Server Operating Systems

Lecture 8
Startup and Initialisation

Startup and Initialisation

- Booting Linux
- GRUB
- The Kernel
- The init Process
- Startup Scripts
- Run Levels
- init config and rc

Booting Linux

PC's tend to to have a two-stage loading process:

- The BIOS loads a small boot loader program.
- This program loads the operating system kernel.

Most Linux distributions use one of the following boot loaders:

- LILO
- GRUB - A more flexible boot loader than LILO
- LOADLIN - Can be used to boot from an MS-DOS command line

They present the user with a menu of operating systems present on the computer, and wait for one to be chosen.

GRUB

SUSE Linux installs GRUB by default.

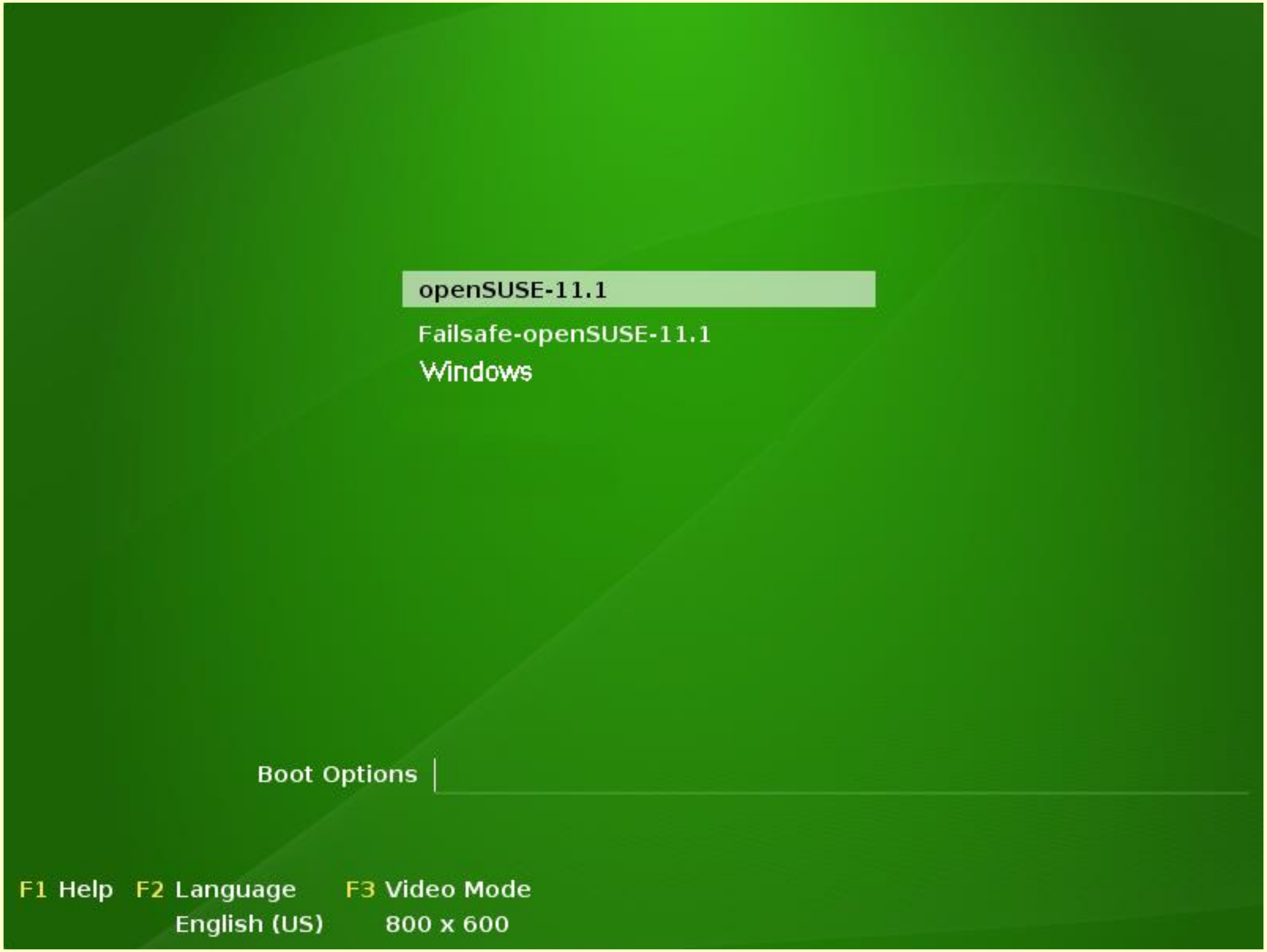
GRUB could be installed in one of the following locations:

1. The MBR of the main hard disk drive
   - This means that it would be the first thing the user sees.

2. The boot record of the Linux root partition
   - Another boot loader could be installed in the MBR, which would redirect to GRUB if that menu option was selected.

