rc.S: Testing filesystem status: Read-only file system
Parallelizing fsck version 1.12 (9-Jul-98)
/dev/hda1: clean, 26031/123952 files, 336526/493888 blocks
Remounting root device with read-write enabled.
none on /proc type proc (rw)
----------------------------------------------------
INITTAB --> rc.modules
----------------------------------------------------
Updating module dependencies for Linux 2.0.35:
lp1 at 0x0378, (polling)
SLIP: version 0.8.4-NET3.019-NEWTTY-MODULAR (dynamic channels, max=256).
PPP: version 2.2.0 (dynamic channel allocation)
PPP Dynamic channel allocation code copyright 1995 Caldera, Inc.
PPP line discipline registered.
No PS/2 mouse device found on this machine.
----------------------------------------------------
INITTAB --> rc.pcmcia
----------------------------------------------------
Starting PCMCIA services:Linux PCMCIA Card Services 3.0.5
kernel build: 2.0.35 #2 Sun Oct 11 03:38:56 CDT 1998
options: [pci] [cardbus]
<Probing for PCIC: edit /etc/rc.d/rc.pcmcia>
Intel PCIC probe: not found.
Databook TCIC-2 PCMCIA probe: not found.
Mar 11 10:53:17 cardmgr[48]: starting, version is 3.0.5
INIT: Entering runlevel: 3
----------------------------------------------------
INITTAB --> rc.M
----------------------------------------------------
Going multiuser...
Mar 11 10:56:02 cardmgr[48]: no sockets found!
Mar 11 10:56:02 cardmgr[48]: exiting
----------------------------------------------------
INITTAB --> rc.inet1
----------------------------------------------------
eth0: Setting Rx mode to 1 addresses.
----------------------------------------------------
INITTAB --> rc.inet2
----------------------------------------------------
Mounting remote file systems...
Starting daemons: syslogd klogd portmap inetd lpd mountd nfsd
Starting sendmail daemon (/usr/sbin/sendmail -bd -q15m)...
----------------------------------------------------
INITTAB --> rc.httpd
----------------------------------------------------
/var/lib/apache/sbin/apachectl start: httpd started
----------------------------------------------------
INITTAB --> rc.local
----------------------------------------------------
Running gpm...
```

```
Welcome to Linux 2.0.35.

freds486 login:
```

# RedHat 6.0 inittab

## Read this -->

/etc-->
inittab...
/etc/rc.d-->
rc...
rc.local...
rc.news...
rc.sysinit...

init.d/-->

- amd*
- apmd*
- arpwatch*
- atd*
- autofs*
- bootparamd*
- crond*
- dhcpd*
- functions*
- gated*
- gpm*
- halt*
- httpd*
- inet*
- innd*
- keytable*
- killall
- linuxconf -> /usr/lib/linuxconf/redhat/scripts/linuxconf
- mars-nwe*
- lpd
- mcserv*
- named*
- netfs*
- network*
- nfs*
- nscd*
- pcmcia*
- portmap*
- postgresql*
- random*
- routed*
- rstatd*

rusersd*
rwalld*
rwhod*
sendmail*
single*
smb*
snmpd*
sound*
squid*
syslog*
xfs*
xntpd*
ypbind*
yppasswdd*
ypserv*

rc0.d/

K00linuxconf -> ../init.d/linuxconf
K05innd -> ../init.d/innd*
