

information technology & management

viabilityit

INTRO TO OPEN SOURCE
OPERATING SYSTEMS

ILLINOIS INSTITUTE OF TECHNOLOGY

ITMO456

System Initialization and X Windows

Sean Hughes-Durkin

ITMO/IT-O 456 Fall 2017

Information Technology & Management
Programs

School of Applied Technology

Objectives

At the end of this lesson students should be able to:

- Summarize the major steps necessary to boot a Linux system
- Detail the configuration of common Linux boot loaders
- Explain the UNIX SysV and Systemd system initialization processes
- Start, stop, and restart daemons

Objectives

At the end of this lesson students should be able to:

- Configure the system to start and stop daemons upon entering certain runlevels and targets
- Explain the purpose of the major Linux GUI components
 - X Windows
 - Window manager
 - Desktop environment

Objectives

At the end of this lesson students should be able to:

- List common window managers and desktop environments used in Linux
- Configure X Windows settings

The Boot Process

- ◆ Power On Self Test (POST)
 - Series of tests run when computer is turned on
 - Ensures functionality of hardware
- ◆ Master Boot Record (MBR)
 - Defines partitions and boot loader
 - Normally located on first HDD sector
- ◆ Boot loader
 - Program used to load an operating system

The Boot Process

- ◆ Active partition
 - Partition that the MBR points to
 - One per HDD
- ◆ MBR might contain pointer to a partition containing a boot loader on the first sector

The Boot Process

◆ `/boot`

- Directory that contains the kernel and boot-related files

◆ `vmlinuz-<kernel version>`

- The Linux kernel file

◆ `initramfs-<kernel version>`

- Contains the tools and scripts needed to mount the file systems before the `init` binary on the real root file system is called

The Boot Process

```
[root@itmo456 ~]# ll -h /boot/initramfs-* /boot/vmlinuz-*
-rw-----, 1 root root 38M Sep 20 11:34 /boot/initramfs-0-rescue-ac106f57b96b43478e83ed5079be578e.img
-rw-----, 1 root root 12M Sep 20 11:34 /boot/initramfs-3.11.10-301.fc20.x86_64.img
-rw-----, 1 root root 17M Sep 20 16:12 /boot/initramfs-3.19.8-100.fc20.x86_64.img
-rwxr-xr-x, 1 root root 5.0M Sep 20 11:34 /boot/vmlinuz-0-rescue-ac106f57b96b43478e83ed5079be578e
-rwxr-xr-x, 1 root root 5.0M Dec 5 2013 /boot/vmlinuz-3.11.10-301.fc20.x86_64
-rwxr-xr-x, 1 root root 5.6M May 12 12:14 /boot/vmlinuz-3.19.8-100.fc20.x86_64
[root@itmo456 ~]#
```


The Boot Process

◆ Daemon

- **D**isk and **e**xecution **m**onitor
- System process that performs useful tasks (often in the background)
- e.g., proxy, VPN, printing, etc

◆ Init (initialize) daemon

- First process started by Linux kernel
- Loads all other daemons
- Brings system to usable state

The Boot Process

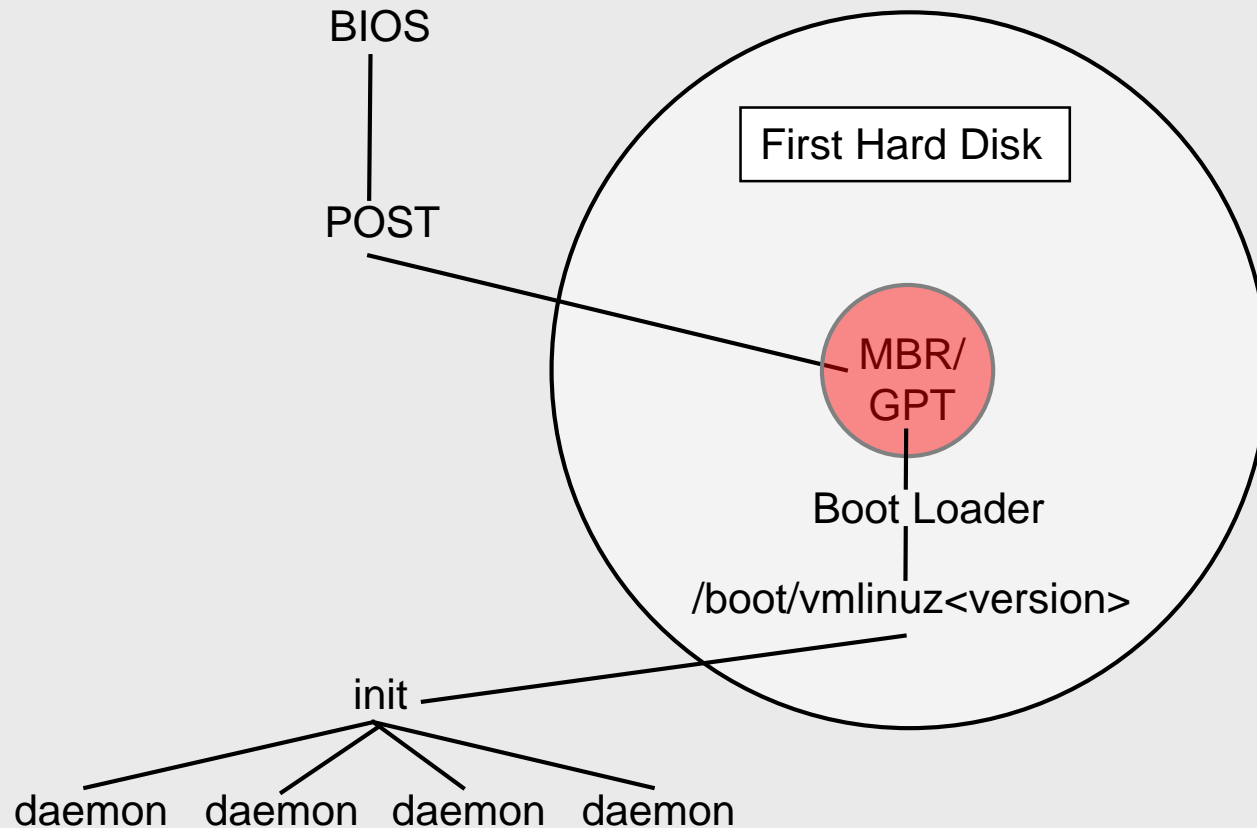


Figure 8-1: The boot process

Boot Loaders

- ◆ Primary function
 - Load the Linux kernel into memory
- ◆ Other functions
 - Pass information to kernel
 - Boot another OS
- ◆ Normal Linux boot loader:
 - GRand Unified Boot loader (GRUB)
 - Older (rarely found): Linux Loader (LILO)

Boot Loaders

- ◆ Boot loaders are typically loaded by the system BIOS from the MBR or the first sector of the active partition of a hard disk

GRUB

- ◆ GRand Unified Bootloader (**GRUB**)
 - Resembles common UNIX boot loaders
 - More recent than the LILO boot loader
 - Only boot loader supplied with Fedora and most current Linux distributions
- ◆ To view GRUB command line press c at the kernel menu
 - Should display automatically if multiple OSes or multiple kernels are installed

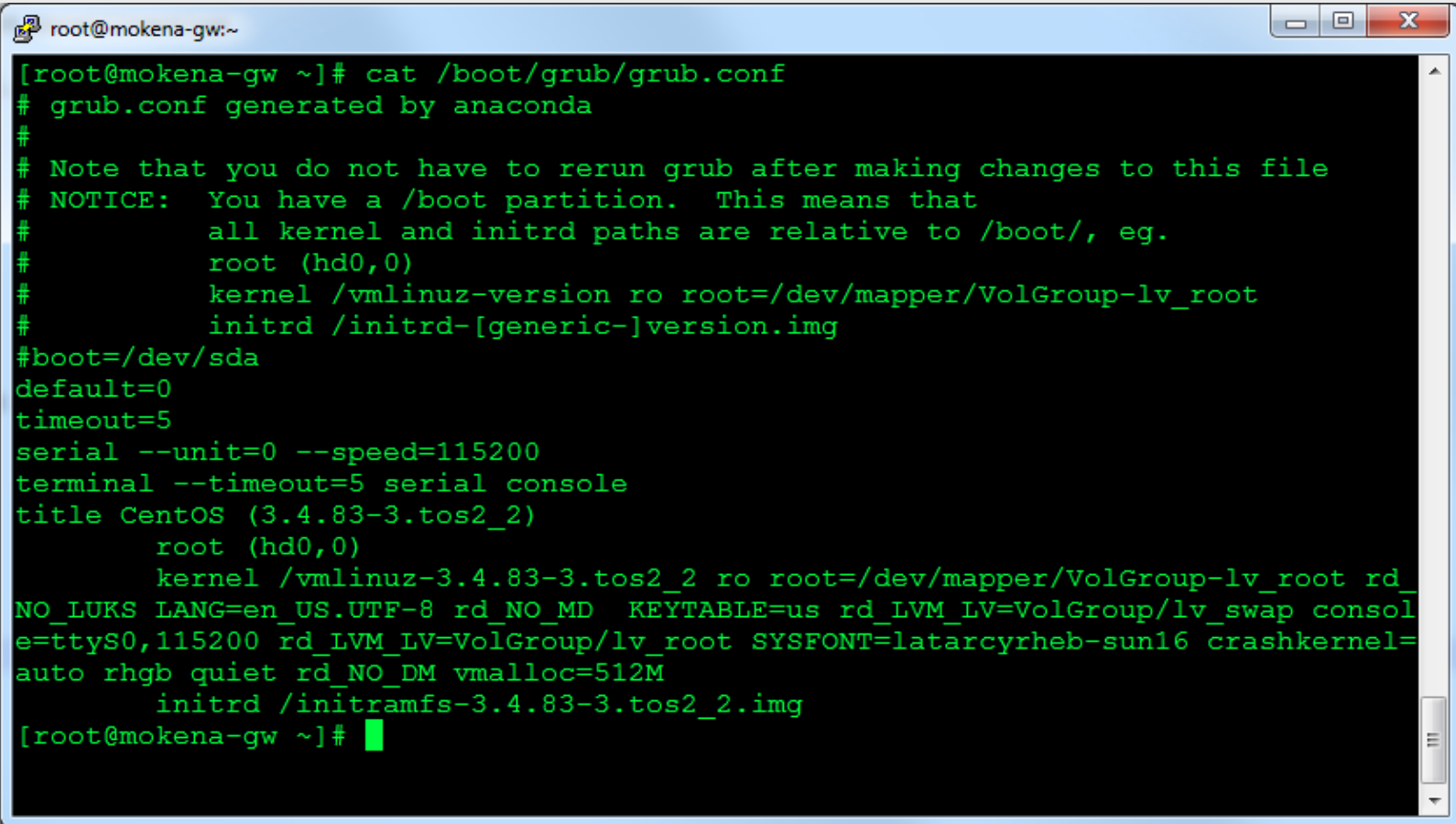
GRUB

- ◆ Stage 1
 - First major part of GRUB
 - Typically resides on MBR
 - Points to Stage 1.5
- ◆ Stage1.5
 - Loads filesystem support
 - Resides in `/boot/grub`
- ◆ Stage2
 - Performs boot loader functions
 - Displays graphical boot loader screen

GRUB

- ◆ To configure, edit
`/boot/grub/grub.conf`
or
`/boot/grub/menu.lst`
 - Read directly by Stage 2 boot loader
 - HDDs & partitions identified by numbers
 - Format: (hd<drive#>,<partition#>)