3. On a floppy disk
   - The system would only load GRUB (and therefore Linux) if you had the floppy in the drive and the floppy drive was first in the boot order.

SUSE GRUB Boot Menu



openSUSE-11.1

Failsafe-openSUSE-11.1

Windows

Boot Options

F1 Help F2 Language English (US) F3 Video Mode 800 x 600

SUSE GRUB Config File

Located at: /boot/grub/menu.lst

```
# Modified by YaST2. Last modification on Sat Sep 27 16:09:22 UTC 2008
default 0
timeout 8
gfxmenu (hd0,5)/boot/message

###Don't change this comment - YaST2 identifier: Original name: linux###
title openSUSE 11.1
      root (hd0,5)
      kernel /boot/vmlinuz-2.6.22.5-31-default
root=/dev/disk/by-id/scsi-SATA_ST3120026A_3LJ18NDS-part6 vga=0x317
resume=/dev/sda8 splash=silent showopts
      initrd /boot/initrd-2.6.22.5-31-default

###Don't change this comment - YaST2 identifier: Original name: failsafe###
title Failsafe -- openSUSE 11.1
      root (hd0,5)
      kernel /boot/vmlinuz-2.6.22.5-31-default
root=/dev/disk/by-id/scsi-SATA_ST3120026A_3LJ18NDS-part6 vga=normal showopts ide=nodma
apm=off acpi=off noresume nosmp noapic maxcpus=0 edd=off 3
      initrd /boot/initrd-2.6.22.5-31-default

###Don't change this comment - YaST2 identifier: Original name: windows###
title Windows
      rootnoverify (hd0,5)
      chainloader (hd0,0)+1
```

The Kernel

The Kernel is the program at the heart of the operating system.

Most Linux distributions use a standard name:

vmlinuz followed by a version number.

eg **`vmlinuz-2.6.22.5-31-default`**

Hardware Initialisation

After the kernel is brought in from disk into main memory, it begins execution and one of the first things it does is initialize the system's hardware.

All those cryptic messages you see fly by when the Linux kernel first starts up are messages from the compiled-in kernel drivers initializing and configuring your hardware.

Sometimes the kernel needs help in configuring your hardware. Information such as IRQ, DMA, and I/O base addresses need to be specified to the kernel.

With Linux these can be specified via it's "command line".

```
vga=normal showopts ide=nodma apm=off acpi=off
noresume nosmp noapic maxcpus=0 edd=off 3
```

The init Process

The first program the kernel attempts to execute after basic system initialization is complete is called **init**.

The **init** process is the mother of all processes running on a UNIX system. If this process dies, so does the system.

init’s job after basic hardware initialization is complete is to take over the system start-up procedure and complete the system bootstrap process.

It does this by running a set of initialisation scripts that are usually stored in /etc or /sbin

The initialisation scripts carry out the following tasks:

- Check the integrity of the filesystems, using fsck.
- Mounting local disks
- Designating paging areas
- Performing filesystem cleanup activities
- Starting system daemons (services) for printing, mail, error logging etc.
- Starting networking daemons and mounting remote disks
- Enabling user logins, usually by starting getty processes.

The init Process

In general there are two styles of system start-up scripts:

- The System V style: used by most UNIX and UNIX-like operating systems today including Solaris, HP-UX and most distributions of Linux.
- The BSD style: used by the various flavors of BSD (FreeBSD, NetBSD, OpenBSD, BSDi) and Slackware Linux.

This lecture will describe the System V style.

Startup Scripts

All start-up scripts are typically kept in a directory named init.d which usually lives somewhere under “/etc”.