K05keytable -> ../init.d/keytable*
K08autofs -> ../init.d/autofs*
K10xfs -> ../init.d/xfs*
K15gpm -> ../init.d/gpm*
K15httpd -> ../init.d/httpd*
K15sound -> ../init.d/sound*
K20bootparamd -> ../init.d/bootparamd*
K20nfs -> ../init.d/nfs*
K20rstatd -> ../init.d/rstatd*
K20rusersd -> ../init.d/rusersd*
K20rwalld -> ../init.d/rwalld*
K20rwhod -> ../init.d/rwhod*
K25squid -> ../init.d/squid*
K28amd -> ../init.d/amd*
K30mcserv -> ../init.d/mcserv*
K30sendmail -> ../init.d/sendmail*
K34yppasswdd -> ../init.d/yppasswdd*
K35dhcpd -> ../init.d/dhcpd*
K45named -> ../init.d/named*
K50inet -> ../init.d/inet*
K50snmpd -> ../init.d/snmpd*
K55routed -> ../init.d/routed*
K60atd -> ../init.d/atd*
K60crond -> ../init.d/crond*
K60lpd -> ../init.d/lpd*
K60mars-nwe -> ../init.d/mars-nwe*
K75gated -> ../init.d/gated*
K80nscd -> ../init.d/nscd*
K80random -> ../init.d/random*
K85netfs -> ../init.d/netfs*
K88ypserv -> ../init.d/ypserv*
K89portmap -> ../init.d/portmap*
K90killall -> ../init.d/killall*
K90network -> ../init.d/network*
K92apmd -> ../init.d/apmd*
K96pcmcia -> ../init.d/pcmcia*
K99syslog -> ../init.d/syslog*
S00halt -> ../init.d/halt*

rc1.d/

K00linuxconf -> ../init.d/linuxconf
K05innd -> ../init.d/innd*
K05keytable -> ../init.d/keytable*
K08autofs -> ../init.d/autofs*
K10xfs -> ../init.d/xfs*
K15gpm -> ../init.d/gpm*
K15httpd -> ../init.d/httpd*
K15sound -> ../init.d/sound*
K20bootparamd -> ../init.d/bootparamd*
K20nfs -> ../init.d/nfs*
K20rstatd -> ../init.d/rstatd*
K20rusersd -> ../init.d/rusersd*
K20rwalld -> ../init.d/rwalld*
K20rwhod -> ../init.d/rwhod*
K25squid -> ../init.d/squid
K28amd -> ../init.d/amd*
K30mcserv -> ../init.d/mcserv*
K30sendmail -> ../init.d/sendmail*
K34yppasswdd -> ../init.d/yppasswdd*
K35dhcpd -> ../init.d/dhcpd*
K45named -> ../init.d/named*
K50inet -> ../init.d/inet*
K50snmpd -> ../init.d/snmpd*
K55routed -> ../init.d/routed*
K60atd -> ../init.d/atd*
K60crond -> ../init.d/crond*
K60lpd -> ../init.d/lpd*
K60mars-nwe -> ../init.d/mars-nwe*
K75gated -> ../init.d/gated*
K80nscd -> ../init.d/nscd*
K85netfs -> ../init.d/netfs*
K88ypserv -> ../init.d/ypserv*
K89portmap -> ../init.d/portmap*
K90network -> ../init.d/network*
K92apmd -> ../init.d/apmd*
K96pcmcia -> ../init.d/pcmcia*
K99syslog -> ../init.d/syslog*
S00single -> ../init.d/single*
S20random -> ../init.d/random*

rc2.d/

xK05innd -> ../init.d/innd*
K08autofslrwxrwxrwx 1 root root 15 Jun 30 1999 K15sound -> ../init.d/sound*
K20bootparamd -> ../init.d/bootparamd*
K20nfs -> ../init.d/nfs*
K20rstatd -> ../init.d/rstatd*
K20rusersd -> ../init.d/rusersd*
K20rwalld -> ../init.d/rwalld*
K20rwhod -> ../init.d/rwhod*
K25squid -> ../init.d/squid*
K28amd -> ../init.d/amd*
K30mcserv -> ../init.d/mcserv*
K34yppasswdd -> ../init.d/yppasswdd*
K50inet -> ../init.d/inet*

K50snmpd -> ../init.d/snmpd*
K55routed -> ../init.d/routed*
K60atd -> ../init.d/atd*
K60mars-nwe -> ../init.d/mars-nwe*