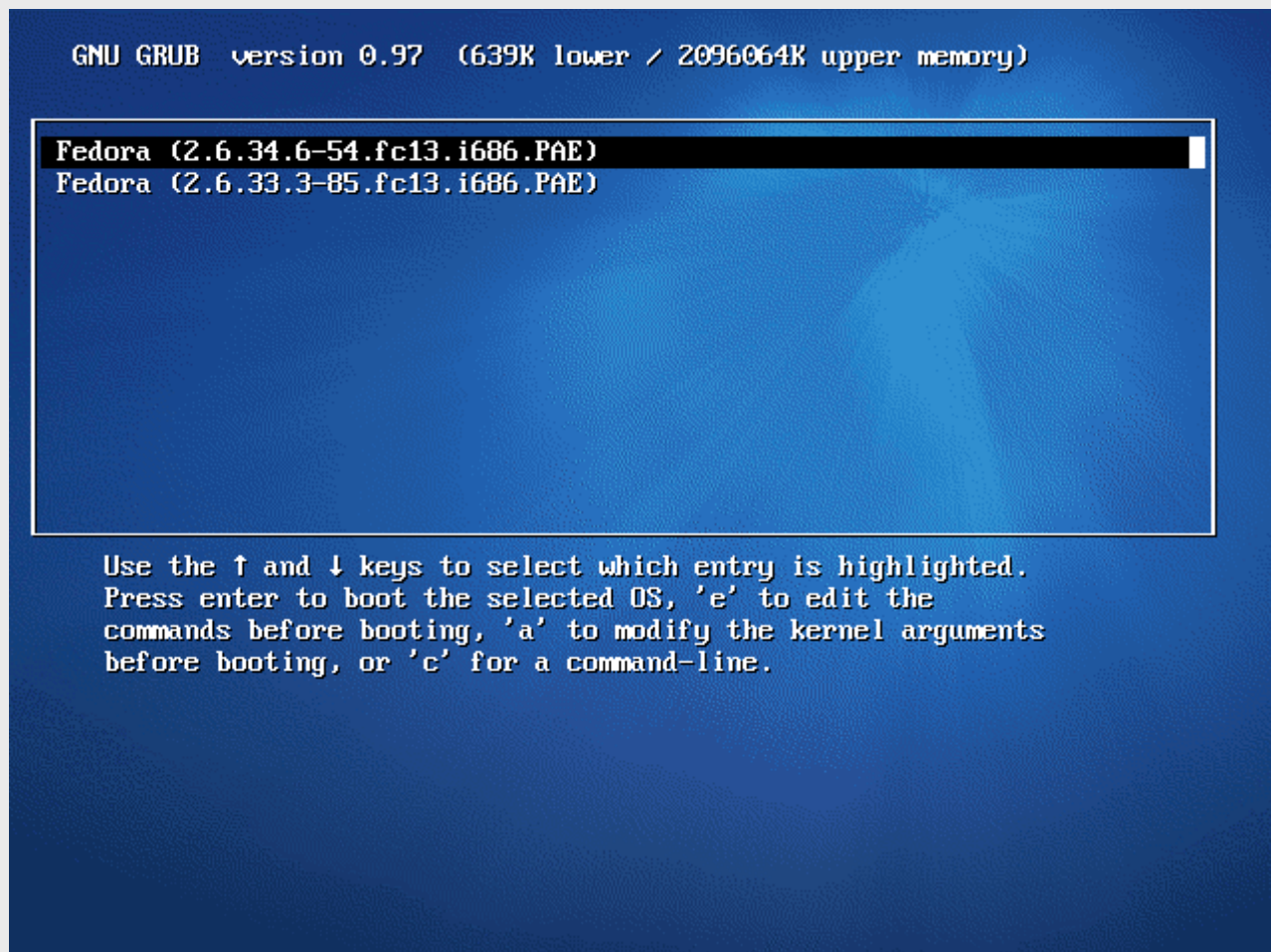
GRUB

A terminal window titled 'root@mokena-gw:~' with standard window controls. The terminal displays the output of the command 'cat /boot/grub/grub.conf'. The output shows the GRUB configuration file generated by anaconda, including boot device settings, kernel paths, and various boot options like 'serial', 'terminal', 'title', 'root', 'kernel', 'rd_NO_LUKS', 'LANG', 'rd_NO_MD', 'KEYTABLE', 'rd_LVM_LV', 'SYSFONT', 'crashkernel', 'auto', 'rhgb', 'quiet', 'rd_NO_DM', 'vmalloc=512M', and 'initrd'.

```
[root@mokena-gw ~]# cat /boot/grub/grub.conf
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You have a /boot partition.  This means that
#           all kernel and initrd paths are relative to /boot/, eg.
#           root (hd0,0)
#           kernel /vmlinuz-version ro root=/dev/mapper/VolGroup-lv_root
#           initrd /initrd-[generic-]version.img
#boot=/dev/sda
default=0
timeout=5
serial --unit=0 --speed=115200
terminal --timeout=5 serial console
title CentOS (3.4.83-3.tos2_2)
    root (hd0,0)
    kernel /vmlinuz-3.4.83-3.tos2_2 ro root=/dev/mapper/VolGroup-lv_root rd_
NO_LUKS LANG=en_US.UTF-8 rd_NO_MD KEYTABLE=us rd_LVM_LV=VolGroup/lv_swap consol
e=ttyS0,115200 rd_LVM_LV=VolGroup/lv_root SYSFONT=latarcyrheb-sun16 crashkernel=
auto rhgb quiet rd_NO_DM vmalloc=512M
    initrd /initramfs-3.4.83-3.tos2_2.img
[root@mokena-gw ~]#
```


GRUB

Figure 8-2:
GRUB
bootloader
screen in
Fedora



GRUB

GRUB
bootloader
screen in
Ubuntu

```
GNU GRUB  version 1.98-1ubuntu7

Ubuntu, with Linux 2.6.32-24-generic
Ubuntu, with Linux 2.6.32-24-generic (recovery mode)
Ubuntu, with Linux 2.6.32-21-generic
Ubuntu, with Linux 2.6.32-21-generic (recovery mode)
Memory test (memtest86+)
Memory test (memtest86+, serial console 115200)
```

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line.

GRUB

- ◆ GRUB root partition
 - Partition containing the stage 2 boot loader and the **grub.conf** file
- ◆ GRUB normally allows manipulation of boot loader
 - To prevent, enable password protection
- ◆ **grub-md5-crypt** command
 - Generates encrypted password for use in **grub.conf** file

GRUB

- ◆ Press any key during first five seconds after the BIOS POST to get graphical GRUB boot menu
 - Manipulate the boot process
 - Press 'c' to get a **grub>** prompt to enter commands
 - Help screen provides list of all available commands

GRUB

- ◆ **grub-install** command
 - Installs GRUB boot loader
 - Typically for reinstallation when GRUB becomes damaged

GRUB

Configuring
the GRUB
boot loader
screen

GNU GRUB version 0.97 (639K lower / 2096064K upper memory)

```
root (hd0,0)
kernel /vmlinuz-2.6.34.6-54.fc13.i686.PAE ro root=/dev/mapper/vg_itm4-
initrd /initramfs-2.6.34.6-54.fc13.i686.PAE.img
```

Use the ↑ and ↓ keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('O' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.

GRUB

GRUB
prompt
boot loader
screen

```
GNU GRUB  version 0.97  (639K lower / 2096064K upper memory)

[ Minimal BASH-like line editing is supported.  For the first word, TAB
  lists possible command completions.  Anywhere else TAB lists the possible
  completions of a device/filename.  ESC at any time exits.]

grub> █
```

GRUB

Figure 8-5:
Viewing help
at the GRUB
boot loader
screen
prompt

```
GNU GRUB  version 0.97  (639K lower / 2096064K upper memory)

[ Minimal BASH-like line editing is supported.  For the first word, TAB
  lists possible command completions.  Anywhere else TAB lists the possible
  completions of a device/filename.  ESC at any time exits.]

grub> help
background RRGGBB
boot
chainloader [--force] FILE
color NORMAL [HIGHLIGHT]
displayapm
find FILENAME
geometry DRIVE [CYLINDER HEAD SECTOR]
help [--all] [PATTERN ...]
initrd FILE [ARG ...]
makeactive
md5crypt
modulenounzip FILE [ARG ...]
partnew PART TYPE START LEN
reboot
rootnoverify [DEVICE [HDBIAS]]
setkey [TO_KEY FROM_KEY]
grub will attempt to avoid printing an
terminal [--dumb] [--no-echo] [--no-ed
testvbe MODE
uppermem KBYTES

blocklist FILE
cat FILE
clear
configfile FILE
displaymem
foreground RRGGBB
halt [--no-apm]
hide PARTITION
kernel [--no-mem-option] [--type=TYPE]
map TO_DRIVE FROM_DRIVE
module FILE [ARG ...]
pager [FLAG]
parttype PART TYPE
root [DEVICE [HDBIAS]]
serial [--unit=UNIT] [--port=PORT] [--
setup [--prefix=DIR] [--stage2=STAGE2_
splashimage FILE
terminfo [--name=NAME --cursor-address
unhide PARTITION
vbeprobe [MODE]

grub> |
```

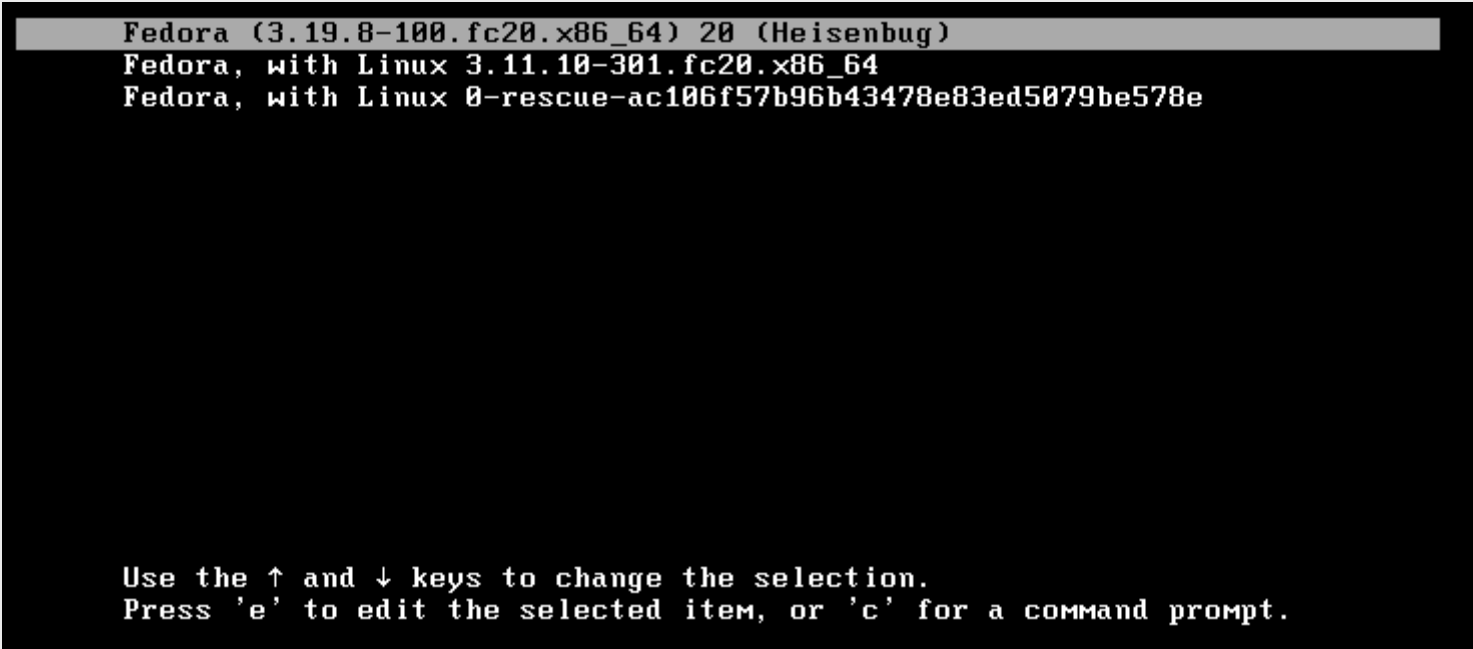

GRUB2

- ◆ Grand Unified Bootloader version 2
 - most common boot loader used on modern Linux systems
 - Similar structure to GRUB
 - Stage2 loads a terminal-friendly boot loader screen
- ◆ When a new device driver needs to be loaded by the boot loader, package may add a file to `/etc/default/grub.d`

GRUB2

- ◆ grub2 configuration is
`/boot/grub/grub.cfg` or
`/boot/grub2/grub.cfg`
- ◆ To configure or edit, add or modify lines
in the `/etc/default/grub` file
- ◆ After modifying the file or adding scripts
to the `/etc/grub.d` directory, run `grub2-
mkconfig` to rebuild
`/boot/grub/grub.cfg`