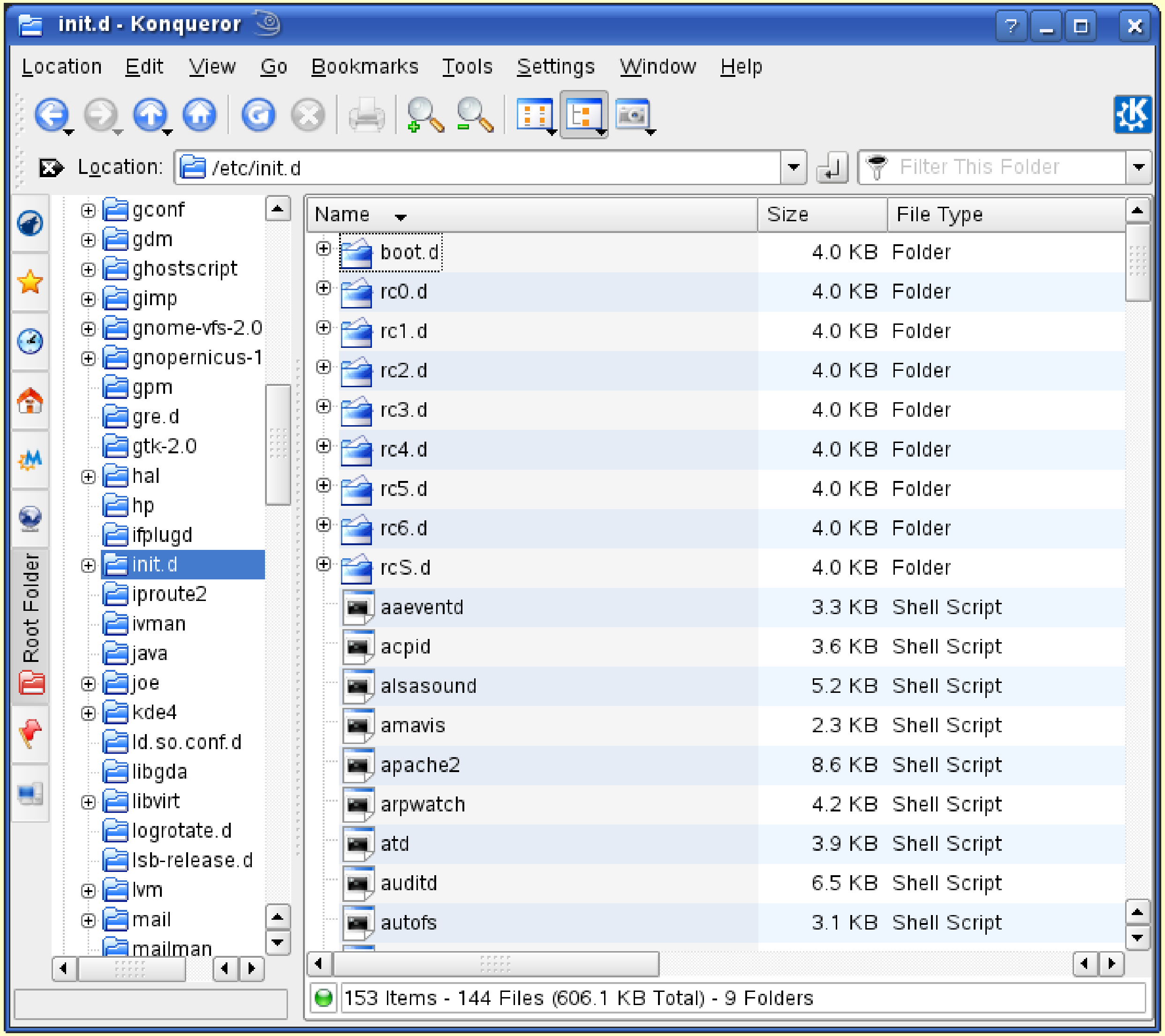
- Red Hat Linux places this directory under “/etc/rc.d”.
- HP-UX places this directory under “/sbin”.
- SUSE places this directory in "/etc"

Each start-up script can usually accept at least two command line arguments: **start** and **stop**.

- **start** tells the script to start whatever it is that script is responsible for.
- **stop** tells the script to stop whatever it is that script is responsible for.
- eg `rcapache2 start` or `rcapache stop`

Most of the time, you are not starting the executable directly, you are running a script which starts the executable.

/etc/init.d



Location: /etc/init.d

- gconf
- gdm
- ghostscript
- gimp
- gnome-vfs-2.0
- gnopernicus-1
- gpm
- gre.d
- gtk-2.0
- hal
- hp
- ifplugd
- init.d
- iproute2
- ivman
- java
- joe
- kde4
- ld.so.conf.d
- libgda
- libvirt
- logrotate.d
- lsb-release.d
- lvm
- mail
- mailman

Name	Size	File Type
boot.d	4.0 KB	Folder
rc0.d	4.0 KB	Folder
rc1.d	4.0 KB	Folder
rc2.d	4.0 KB	Folder
rc3.d	4.0 KB	Folder
rc4.d	4.0 KB	Folder
rc5.d	4.0 KB	Folder
rc6.d	4.0 KB	Folder
rcS.d	4.0 KB	Folder
aaeventd	3.3 KB	Shell Script
acpid	3.6 KB	Shell Script
alsasound	5.2 KB	Shell Script
amavis	2.3 KB	Shell Script
apache2	8.6 KB	Shell Script
arpwatch	4.2 KB	Shell Script
atd	3.9 KB	Shell Script
auditd	6.5 KB	Shell Script
autofs	3.1 KB	Shell Script

153 Items - 144 Files (606.1 KB Total) - 9 Folders

A Typical Script - cron

```
if [ -x /usr/sbin/cron ]; then
   if [ "`ps -ef | grep -c [c]ron `" = "0" ]; then
      rm -f /etc/cron.d/FIFO
      /usr/sbin/cron
      echo "cron started."
   else
      echo "cron is already running."
   fi
fi
```

Makes sure that the cron executable exists

Checks to see if cron process is already running

If not, deletes FIFO file, starts executable and displays message.

Otherwise, displays suitable message.

Run Levels

Most computer systems can be in one of three states:

- Off
- Single-user mode
- Multi-user mode.