K75gated -> ../init.d/gated
K80nscd -> ../init.d/nscd*
K85netfs -> ../init.d/netfs*
K88ypserv -> ../init.d/ypserv*
K89portmap -> ../init.d/portmap*
S05apmd -> ../init.d/apmd*
S10network -> ../init.d/network*
S20random -> ../init.d/random*
S30syslog -> ../init.d/syslog*
S40crond -> ../init.d/crond*
S45pcmcia -> ../init.d/pcmcia*
S60lpd -> ../init.d/lpd*
S65dhcpd -> ../init.d/dhcpd*
S75keytable -> ../init.d/keytable*
S80sendmail -> ../init.d/sendmail*
S85gpm -> ../init.d/gpm*
S90xfs -> ../init.d/xfs*
S99linuxconf -> ../init.d/linuxconf
S99local -> ../rc.local*  -> ../init.d/autofs*

rc3.d/

K08autofs -> ../init.d/autofs*
K10xntpd -> ../init.d/xntpd*
K20bootparamd -> ../init.d/bootparamd*
K20rstatd -> ../init.d/rstatd*
K20rusersd -> ../init.d/rusersd*
K20rwalld -> ../init.d/rwalld*
K20rwhod -> ../init.d/rwhod*
K25squid -> ../init.d/squid*
K30mcserv -> ../init.d/mcserv*
K34yppasswdd -> ../init.d/yppasswdd*
K45arpwatch -> ../init.d/arpwatch*
K55routed -> ../init.d/routed*
K60mars-nwe -> ../init.d/mars-nwe*
K75gated -> ../init.d/gated*
K80nscd -> ../init.d/nscd*
K88ypserv -> ../init.d/ypserv*
K96pcmcia -> ../init.d/pcmcia*
S05apmd -> ../init.d/apmd*
S10network -> ../init.d/network*
S11portmap -> ../init.d/portmap*
S15netfs -> ../init.d/netfs*
S20random -> ../init.d/random*
S30syslog -> ../init.d/syslog*
S40atd -> ../init.d/atd*
S40crond -> ../init.d/crond*
S50inet -> ../init.d/inet*
S50snmpd -> ../init.d/snmpd*
S55named -> ../init.d/named*
S60lpd -> ../init.d/lpd*
S60nfs -> ../init.d/nfs*

S65dhcpd -> ../init.d/dhcpd*
S72amd -> ../init.d/amd*
S75keytable -> ../init.d/keytable*
S80sendmail -> ../init.d/sendmail*
S85gpm -> ../init.d/gpm*
S85httpd -> ../init.d/httpd*
S85postgresql -> ../init.d/postgresql*
S85sound -> ../init.d/sound*
S90xfs -> ../init.d/xfs*
S91smb -> ../init.d/smb*
S95innd -> ../init.d/innd*
S99linuxconf -> ../init.d/linuxconf
S99local -> ../rc.local*

rc4.d/

K08autofs -> ../init.d/autofs*
K20bootparamd -> ../init.d/bootparamd*
K20rstatd -> ../init.d/rstatd*
K20rusersd -> ../init.d/rusersd*
K20rwalld -> ../init.d/rwalld*
K20rwhod -> ../init.d/rwhod*
K25squid -> ../init.d/squid*
K30mcserv -> ../init.d/mcserv*
K34yppasswdd -> ../init.d/yppasswdd*
K50snmpd -> ../init.d/snmpd*
K55routed -> ../init.d/routed*
K60mars-nwe -> ../init.d/mars-nwe*
K75gated -> ../init.d/gated*
K80nscd -> ../init.d/nscd*
K88ypserv -> ../init.d/ypserv*
S05apmd -> ../init.d/apmd*
S10network -> ../init.d/network*
S11portmap -> ../init.d/portmap*
S15netfs -> ../init.d/netfs*
S20random -> ../init.d/random*
S30syslog -> ../init.d/syslog*
S40atd -> ../init.d/atd*
S40crond -> ../init.d/crond*
S45pcmcia -> ../init.d/pcmcia*
S50inet -> ../init.d/inet*
S55named -> ../init.d/named*
S60lpd -> ../init.d/lpd*
S60nfs -> ../init.d/nfs*
S65dhcpd -> ../init.d/dhcpd*
S72amd -> ../init.d/amd*
S75keytable -> ../init.d/keytable*
S80sendmail -> ../init.d/sendmail*