GRUB2



```
Fedora (3.19.8-100.fc20.x86_64) 20 (Heisenbug)
Fedora, with Linux 3.11.10-301.fc20.x86_64
Fedora, with Linux 0-rescue-ac106f57b96b43478e83ed5079be578e
```

```
Use the ↑ and ↓ keys to change the selection.
Press 'e' to edit the selected item, or 'c' for a command prompt.
```

GRUB2

- ◆ Use the **grub2-install** command if the GRUB2 boot loader becomes damaged
 - **# grub2-install /dev/sda**

LILO

- ◆ Stands for Linux Loader
- ◆ Traditional Linux boot loader
 - No longer supported by Fedora or Ubuntu
- ◆ Typically located on MBR
- ◆ Lilo boot: prompt appears following BIOS POST
 - Allows choice of OS to load at startup
- ◆ To configure, edit **`/etc/lilo.conf`** file

GRUB vs LILO

- ◆ LILO has no interactive command interface, whereas GRUB does
- ◆ LILO does not support booting from a network, whereas GRUB does
- ◆ LILO stores information in the MBR. If you make a change, you have to rewrite the MBR. If this fails for some reason, it is unrecoverable where GRUB would default to the command line for troubleshooting

Linux Initialization

- ◆ The kernel resumes control after Linux is loaded
 - Executes first daemon process (init daemon)
 - Brings the system into a usable state
- ◆ Recent Linux distributions have adopted the Systemd system initialization process
 - Older Linux systems used a UNIX standard called SysV
 - Systemd is completely compatible with SysV
 - Implements new features for management

Linux Initialization

- ◆ 2 SysV system initialization processes
 - Traditional SysV
 - Upstart
- ◆ In both systems
 - The init daemon runs a series of scripts to start other daemons to provide system services
- ◆ The init daemon is responsible for starting and stopping daemons after system initialization

Runlevels

◆ Runlevel

- Defines number and type of daemons loaded into memory and executed
- init daemon responsible for changing runlevels
 - Often called initstates
- Seven standard runlevels

Runlevels

◆ **runlevel** command

- Displays the current and most recent previous runlevel

◆ **init** command

- Changes the operating system from one runlevel to another

Runlevels

Runlevel	Common Name	Description
0	halt	A system that has no daemons active in memory and is ready to be powered off.
1 s S single	single user mode	A system that has only enough daemons to allow one user to log in and perform system maintenance tasks. A user is automatically logged into the system as the root user when entering Single User Mode.
2	multiuser mode	A system that has most daemons started and allows multiple users to log in and use system services. Most common network services other than specialized network services are available in this runlevel as well.

Table 8-3: Linux runlevels

Runlevels

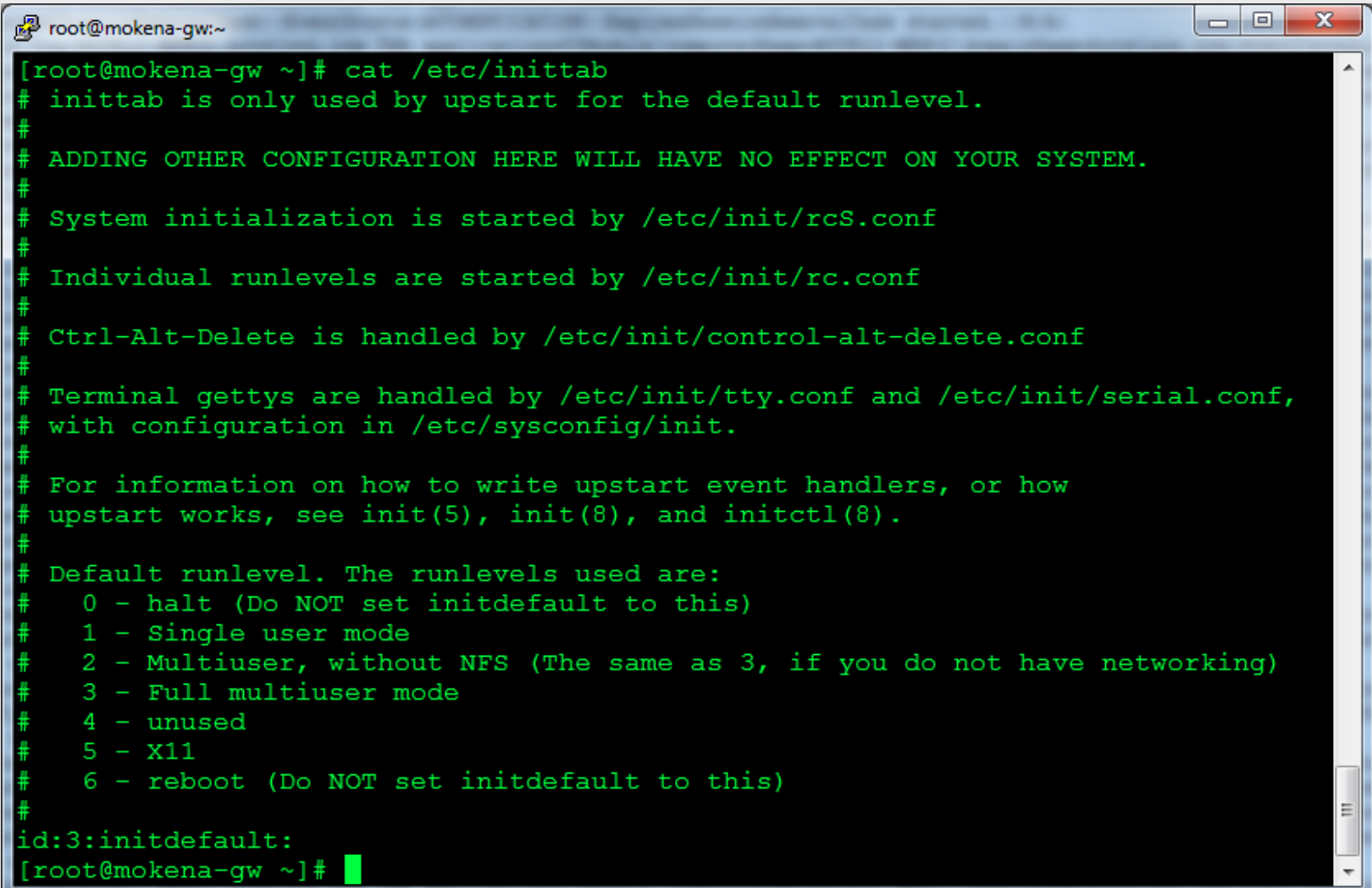
Runlevel	Common Name	Description
3	extended multiuser mode	A system that has the same abilities as Multi-User Mode, yet with all extra networking services started (i.e., NFS).
4	not used	Not normally used, but can be customized to suit your needs
5	graphical mode	A system that has the same abilities as extended multiuser mode, yet with a graphical login program called the GNOME Display Manager (gdm) started on tty7 that allows for graphical logins.
6	reboot	A special runlevel used to reboot the system.

Table 8-3: Linux runlevels

The `/etc/inittab` File

- ◆ Indicates default runlevel which the init daemon enters
 - Syntax: `id:5:initdefault:`
- ◆ Contains single uncommented line and series of explanatory comments
- ◆ Consulted to start or stop daemons

The /etc/inittab File



```
root@mokena-gw:~  
[root@mokena-gw ~]# cat /etc/inittab  
# inittab is only used by upstart for the default runlevel.  
#  
# ADDING OTHER CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.  
#  
# System initialization is started by /etc/init/rcS.conf  
#  
# Individual runlevels are started by /etc/init/rc.conf  
#  
# Ctrl-Alt-Delete is handled by /etc/init/control-alt-delete.conf  
#  
# Terminal gettys are handled by /etc/init/tty.conf and /etc/init/serial.conf,  
# with configuration in /etc/sysconfig/init.  
#  
# For information on how to write upstart event handlers, or how  
# upstart works, see init(5), init(8), and initctl(8).  
#  
# Default runlevel. The runlevels used are:  
# 0 - halt (Do NOT set initdefault to this)  
# 1 - Single user mode  
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)  
# 3 - Full multiuser mode  
# 4 - unused  
# 5 - X11  
# 6 - reboot (Do NOT set initdefault to this)  
#  
id:3:initdefault:  
[root@mokena-gw ~]#
```

Runtime Configuration Scripts

- ◆ Runtime configuration (rc) scripts:
 - scripts that prepare the system, start daemons and bring system to usable state
 - Executed by init daemon
 - Consulted to start or stop daemons
- ◆ **/etc/rc.d/rc.sysinit**
 - First script executed during system startup
 - Init hardware, set variables, check filesystems, perform system tasks for daemon loading

Runtime Configuration Scripts

- ◆ **dmesg** command: shows output of hardware detection and **/etc/rc.d/rc.sysinit** script
- ◆ **/etc/rc.d/rc.local** file
 - Performs post-system-startup tasks

Runtime Configuration Scripts

- ◆ `/etc/rc.d/rc` command: Executes all files starting with **S** or **K** in `/etc/rc.d/rc*.d/`
 - Runlevel must be specified; 5 is default
 - will execute files in `/etc/rc.d/rc5.d/`
 - Symbolic links to scripts that start daemons
 - **S** means “Start,” **K** means “Kill”
 - The number next to the **S** or **K** is the order in which the script is executed

Runtime Configuration Scripts

```
[root@mokena-gw ~]# ll /etc/rc3.d/
total 0
lrwxrwxrwx 1 root root 17 Jun 19 16:29 K01hostapd -> ../init.d/hostapd
lrwxrwxrwx 1 root root 27 Jul 16 14:43 K01multipath.monitor -> ../init.d/multipath.monitor
lrwxrwxrwx 1 root root 19 Jul 16 14:43 K01pacemaker -> ../init.d/pacemaker
lrwxrwxrwx 1 root root 16 Jan 24 2015 K01smartd -> ../init.d/smartd
lrwxrwxrwx 1 root root 17 Jul 16 14:43 K10cportal -> ../init.d/cportal
lrwxrwxrwx 1 root root 14 Jul 16 14:43 K10sa2d -> ../init.d/sa2d
lrwxrwxrwx 1 root root 16 Oct 1 21:21 K30snortd -> ../init.d/snortd
lrwxrwxrwx 1 root root 18 Jul 16 14:43 K35chainadm -> ../init.d/chainadm
lrwxrwxrwx 1 root root 20 Jul 16 14:43 K35conntrackd -> ../init.d/conntrackd
lrwxrwxrwx. 1 root root 15 Jul 29 2014 K35dhcpd -> ../init.d/dhcpd
lrwxrwxrwx. 1 root root 16 Jul 29 2014 K35dhcpd6 -> ../init.d/dhcpd6
lrwxrwxrwx 1 root root 18 Jul 16 14:43 K35dhcrelay -> ../init.d/dhcrelay
lrwxrwxrwx. 1 root root 18 Jul 29 2014 K35ftpproxy -> ../init.d/ftpproxy
lrwxrwxrwx. 1 root root 13 Jul 29 2014 K35nmb -> ../init.d/nmb
lrwxrwxrwx. 1 root root 18 Jul 29 2014 K35poppproxy -> ../init.d/poppproxy
lrwxrwxrwx. 1 root root 13 Jul 29 2014 K35smb -> ../init.d/smb
lrwxrwxrwx. 1 root root 19 Jul 29 2014 K35smtpproxy -> ../init.d/smtpproxy
lrwxrwxrwx. 1 root root 19 Jul 29 2014 K40polserver -> ../init.d/polserver
lrwxrwxrwx. 1 root root 20 Jul 29 2014 K50netconsole -> ../init.d/netconsole
lrwxrwxrwx. 1 root root 13 Jul 29 2014 K60nfs -> ../init.d/nfs
lrwxrwxrwx. 1 root root 20 Jul 29 2014 K69rpcsvcgssd -> ../init.d/rpcsvcgssd
lrwxrwxrwx. 1 root root 17 Jul 29 2014 K73winbind -> ../init.d/winbind
```