System V Unix takes this to extremes. There are 7 standard run levels:

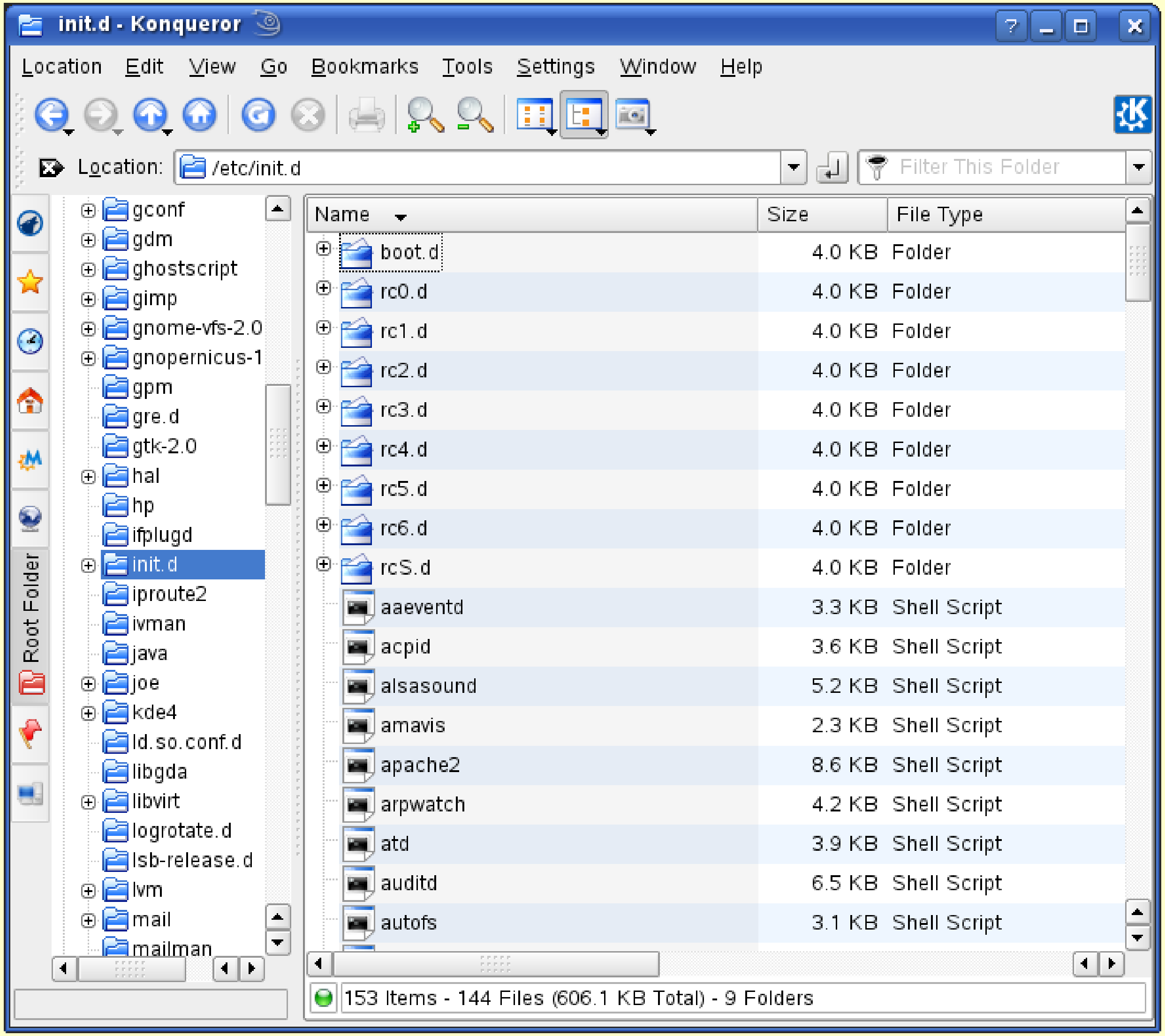
– 0	Power off state. Ready to turn off.
– 1 or S	Single-user mode.
– 2	Multi-user mode
– 3	Remote file sharing - used for Server systems.
– 4	Undefined
– 5	Maintenance / safe mode
– 6	Shutdown and reboot

- Each run-level gets its own directory

- This directory follows the naming convention of rc*n*.d where *n* is the run-level
  - i.e. scripts for run-level 2 would be found under a directory named rc2.d.

- This directory contains scripts which are executed when that run-level is entered.
  - While this directory can contain actual scripts, it usually consists of symbolic links to the scripts in the init.d directory.