S85gpm -> ../init.d/gpm*
S85httpd -> ../init.d/httpd*
S85sound -> ../init.d/sound*
S90xfs -> ../init.d/xfs*
S95innd -> ../init.d/innd*
S99linuxconf -> ../init.d/linuxconf

rc5.d/

K08autofs -> ../init.d/autofs*

K10xntpd -> ../init.d/xntpd*
K20bootparamd -> ../init.d/bootparamd*
K20rstatd -> ../init.d/rstatd*
K20rusersd -> ../init.d/rusersd*
K20rwalld -> ../init.d/rwalld*
K20rwhod -> ../init.d/rwhod*
K25squid -> ../init.d/squid*
K30mcserv -> ../init.d/mcserv*
K34yppasswdd -> ../init.d/yppasswdd*
K45arpwatch -> ../init.d/arpwatch*
K55routed -> ../init.d/routed*
K60mars-nwe -> ../init.d/mars-nwe*
K75gated -> ../init.d/gated*
K80nscd -> ../init.d/nscd*
K88ypserv -> ../init.d/ypserv*
K96pcmcia -> ../init.d/pcmcia*
S05apmd -> ../init.d/apmd*
S10network -> ../init.d/network*
S11portmap -> ../init.d/portmap*
S15netfs -> ../init.d/netfs*
S20random -> ../init.d/random*
S30syslog -> ../init.d/syslog*
S40atd -> ../init.d/atd*
S40crond -> ../init.d/crond*
S50inet -> ../init.d/inet*
S50snmpd -> ../init.d/snmpd*
S55named -> ../init.d/named*
S60lpd -> ../init.d/lpd*
S60nfs -> ../init.d/nfs*
S65dhcpd -> ../init.d/dhcpd*
S72amd -> ../init.d/amd*
S75keytable -> ../init.d/keytable*
S80sendmail -> ../init.d/sendmail*
S85gpm -> ../init.d/gpm*
S85httpd -> ../init.d/httpd*
S85postgresql -> ../init.d/postgresql*
S85sound -> ../init.d/sound*
S90xfs -> ../init.d/xfs*
S91smb -> ../init.d/smb*
S95innd -> ../init.d/innd*
S99linuxconf -> ../init.d/linuxconf
S99local -> ../rc.local*

rc6.d/

K00linuxconf -> ../init.d/linuxconf
K05innd -> ../init.d/innd*
K05keytable -> ../init.d/keytable*
K08autofs -> ../init.d/autofs*
K10xfs -> ../init.d/xfs*
K15gpm -> ../init.d/gpm*
K15httpd -> ../init.d/httpd*
K15sound -> ../init.d/sound*
K20bootparamd -> ../init.d/bootparamd*
K20nfs -> ../init.d/nfs*
K20rstatd -> ../init.d/rstatd*
K20rusersd -> ../init.d/rusersd*
K20rwalld -> ../init.d/rwalld*

K20rwhod -> ../init.d/rwhod*
K25squid -> ../init.d/squid*
K28amd -> ../init.d/amd*
K30mcserv -> ../init.d/mcserv*
K30sendmail -> ../init.d/sendmail*
K34yppasswdd -> ../init.d/yppasswdd*
K35dhcpd -> ../init.d/dhcpd*
K45named -> ../init.d/named*
K50inet -> ../init.d/inet*
K50snmpd -> ../init.d/snmpd*
K55routed -> ../init.d/routed*
K60atd -> ../init.d/atd*
K60crond -> ../init.d/crond*
K60lpd -> ../init.d/lpd*
K60mars-nwe -> ../init.d/mars-nwe*
K75gated -> ../init.d/gated*
K80nscd -> ../init.d/nscd*
K80random -> ../init.d/random*
K85netfs -> ../init.d/netfs*
K88ypserv -> ../init.d/ypserv*
K89portmap -> ../init.d/portmap*
K90killall -> ../init.d/killall*
K90network -> ../init.d/network*
K92apmd -> ../init.d/apmd*
K96pcmcia -> ../init.d/pcmcia*
K99syslog -> ../init.d/syslog*
S00reboot -> ../init.d/halt*

## Glossary

Some terms and features are best understood when referenced under various contexts. To that end, the glossary contains extracts from various references. Read on...