Runtime Configuration Scripts

```
lrwxrwxrwx. 1 root root 17 Jul 29 2014 S01sysstat -> ../init.d/sysstat
lrwxrwxrwx. 1 root root 22 Jul 29 2014 S02lvm2-monitor -> ../init.d/lvm2-monitor
lrwxrwxrwx. 1 root root 22 Oct 1 21:21 S04snort_inline -> ../init.d/snort_inline
lrwxrwxrwx. 1 root root 20 Aug 11 16:35 S05pipefilter -> ../init.d/pipefilter
lrwxrwxrwx. 1 root root 18 Jul 29 2014 S08iptables -> ../init.d/iptables
lrwxrwxrwx. 1 root root 22 Jul 29 2014 S09trustos-boot -> ../init.d/trustos-boot
lrwxrwxrwx. 1 root root 17 Jul 16 14:43 S10network -> ../init.d/network
lrwxrwxrwx. 1 root root 21 Jul 29 2014 S11portreserve -> ../init.d/portreserve
lrwxrwxrwx. 1 root root 17 Jul 29 2014 S12rsyslog -> ../init.d/rsyslog
lrwxrwxrwx. 1 root root 21 Jul 16 14:43 S20scan-driver -> ../init.d/scan-driver
lrwxrwxrwx. 1 root root 23 Jul 16 14:43 S24openvpn-ruvpn -> ../init.d/openvpn-ruvpn
lrwxrwxrwx. 1 root root 15 Jul 29 2014 S25netfs -> ../init.d/netfs
lrwxrwxrwx. 1 root root 15 Jul 29 2014 S26acpid -> ../init.d/acpid
lrwxrwxrwx. 1 root root 20 Sep 9 03:28 S26lm_sensors -> ../init.d/lm_sensors
lrwxrwxrwx. 1 root root 19 Jul 29 2014 S26udev-post -> ../init.d/udev-post
lrwxrwxrwx. 1 root root 17 Jul 16 14:43 S55dnsmasq -> ../init.d/dnsmasq
lrwxrwxrwx. 1 root root 14 Nov 20 2014 S55sshd -> ../init.d/sshd
lrwxrwxrwx. 1 root root 17 Sep 9 03:26 S57ntpd -> ../init.d/ntpd
lrwxrwxrwx. 1 root root 14 Jan 10 2015 S58ntpd -> ../init.d/ntpd
lrwxrwxrwx. 1 root root 15 Jul 16 14:43 S61clamd -> ../init.d/clamd
lrwxrwxrwx. 1 root root 16 Jul 16 14:43 S61p3scan -> ../init.d/p3scan
lrwxrwxrwx. 1 root root 20 Oct 1 15:09 S64postgresql -> ../init.d/postgresql
lrwxrwxrwx. 1 root root 20 Jul 16 14:43 S65clockspeed -> ../init.d/clockspeed
lrwxrwxrwx. 1 root root 14 Jul 16 14:43 S65havp -> ../init.d/havp
lrwxrwxrwx. 1 root root 15 Oct 2 16:46 S65squid -> ../init.d/squid
lrwxrwxrwx. 1 root root 16 Jul 29 2014 S65sweepq -> ../init.d/sweepq
lrwxrwxrwx. 1 root root 19 Oct 3 16:22 S70collector -> ../init.d/collector
lrwxrwxrwx. 1 root root 17 Jul 16 14:43 S70fdalink -> ../init.d/fdalink
```

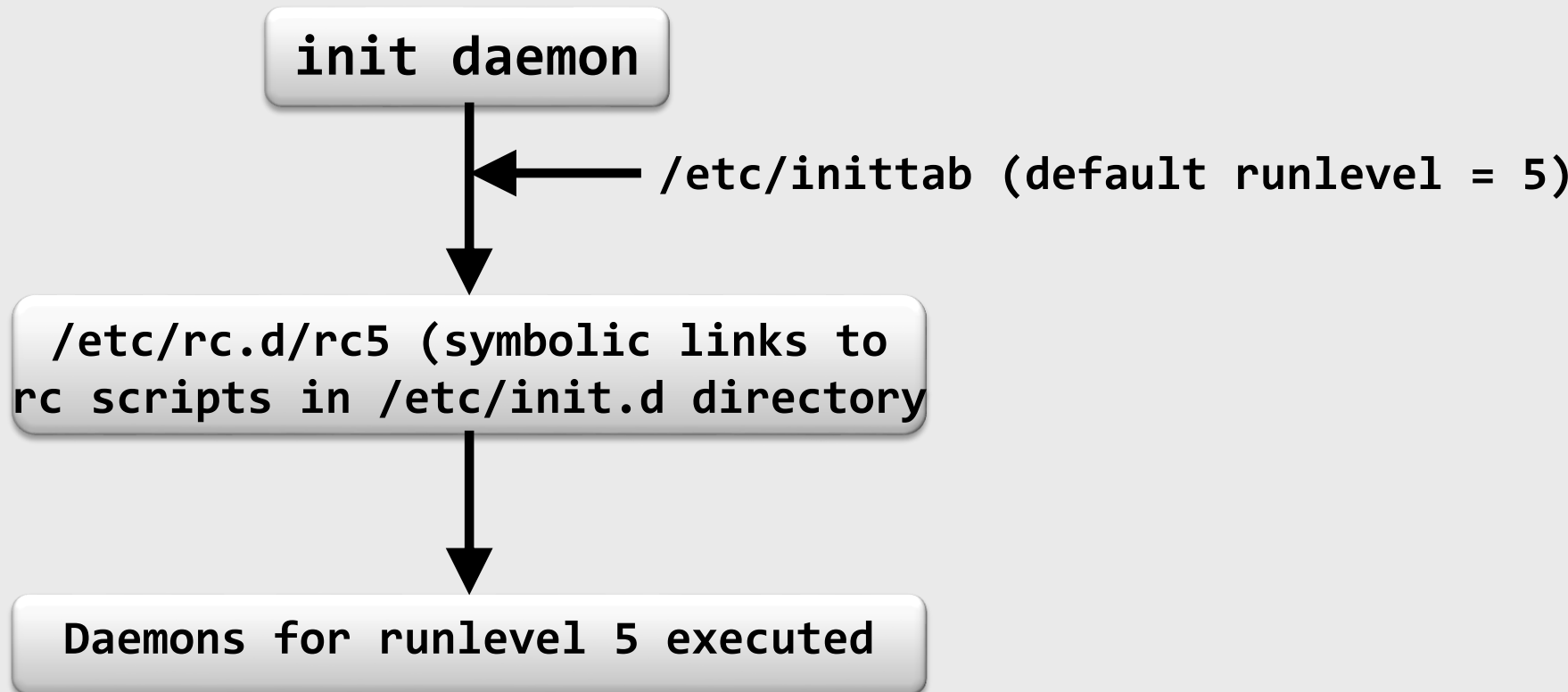
Runtime Configuration Scripts

- ◆ init daemon executes script for default runlevel (5) `/etc/rc.d/rc5` script
 - Executes all files that start with S or K in the `/etc/rc.d/rc5.d` directory

Runtime Configuration Scripts

```
[root@mokena-gw ~]# ll -d /etc/rc*
lrwxrwxrwx. 1 root root 7 Jul 29 2014 /etc/rc -> rc.d/rc
lrwxrwxrwx. 1 root root 10 Jul 29 2014 /etc/rc0.d -> rc.d/rc0.d
lrwxrwxrwx. 1 root root 10 Jul 29 2014 /etc/rc1.d -> rc.d/rc1.d
lrwxrwxrwx. 1 root root 10 Jul 29 2014 /etc/rc2.d -> rc.d/rc2.d
lrwxrwxrwx. 1 root root 10 Jul 29 2014 /etc/rc3.d -> rc.d/rc3.d
lrwxrwxrwx. 1 root root 10 Jul 29 2014 /etc/rc4.d -> rc.d/rc4.d
lrwxrwxrwx. 1 root root 10 Jul 29 2014 /etc/rc5.d -> rc.d/rc5.d
lrwxrwxrwx. 1 root root 10 Jul 29 2014 /etc/rc6.d -> rc.d/rc6.d
drwxr-xr-x. 10 root root 4096 Jul 29 2014 /etc/rc.d
lrwxrwxrwx. 1 root root 13 Jul 29 2014 /etc/rc.local -> rc.d/rc.local
lrwxrwxrwx. 1 root root 15 Jul 29 2014 /etc/rc.sysinit -> rc.d/rc.sysinit
[root@mokena-gw ~]#
```

Runtime Configuration Scripts



A traditional UNIX SysV system initialization process

Runtime Configuration Scripts

- ◆ When user specifies `runlevel1`, `init` daemon executes files in the `/etc/rc.d/rc1.d` directory
- ◆ Message during system initialization indicates whether each runtime configuration script has loaded successfully
 - Hidden by graphical boot screen display
 - Use Esc key to remove the graphical screen
- ◆ Output of runtime configuration scripts is logged to the `/var/log/messages` file

Runtime Configuration Scripts

```
INIT: version 2.85 booting
Setting default font (latarcyrheb-sun16):           [ OK ]

        Welcome to Fedora Core
        Press 'I' to enter interactive startup.

iConfiguring kernel parameters:                    [ OK ]
Setting clock (localtime): Tue Jul  6 01:16:15 CDT 2004 [ OK ]
Loading default keymap (us):                       [ OK ]
Setting hostname localhost.localdomain:             [ OK ]
Your system appears to have shut down uncleanly
Press Y within 1 seconds to force file system integrity check...
Checking root filesystem
/: clean, 103823/2060352 files, 540237/4118664 blocks
                                                    [ OK ]
Remounting root filesystem in read-write mode:      [ OK ]
Activating swap partitions:                        [ OK ]
Finding module dependencies:                       [ OK ]
Checking filesystems
/boot: recovering journal
/boot: clean, 34/26104 files, 9650/104388 blocks
                                                    [ OK ]
Mounting local filesystems:                        [ OK ]
Enabling local filesystem quotas:                  [ OK ]
Enabling swap space:                               [ OK ]
```

System startup as a result of `/etc/rc.d/rc.sysinit`

Runtime Configuration Scripts

```
INIT: Entering runlevel: 5
Entering non-interactive startup
Applying Intel IA32 Microcode update: [ OK ]
Checking for new hardware [ OK ]
Updating /etc/fstab [ OK ]
Applying iptables firewall rules: [ OK ]
Setting network parameters: [ OK ]
Bringing up loopback interface: [ OK ]
Bringing up interface eth0: [ OK ]
Starting system logger: [ OK ]
Starting kernel logger: [ OK ]
Starting irqbalance: [ OK ]
Starting portmapper: [ OK ]
Starting NFS statd: [ OK ]
Initializing random number generator: [ OK ]
Starting pcmcia: [ OK ]
Mounting other filesystems: [ OK ]
Starting up APM daemon: [ OK ]
Starting automount: No Mountpoints Defined [ OK ]
Starting smartd: [ OK ]
Starting cups: [ OK ]
Starting sshd: [ OK ]
Starting xinetd: [ OK ]
Starting sendmail: [ ]
```

System startup as a result of `/etc/rc.d/rc*.d`

Configuring Daemon Startup

- ◆ Most symbolic links in `/etc/rc.d/rc*.d` point to daemon executable files in `/etc/init.d`
- ◆ For example, to restart the cron daemon, type
 - `/etc/init.d/cron restart`
- ◆ **service** command: start, stop, or restart daemons within `/etc/rc.d/init.d` directory
- ◆ The upstart init system also provides the **stop**, **start**, and **restart** commands

Configuring Daemon Startup

- ◆ To configure a daemon to start or stop in a particular runlevel:
 - Create or modify symbolic links within `/etc/rc[runlevel].d` directories
- ◆ **chkconfig** command: view and modify daemons that are started in each runlevel
- ◆ The **chkconfig** command is not available in Ubuntu Server 14.04
 - Use the `update-rc.d` command to configure files within `/etc/rc[runlevel].d` directories