/etc/init.d



/etc/init.d

gconf
gdm
ghostscript
gimp
gnome-vfs-2.0
gnopernicus-1
gpm
gre.d
gtk-2.0
hal
hp
ifplugd
init.d
iproute2
ivman
java
joe
kde4
ld.so.conf.d
libgda
libvirt
logrotate.d
lsb-release.d
lvm
mail
mailman

Name	Size	File Type
boot.d	4.0 KB	Folder
rc0.d	4.0 KB	Folder
rc1.d	4.0 KB	Folder
rc2.d	4.0 KB	Folder
rc3.d	4.0 KB	Folder
rc4.d	4.0 KB	Folder
rc5.d	4.0 KB	Folder
rc6.d	4.0 KB	Folder
rcS.d	4.0 KB	Folder
aaeventd	3.3 KB	Shell Script
acpid	3.6 KB	Shell Script
alsasound	5.2 KB	Shell Script
amavis	2.3 KB	Shell Script
apache2	8.6 KB	Shell Script
arpwatch	4.2 KB	Shell Script
atd	3.9 KB	Shell Script
auditd	6.5 KB	Shell Script
autofs	3.1 KB	Shell Script

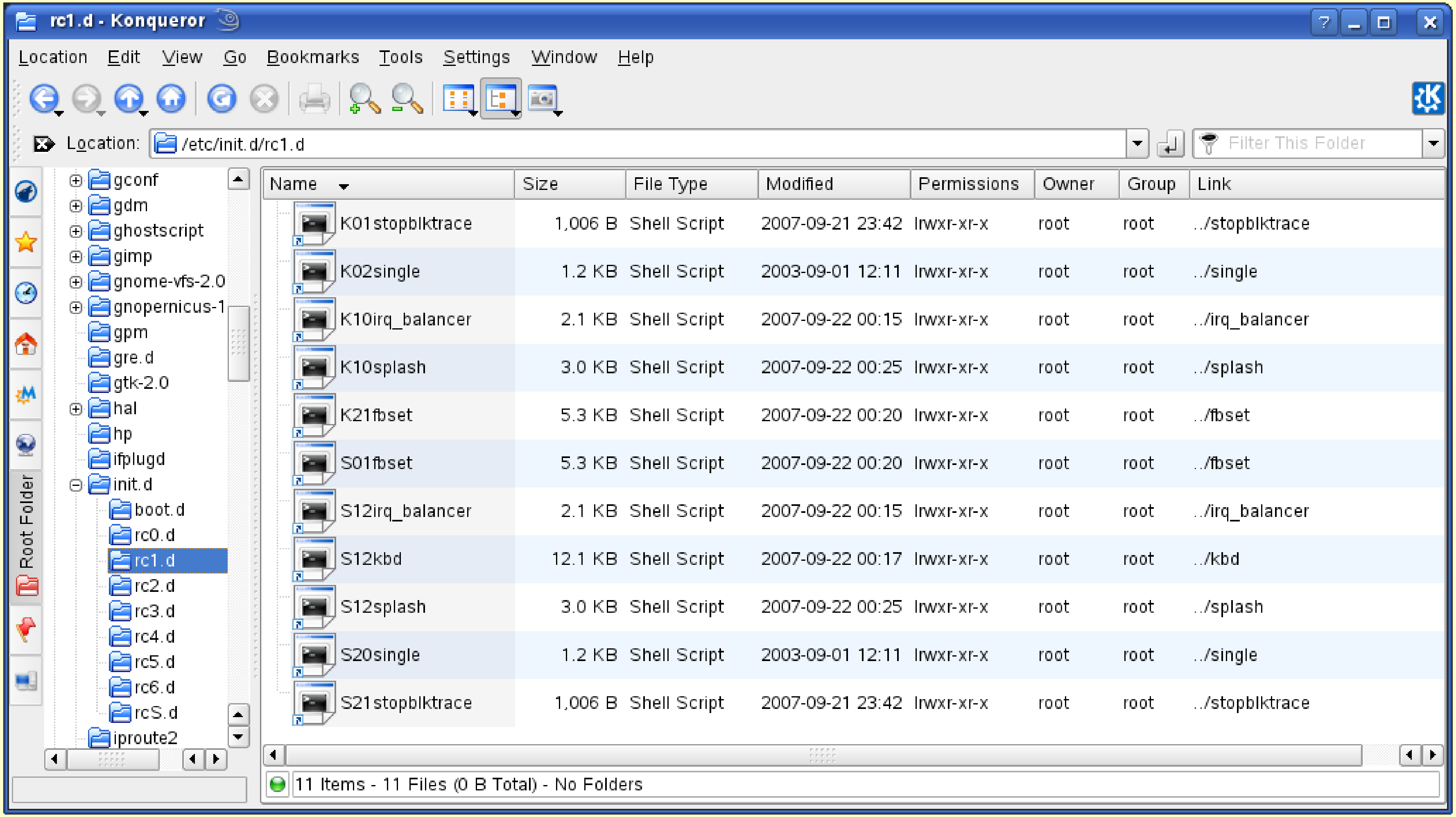
153 Items - 144 Files (606.1 KB Total) - 9 Folders

Scripts in the run-level directory are executed in alphanumeric order

If the script name begins with an “S”, the script is passed the “start” command line parameter.

If it begins with a “K”, it is passed the “stop” command line parameter.