| Term | Reference | Text |
|---|---|---|
| Run Levels | <3> | "..... But getting the kernel fully loaded and running isn't the end of the boot process; it's just the first stage (sometimes calledrun level 1).<br><br>The kernel's next step is to check to make sure your disks are OK. Disk file systems are fragile things; if they've been damaged by a hardware failure or a sudden power outage, there are good reasons to take recovery steps before your Unix is all the way up. We'll go into some of this later on when we talk about how file systems can go wrong.<br><br>The kernel's next step is to start several daemons. A daemon is a program like a print spooler, a mail listener or a WWW server that lurks in the background, waiting for things to do. These special programs often have to coordinate several requests that could conflict. They are daemons because it's often easier to write one program that runs constantly and knows about all requests than it would be to try to make sure that a flock of copies (each processing one request and all running at the same time) don't step on each other. The particular collection of daemons your system starts may vary, but will almost always include a print spooler (a gatekeeper daemon for your printer).<br><br>Once all daemons are started, we're at run level 2. The next step is to prepare for users. The kernel starts a copy of a program called getty to watch your console (and maybe more copies to watch dial-in serial ports). This program is what issues the login prompt to your console. We're now at run level 3 and ready for you to log in and run programs.<br><br>When you log in (give a name and password) you identify yourself to getty and the computer. It then runs a program called (naturally enough) login, which does some housekeeping things and then starts up a command |

| | | |
|---|---|---|
| | | interpreter, the shell. (Yes, getty and login could be one program. They're separate for historical reasons not worth going into here.)<br><br>In the next section, we'll talk about what happens when you run programs from the shell.<br>" |
| | <4> | "Briefly, init steps through a series of run levels, which correspond to various operationing states of the system. Run level 1 is entered immediately after the system boots, run levels 2 and 3 are the normal, multiuser operation modes of the system, run level 4 starts the X Window System via the X display manager xdm, and run level 6 reboots the system. The run level(s) associated with each command are the second item in each line of the /etc/inittab file.<br>" |
| | <2> | "Init(8) distinguishes multiple runlevels, each of which can have its own set of processes that are started. Valid runlevels are 0-6 plus A, B, and C for ondemand entries." |
| | <2> | "The runlevels field may contain multiple characters for different runlevels. For example, 123 specifies that the process should be started in runlevels 1, 2, and 3. The runlevels for ondemand entries may contain an A, B, or C. The runlevels field of sysinit, boot, and bootwait entries are ignored. When the system runlevel is changed, any running processes that are not specified for the new runlevel are killed, first with SIGTERM, then with SIGKILL." |
| | <6> | A runlevel is a software configuration of the system which allows only a selected group of processes to exist. The processes spawned by init for each of these runlevels are defined in the /etc/inittab file. Init can be in one of eight runlevels: 0-6 and S or s. The runlevel is changed by having a privileged user run telinit, which sends appropriate signals to init, telling it which runlevel to change to.<br><br>Runlevels 0, 1, and 6 are reserved. Runlevel 0 is used to halt the system, runlevel 6 is used to reboot the system, and runlevel 1 is used to get the system down into single user mode. Runlevel S is not really meant to be used directly, but more for the scripts that are executed when entering runlevel 1. For more information on this, see the manpages for shutdown(8) and inittab(5).<br><br>Runlevels 7-9 are also valid, though not really documented. This is because traditional" Unix variants don't use them. In case you're curious, runlevels S and s are in fact the same. Internally they are aliases for the same runlevel - this is just a leftover from the systems the author used to use when writing sysvinit. |