Configuring Daemon Startup

```
[root@mokena-gw ~]# chkconfig --list
acpid          0:off    1:off    2:on     3:on     4:on     5:on     6:off
cgconfig       0:off    1:off    2:off    3:off    4:off    5:off    6:off
cgmred         0:off    1:off    2:off    3:off    4:off    5:off    6:off
chainadm       0:off    1:off    2:off    3:off    4:off    5:off    6:off
check-tun      0:off    1:off    2:on     3:on     4:on     5:on     6:off
clamd          0:off    1:off    2:on     3:on     4:on     5:on     6:off
clockspeed     0:off    1:off    2:on     3:on     4:on     5:on     6:off
collector      0:off    1:off    2:on     3:on     4:on     5:on     6:off
conntrackd     0:off    1:off    2:off    3:off    4:off    5:off    6:off
corosync       0:off    1:off    2:off    3:off    4:off    5:off    6:off
cportal        0:off    1:off    2:off    3:off    4:off    5:off    6:off
crond          0:off    1:off    2:on     3:on     4:on     5:on     6:off
dhcpcd         0:off    1:off    2:off    3:off    4:off    5:off    6:off
dhcpcd6        0:off    1:off    2:off    3:off    4:off    5:off    6:off
dhcrelay       0:off    1:off    2:off    3:off    4:off    5:off    6:off
dnsmasq        0:off    1:off    2:on     3:on     4:on     5:on     6:off
fdalink        0:off    1:off    2:on     3:on     4:on     5:on     6:off
ftpproxy       0:off    1:off    2:off    3:off    4:off    5:off    6:off
getidpconf     0:off    1:off    2:on     3:on     4:on     5:on     6:off
ha-utils       0:off    1:on     2:on     3:on     4:on     5:on     6:off
harp           0:off    1:off    2:on     3:on     4:on     5:on     6:off
hostapd        0:off    1:off    2:off    3:off    4:off    5:off    6:off
httpd          0:off    1:off    2:off    3:off    4:off    5:off    6:off
ip6tables      0:off    1:off    2:off    3:off    4:off    5:off    6:off
ipsec          0:off    1:off    2:off    3:off    4:off    5:off    6:off
iptables       0:off    1:off    2:on     3:on     4:on     5:on     6:off
lighttpd       0:off    1:off    2:off    3:on     4:on     5:on     6:off
lm_sensors     0:off    1:off    2:on     3:on     4:on     5:on     6:off
lvm2-monitor   0:off    1:on     2:on     3:on     4:on     5:on     6:off
```

Configuring Daemon Startup

```
[root@mokena-gw ~]# head /etc/init.d/network
#!/bin/bash
#
# network          Bring up/down networking
#
# chkconfig: 2345 10 90
# description: Activates/Deactivates all network interfaces configured to \
#              start at boot time.
#
### BEGIN INIT INFO
# Provides: $network
[root@mokena-gw ~]# head /etc/init.d/sshd
#!/bin/bash
#
# sshd             Start up the OpenSSH server daemon
#
# chkconfig: 2345 55 25
# description: SSH is a protocol for secure remote shell access. \
#              This service starts up the OpenSSH server daemon.
#
# processname: sshd
# config: /etc/ssh/ssh_host_key
[root@mokena-gw ~]# head /etc/init.d/squid
#!/bin/sh
#
# squid
#
# chkconfig: 345 65 35
# description: Squid HTTP proxy daemon
# failover
#
# $Id: squid.init,v 1.10 2008/07/07 17:32:29 matt Exp $
#
[root@mokena-gw ~]# ll /etc/rc3.d/*network /etc/rc3.d/*sshd /etc/rc3.d/*squid
lrwxrwxrwx 1 root root 17 Jul 16 14:43 /etc/rc3.d/S10network -> ../init.d/network
lrwxrwxrwx 1 root root 14 Nov 20 2014 /etc/rc3.d/S55sshd -> ../init.d/sshd
lrwxrwxrwx 1 root root 15 Oct 2 16:46 /etc/rc3.d/S65squid -> ../init.d/squid
[root@mokena-gw ~]#
```

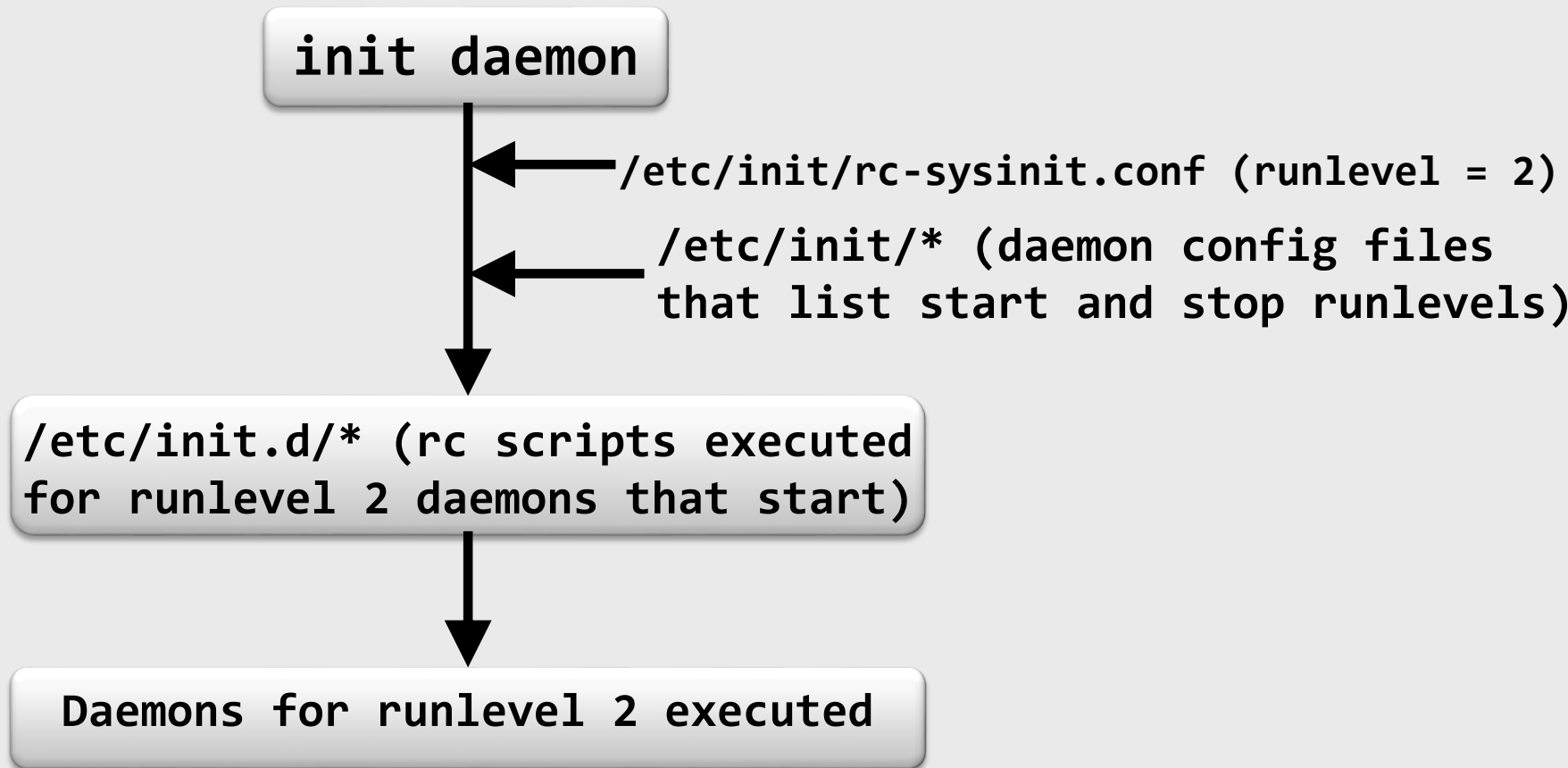
Configuring Daemon Startup

- ◆ On Linux systems using upstart, `/etc/rc.d` directories are not used
- ◆ `init` daemon identifies default runlevel in the `/etc/init/rc-sysinit.conf` file
 - Directly executes `rc` scripts in the `/etc/init.d` directory

Configuring Daemon Startup

- ◆ With upstart, each daemon has a separate configuration file within the `/etc/init` directory
 - Uses standard wildcard notation to identify runlevels it should be started or stopped in
- ◆ Upstart was default for several recent releases of Ubuntu but they have switched to systemd

Runtime Configuration Scripts



An upstart system initialization process

Configuring Daemon Startup

- ◆ Most daemons accept arguments **start**, **stop**, **restart**
 - Can be used to manipulate daemons after system startup
 - i.e., to restart the cron daemon, type **/etc/init.d/cron restart**
- ◆ **service** command
 - Start, stop, or restart daemons within the **/etc/rc.d/init.d** directory
- ◆ Upstart also uses **start**, **stop**, and **restart**

Stop & Start the SysV init Daemon

- ◆ To see a service's status,
`service servicename status`
- ◆ To stop a service,
`service servicename stop`
- ◆ To start a service,
`service servicename start`

Systemd System Init Process

- ◆ Systemd is similar to SysV
 - Can also be used to start, stop, and configure many other OS components
- ◆ Each OS component is called a unit
- ◆ Daemons are called service units
- ◆ Runlevels are called target units (or targets)

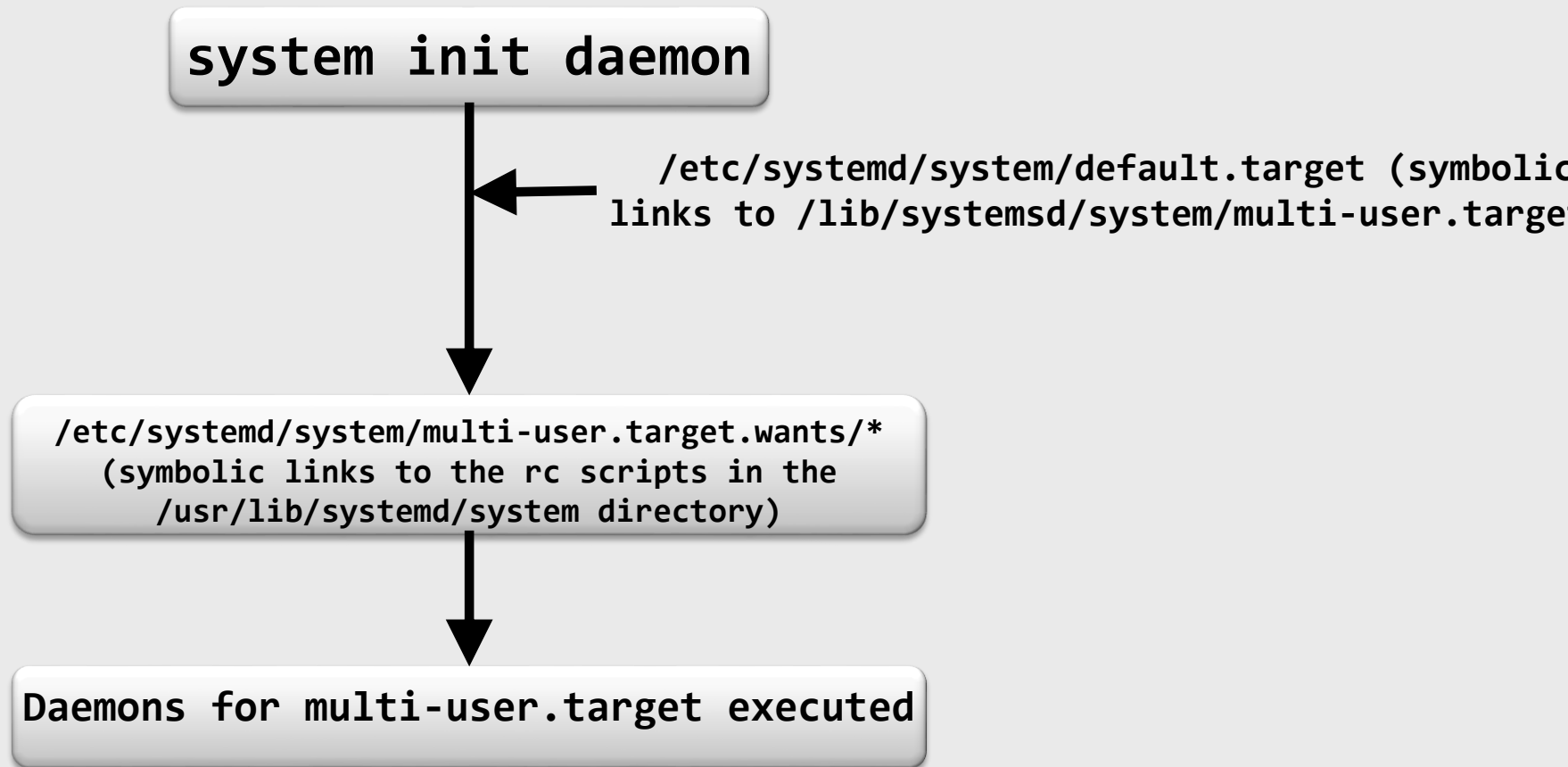
Systemd System Init Process

- ◆ Each target maps to a UNIX SysV runlevel:
 - `poweroff.target` = Runlevel 0
 - `rescue.target` = Runlevel 1 (Single User Mode)
 - `multi-user.target` = Runlevel 2, 3, and 4
 - `graphical.target` = Runlevel 5
 - `reboot.target` = Runlevel 6