Example - /etc/init.d/rc1.d



Location: /etc/init.d/rc1.d

Name	Size	File Type	Modified	Permissions	Owner	Group	Link
K01stopblktrace	1,006 B	Shell Script	2007-09-21 23:42	lrwxr-xr-x	root	root	../stopblktrace
K02single	1.2 KB	Shell Script	2003-09-01 12:11	lrwxr-xr-x	root	root	../single
K10irq_balancer	2.1 KB	Shell Script	2007-09-22 00:15	lrwxr-xr-x	root	root	../irq_balancer
K10splash	3.0 KB	Shell Script	2007-09-22 00:25	lrwxr-xr-x	root	root	../splash
K21fbset	5.3 KB	Shell Script	2007-09-22 00:20	lrwxr-xr-x	root	root	../fbset
S01fbset	5.3 KB	Shell Script	2007-09-22 00:20	lrwxr-xr-x	root	root	../fbset
S12irq_balancer	2.1 KB	Shell Script	2007-09-22 00:15	lrwxr-xr-x	root	root	../irq_balancer
S12kbd	12.1 KB	Shell Script	2007-09-22 00:17	lrwxr-xr-x	root	root	../kbd
S12splash	3.0 KB	Shell Script	2007-09-22 00:25	lrwxr-xr-x	root	root	../splash
S20single	1.2 KB	Shell Script	2003-09-01 12:11	lrwxr-xr-x	root	root	../single
S21stopblktrace	1,006 B	Shell Script	2007-09-21 23:42	lrwxr-xr-x	root	root	../stopblktrace

11 Items - 11 Files (0 B Total) - No Folders

init configuration and rc

SysV **init**'s configuration file is **`/etc/inittab`**

This file tells **init** what script it should run for each run-level.

init executes a master control script called **rc**, which is supplied with the run level number

rc then executes all of the scripts in that run-level's script directory.

For example, for run-level 2, **init** will execute the script

```
/etc/init.d/rc 2
```

This script in turn would execute every script in run-level 2's script directory **`/etc/rc2.d`**

Extract from /etc/inittab

```
# The default runlevel is defined here
id:5:initdefault:

# First script to be executed, if not booting in emergency mode
si::bootwait:/etc/init.d/boot

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

# what to do in single-user mode
ls:S:wait:/etc/init.d/rc S
~~:S:respawn:/sbin/sulogin

# what to do when CTRL-ALT-DEL is pressed
ca::ctrlaltdel:/sbin/shutdown -r -t 4 now
```

Single User Mode

On some systems, single-user mode is a special administrative mode that usually starts the system with a minimal configuration.

Other systems use run-level 5 to perform the same function.

For example, no system daemons are started and extra filesystems may not be mounted.

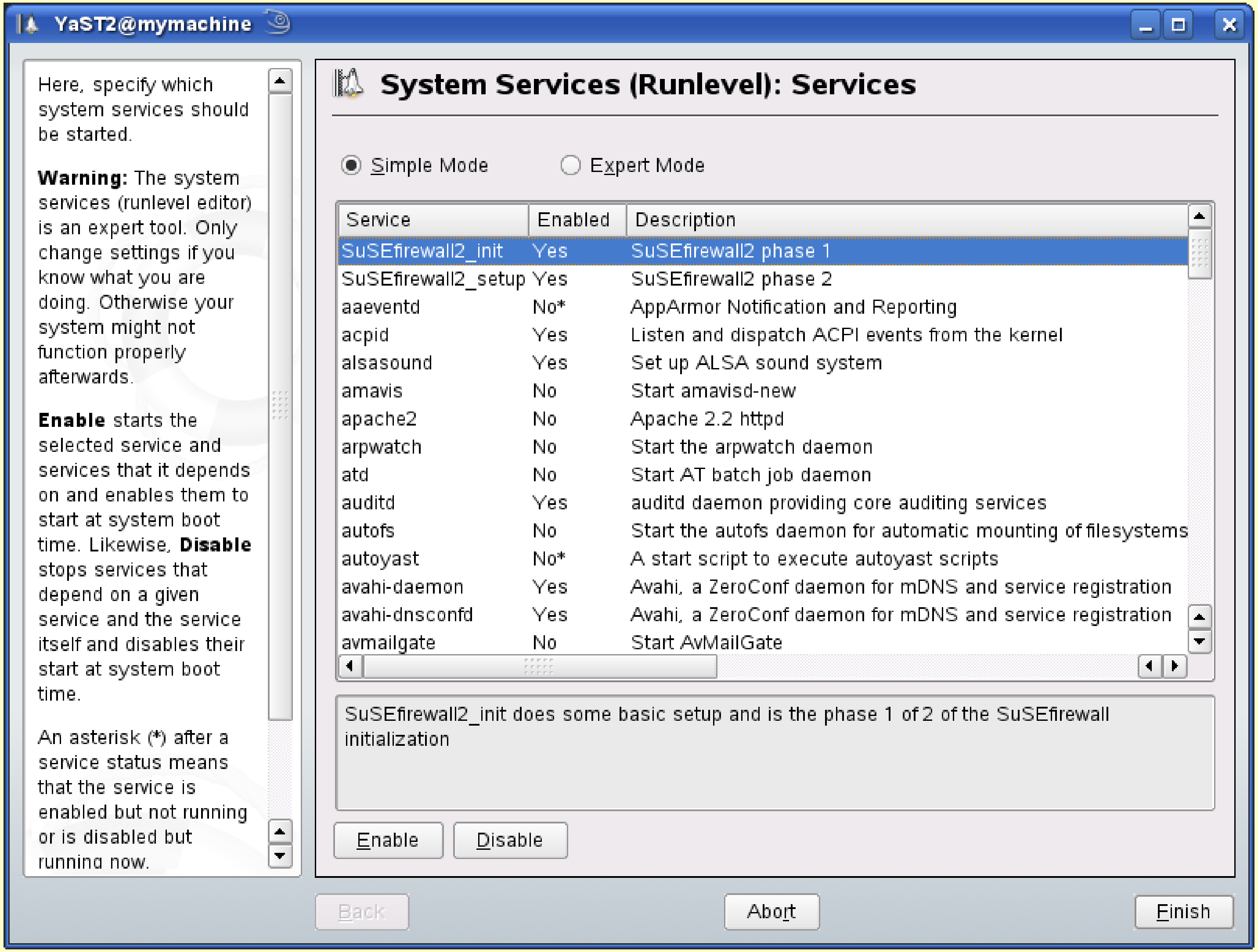
Single-user mode is typically used to repair a broken system such as running fsck to check a sick filesystem.

On Sys V systems, single user mode can also be entered by telling **init** to enter run-level 1 or S.

This can be done via the **telinit** command.

```
telinit 1
```

Yast2 > System > System Services (Runlevel)



Here, specify which system services should be started.

Warning: The system services (runlevel editor) is an expert tool. Only change settings if you know what you are doing. Otherwise your system might not function properly afterwards.

Enable starts the selected service and services that it depends on and enables them to start at system boot time. Likewise, **Disable** stops services that depend on a given service and the service itself and disables their start at system boot time.

An asterisk (*) after a service status means that the service is enabled but not running or is disabled but running now.

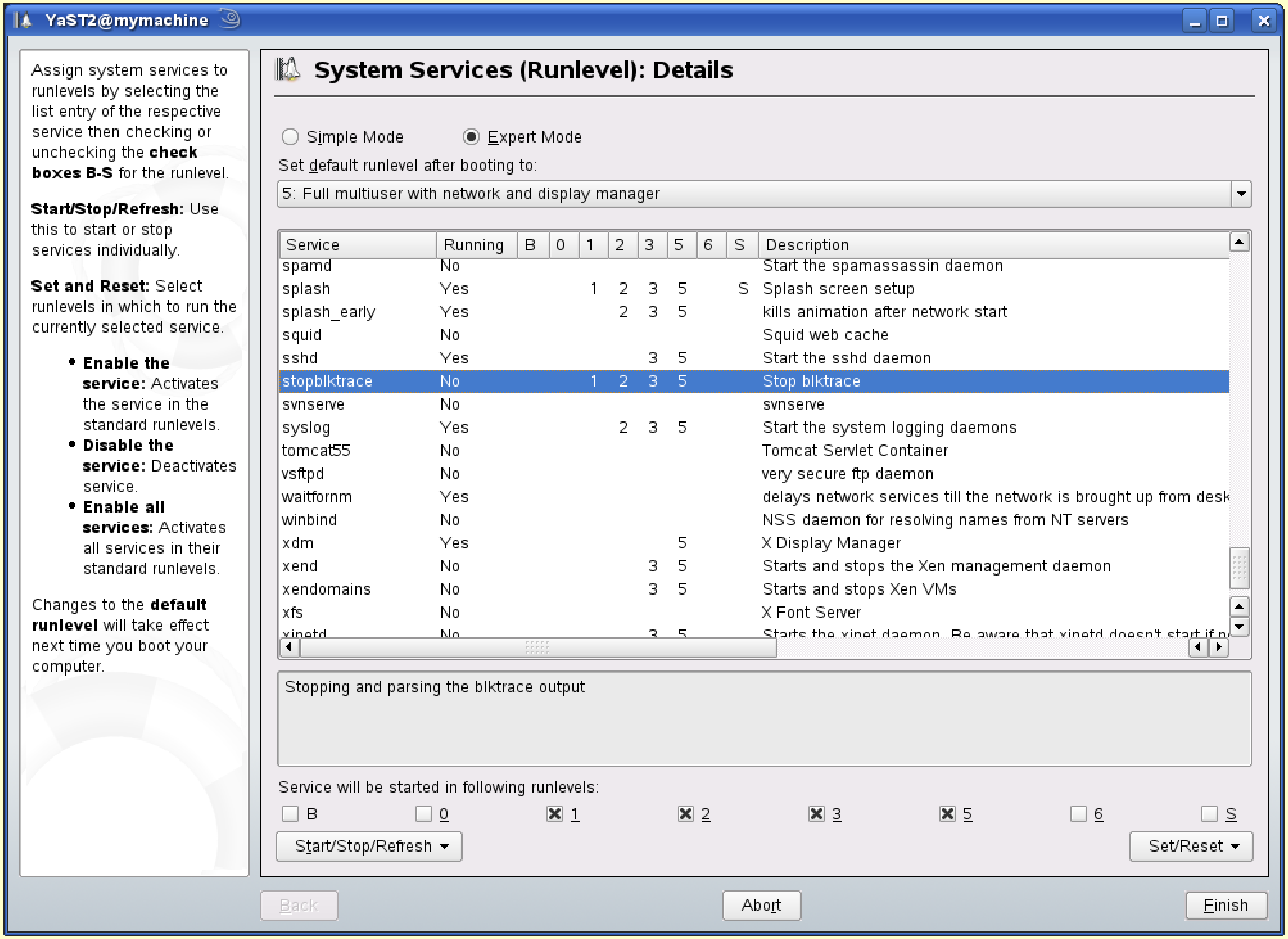
System Services (Runlevel): Services

● Simple Mode ○ Expert Mode

Service	Enabled	Description
SuSEfirewall2_init	Yes	SuSEfirewall2 phase 1
SuSEfirewall2_setup	Yes	SuSEfirewall2 phase 2
aaeventd	No*	AppArmor Notification and Reporting
acpid	Yes	Listen and dispatch ACPI events from the kernel
alsasound	Yes	Set up ALSA sound system
amavis	No	Start amavisd-new
apache2	No	Apache 2.2 httpd
arpwatch	No	Start the arpwatch daemon
atd	No	Start AT batch job daemon
auditd	Yes	auditd daemon providing core auditing services
autofs	No	Start the autofs daemon for automatic mounting of filesystems
autoyast	No*	A start script to execute autoyast scripts
avahi-daemon	Yes	Avahi, a ZeroConf daemon for mDNS and service registration
avahi-dnsconfd	Yes	Avahi, a ZeroConf daemon for mDNS and service registration
avmailgate	No	Start AvMailGate

SuSEfirewall2_init does some basic setup and is the phase 1 of 2 of the SuSEfirewall initialization

Expert Mode



Assign system services to runlevels by selecting the list entry of the respective service then checking or unchecking the **check boxes B-S** for the runlevel.

Start/Stop/Refresh: Use this to start or stop services individually.

Set and Reset: Select runlevels in which to run the currently selected service.

- **Enable the service:** Activates the service in the standard runlevels.