Systemd System Init Process

- ◆ Default target on a system with a GUI installed is the graphical.target
- ◆ To configure a different target, update **`/etc/systemd/system/default.target`** symbolic link

Runtime Configuration Scripts



A Systemd system initialization process

Systemd System Init Process

◆ **systemctl** command

- used to start and stop daemons, as well as configure them to automatically start during system initialization

- Syntax:

systemctl restart crond.service

Systemd System Init Process

◆ **systemctl** arguments

- **status**: see detailed information about a daemon
- **enable**: configure a daemon to start in the default target
- **isolate**: to change between targets

Unit File Status

- ◆ Service and Target unit configuration files have statuses displayed via **systemctl** command:
- ◆ **enabled** - unit is currently enabled
- ◆ **disabled** - unit is current disabled
- ◆ **static**
 - Stands for “statically enabled”
 - Unit is enabled by default
 - Cannot be disabled, even by root

Stop & Start the systemd Daemon

- ◆ To see a service's status,
`systemctl status servicename.service`
- ◆ To stop a service,
`systemctl stop servicename.service`
- ◆ To start a service,
`systemctl start servicename.service`

Stop & Start the systemd Daemon

```
[root@itmo456 ~]# systemctl status sshd
sshd.service - OpenSSH server daemon
  Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled)
  Active: inactive (dead)
  Docs: man:sshd(8)
        man:sshd_config(5)

[root@itmo456 ~]# systemctl start sshd
[root@itmo456 ~]# systemctl status sshd
sshd.service - OpenSSH server daemon
  Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled)
  Active: active (running) since Mon 2015-10-19 13:28:35 CDT; 2s ago
  Docs: man:sshd(8)
        man:sshd_config(5)
  Process: 5209 ExecStartPre=/usr/sbin/ssh-keygen (code=exited, status=0/SUCCESS)
  Main PID: 5212 (sshd)
  CGroup: /system.slice/ssh.service
          └─5212 /usr/sbin/sshd -D

Oct 19 13:28:35 itmo456.iit.edu systemd[1]: Started OpenSSH server daemon.
Oct 19 13:28:35 itmo456.iit.edu sshd[5212]: Server listening on 0.0.0.0 port 22.
Oct 19 13:28:35 itmo456.iit.edu sshd[5212]: Server listening on :: port 22.
[root@itmo456 ~]# systemctl enable sshd
ln -s '/usr/lib/systemd/system/ssh.service' '/etc/systemd/system/multi-user.target.wants/ssh.service'
[root@itmo456 ~]# systemctl status sshd
sshd.service - OpenSSH server daemon
  Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled)
  Active: active (running) since Mon 2015-10-19 13:28:35 CDT; 29s ago
  Docs: man:sshd(8)
        man:sshd_config(5)
  Main PID: 5212 (sshd)
  CGroup: /system.slice/ssh.service
          └─5212 /usr/sbin/sshd -D

Oct 19 13:28:35 itmo456.iit.edu systemd[1]: Started OpenSSH server daemon.
Oct 19 13:28:35 itmo456.iit.edu sshd[5212]: Server listening on 0.0.0.0 port 22.
Oct 19 13:28:35 itmo456.iit.edu sshd[5212]: Server listening on :: port 22.
```

Systemd

- ◆ There are 2 main runlevels in systemd
 - multi-user.target (comparable to runlevel 3)
 - graphical.target (comparable to runlevel 5)
- ◆ You can change between these on demand just like with init

init	systemd	Description
[root@itm456 ~]# init 1	[root@itm456 ~]# systemctl isolate rescue.target	Change to single-user mode
[root@itm456 ~]# init 3	[root@itm456 ~]# systemctl isolate multi- user.target	Change to multi-user mode
[root@itm456 ~]# init 5	[root@itm456 ~]# systemctl isolate graphical.target	Change to graphical- user mode

X Windows System: Linux GUI Components

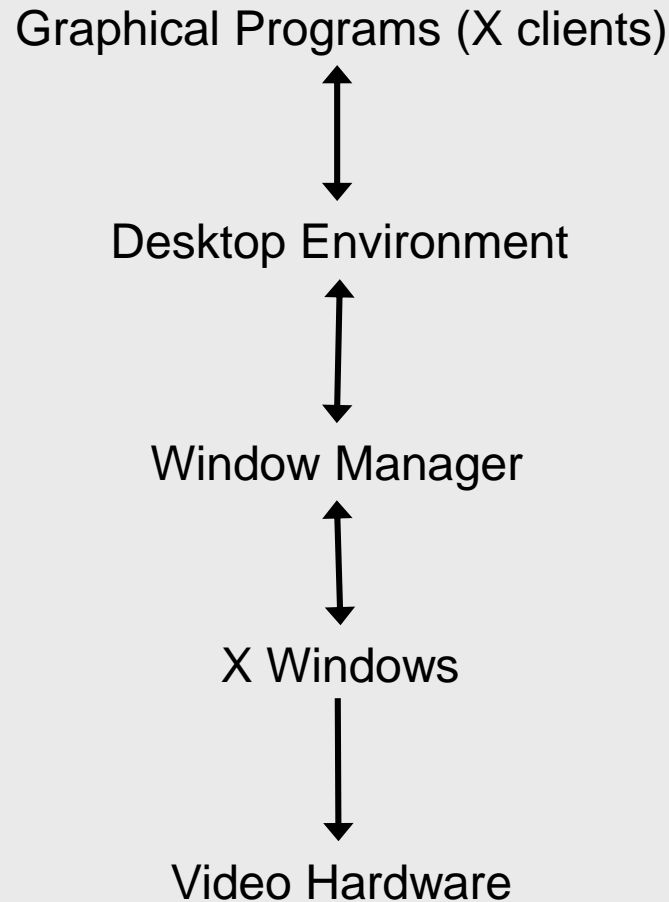


Figure 8-15: Components of the Linux GUI

X Windows

◆ X Windows

- Core component of Linux GUI
- Displays graphics to windows on the terminal screen
- Sometimes referred to as X server

◆ X client

- Requests graphics from X server and displays them on the terminal screen
- Need not run on same computer as X Windows

Wayland

- ◆ Replaces X11 in Fedora
- ◆ New display protocol with new libraries to build display server(s)
- ◆ Like systemd, built for Linux
- ◆ Security focused application isolation
- ◆ Merges display server, window manager, and compositor into one
- ◆ Free of legacy X11 code, more secure than X11

Window Managers & Desktop Environments

- ◆ Window manager
 - Modifies look and feel of X Windows
- ◆ Desktop environment
 - Works with a window manager to provide standard GUI environment
 - Uses standard programs and development tools
 - KDE and GNOME are most common

Window Managers & Desktop Environments

- ◆ K Desktop Environment (KDE)
 - K Windows Manager (kwm)
 - Default desktop environment for SUSE and many other Linux systems
- ◆ Qt toolkit
 - Software toolkit used with KDE

Window Managers & Desktop Environments

◆ GNOME Desktop Environment

- Streamlined desktop environment
- Smaller feature set than KDE
- Runs faster in many lower-memory systems
- Some view it as a more business-oriented desktop
- Default desktop for Red Hat, Fedora, Ubuntu, and others
- Uses GTK+ toolkit

Window Managers & Desktop Environments

◆ GNOME Desktop Environment

- Formerly used Sawfish & Metacity Window Managers
- Currently uses Mutter
- GNOME Shell (GNOME 3) more like Ubuntu's Unity—and many folks hate it

◆ Can configure KDE or GNOME to use a different window manager

- e.g., Compiz Fusion

Window Managers & Desktop Environments

◆ UNITY Shell Interface

- New default for Ubuntu
- Runs on GNOME & Compiz WM
- Well suited for netbooks

◆ Forks of GNOME

- MATE – renamed fork of GNOME 2, in Linux Mint, Fedora 18 and others
- Cinnamon – fork of GNOME 3 Shell, in Linux Mint and others
 - Uses Muffin WM, a fork of GNOME 3's Mutter

Window Managers & Desktop Environments

◆ LXDE Desktop Environment

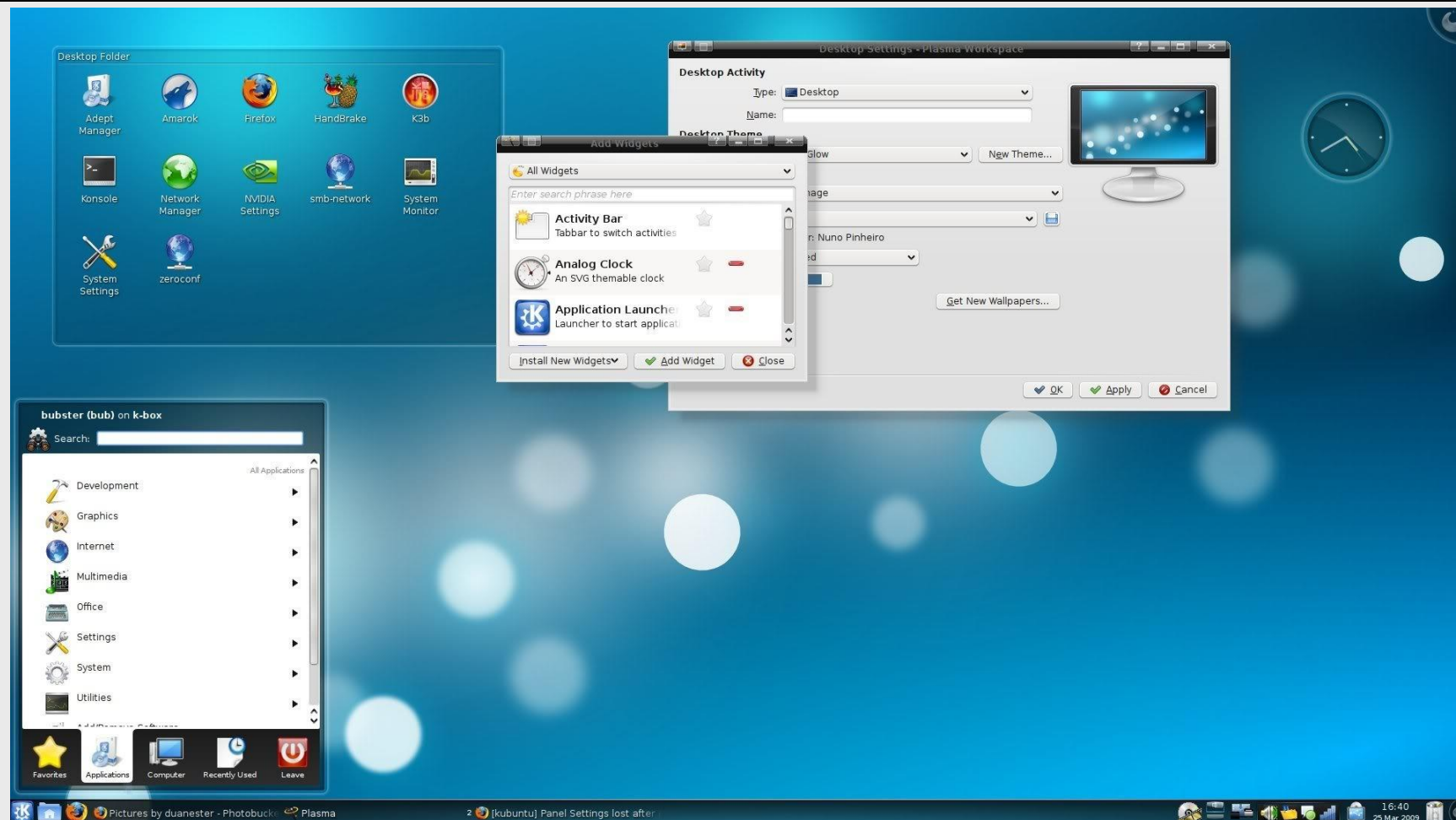
- Lightweight X11 Desktop Environment
- Designed to work well with computers on the low end of the performance spectrum
- More loosely coupled components than other desktop environments
- Native desktop environment of Knoppix, Ubuntu and U-lite
- Uses PCManFM file manager
- Uses the GTK+ toolkit