- **Disable the service:** Deactivates service.
- **Enable all services:** Activates all services in their standard runlevels.

Changes to the **default runlevel** will take effect next time you boot your computer.

System Services (Runlevel): Details

○ Simple Mode ● Expert Mode

Set default runlevel after booting to: 5: Full multiuser with network and display manager

Service	Running	B	0	1	2	3	5	6	S	Description
spamd	No									Start the spamassassin daemon
splash	Yes			1	2	3	5		S	Splash screen setup
splash_early	Yes				2	3	5			kills animation after network start
squid	No									Squid web cache
sshd	Yes					3	5			Start the sshd daemon
stopblktrace	No			1	2	3	5			Stop blktrace
svnserve	No									svnserve
syslog	Yes				2	3	5			Start the system logging daemons
tomcat55	No									Tomcat Servlet Container
vsftpd	No									very secure ftp daemon
waitfornm	Yes									delays network services till the network is brought up from desk
winbind	No									NSS daemon for resolving names from NT servers
xdm	Yes						5			X Display Manager
xend	No					3	5			Starts and stops the Xen management daemon
xendomains	No					3	5			Starts and stops Xen VMs
xfs	No									X Font Server
xinetd	No					3	5			Starts the xinet daemon. Be aware that xinetd doesn't start if n

Stopping and parsing the blktrace output

Service will be started in following runlevels:

☐ B ☐ 0 ☒ 1 ☒ 2 ☒ 3 ☒ 5 ☐ 6 ☐ S

Shutting the system Down

UNIX systems have to be gracefully powered down. You cannot just shut the system off. This can damage the system.

The typical way to shutdown the UNIX system is to use the **shutdown** command.

shutdown allows the system administrator to broadcast a message to all currently logged in users that the system is about to be shutdown.

```
shutdown -h +5 "Shutting system down in five minutes"
```

Most versions of Linux also have the following commands.

- halt
- reboot
- poweroff

Practical

Look at Yast 2 and look at the Bootloader config option

Have a look at your GRUB config file

Have a look at etc/init.d and look inside some of the folders.

Have a look at etc/inittab

Have a look at Yast2 > System Services