Window Managers & Desktop Environments

◆ Xfce Desktop Environment

- Consumes less hardware resources than KDE or GNOME; fast & small
- Default in Xubuntu

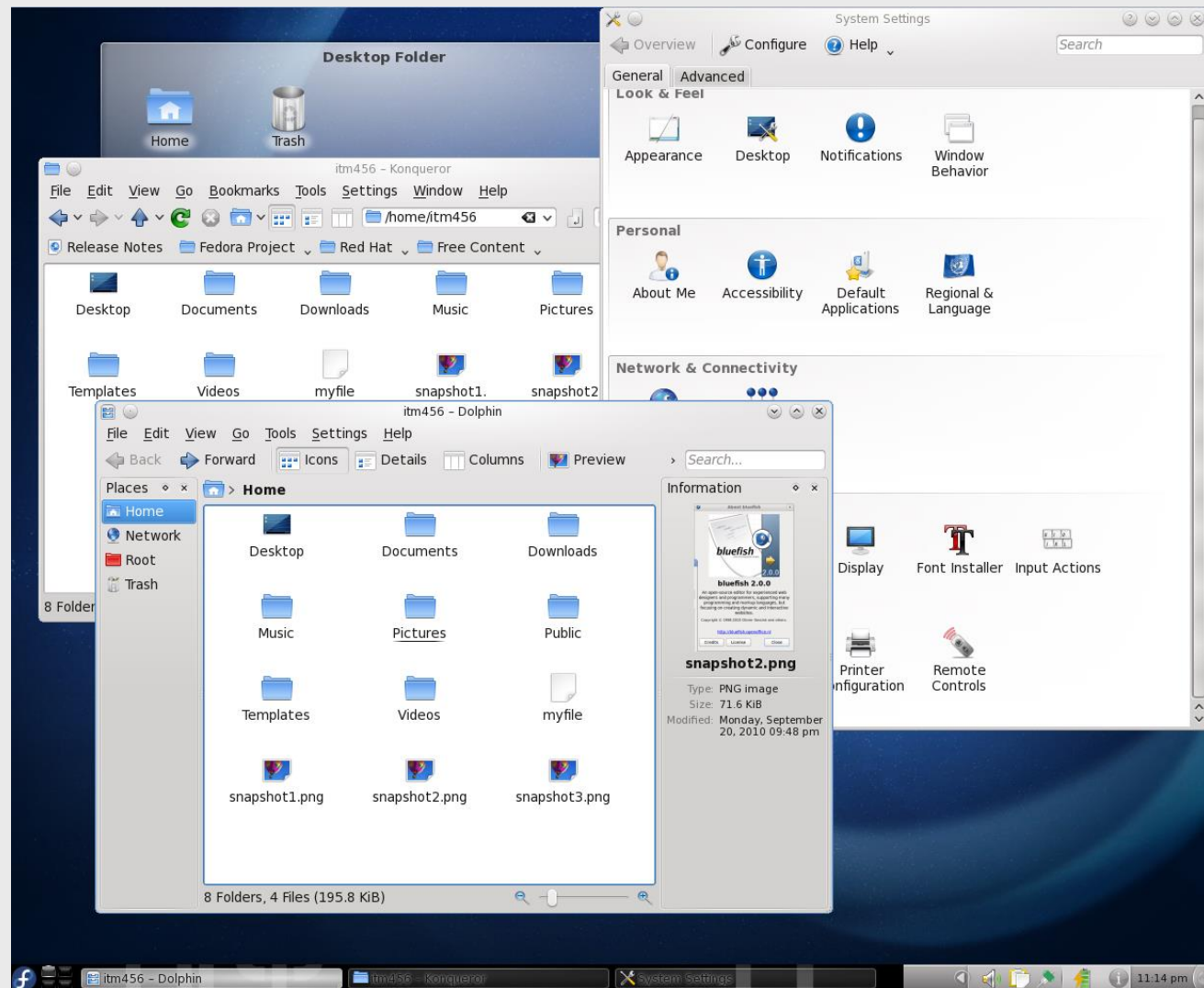
Windows Managers & Desktop Environments



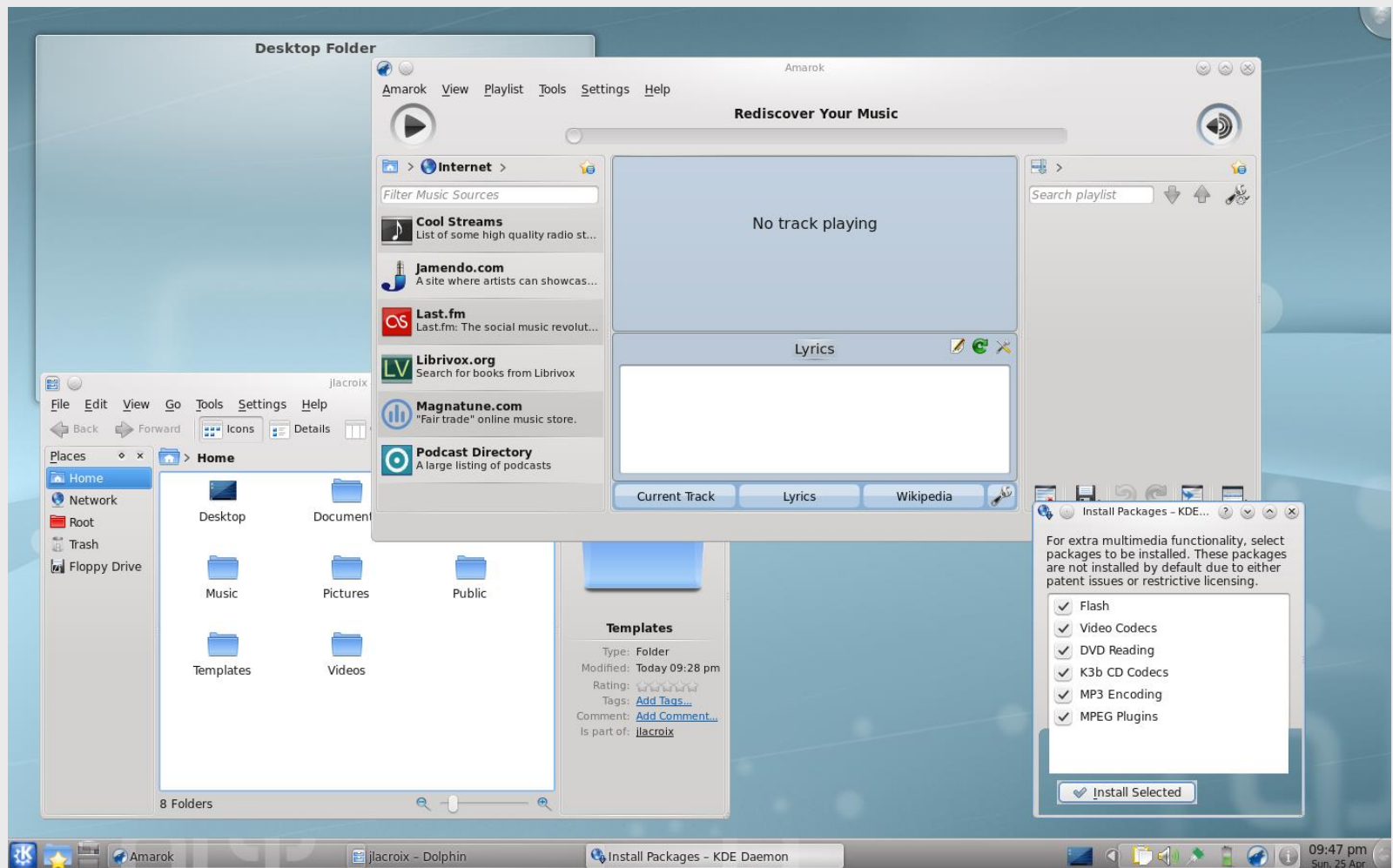
The KDE desktop environment

Window Managers & Desktop Environments

Figure 8-16:
The KDE
desktop
environment
on Fedora



Window Managers & Desktop Environments



The KDE desktop on Kubuntu

ITMO456

Windows Managers & Desktop Environments

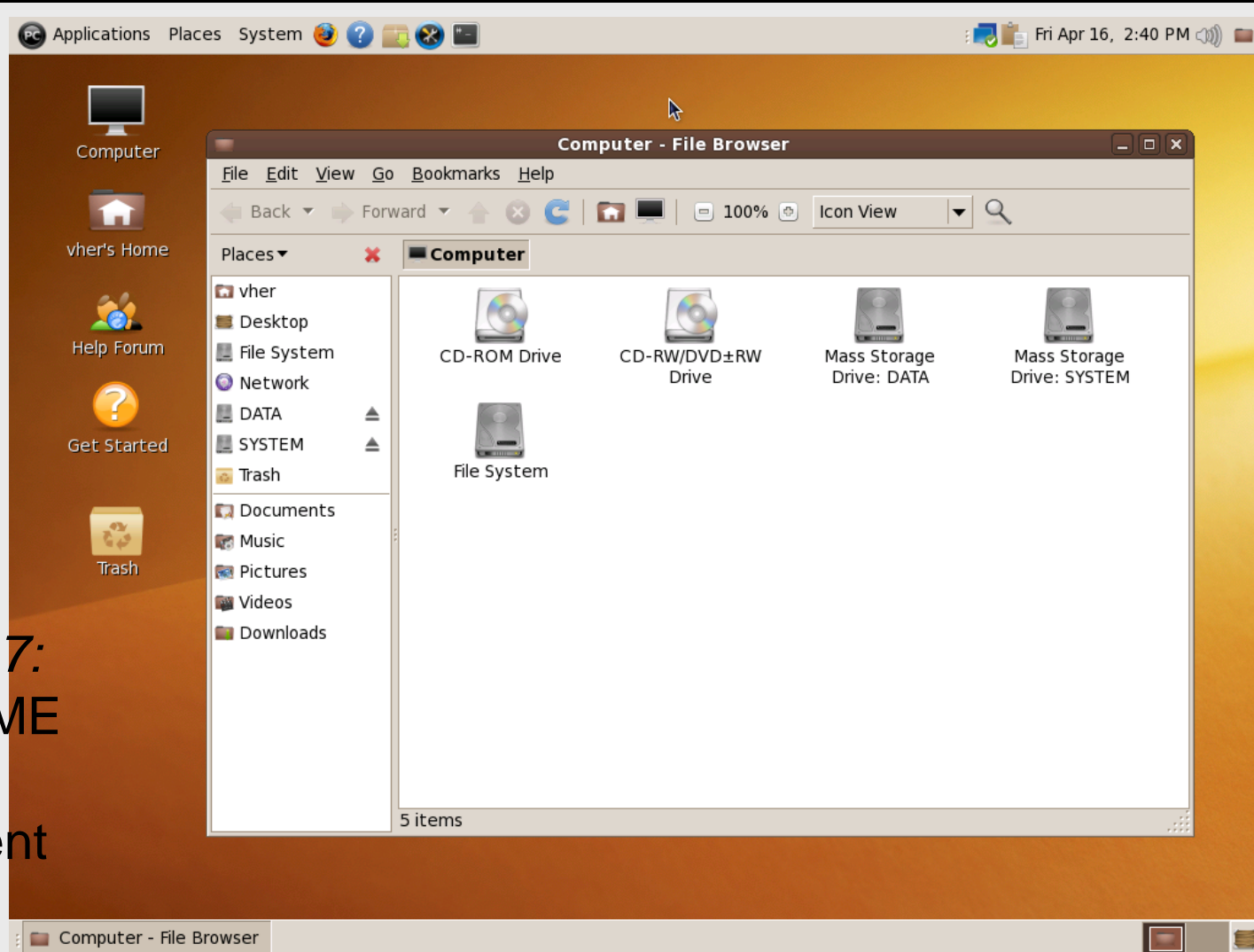


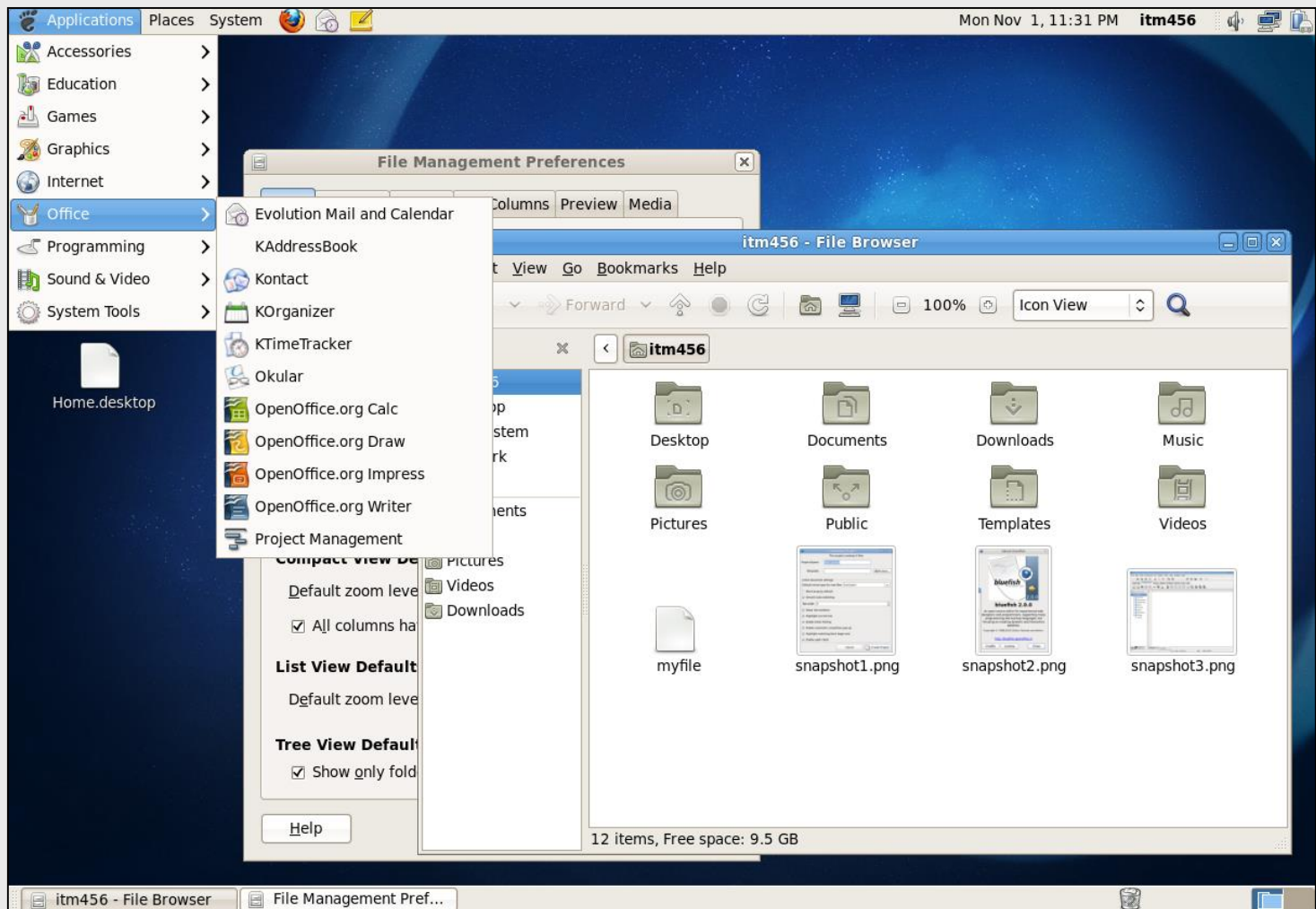
Figure 8-17:
The GNOME
desktop
environment

Window Managers & Desktop Environments



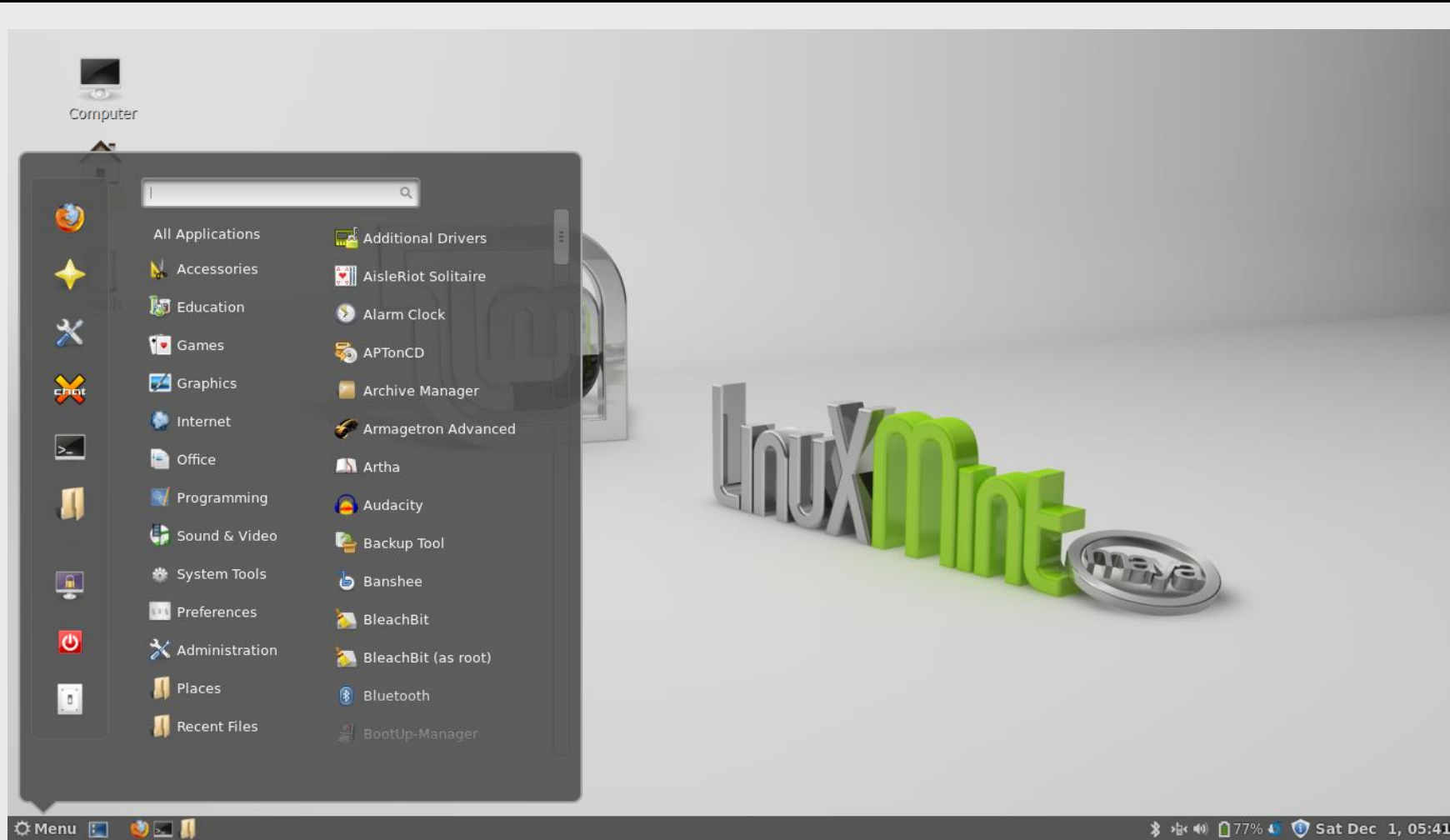
The GNOME 3 desktop

Window Managers & Desktop Environments



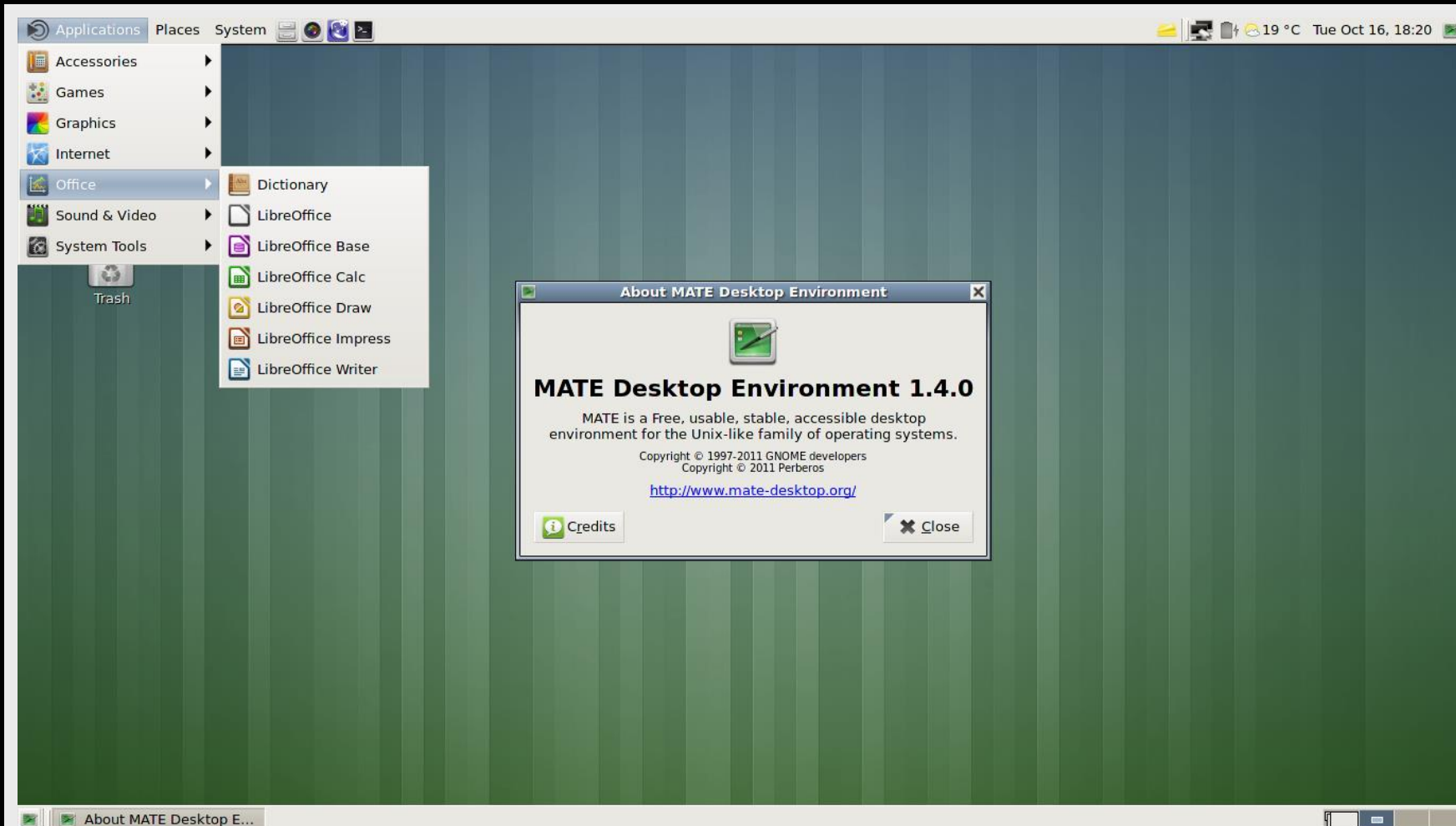
The Gnome desktop on Fedora

Window Managers & Desktop Environments



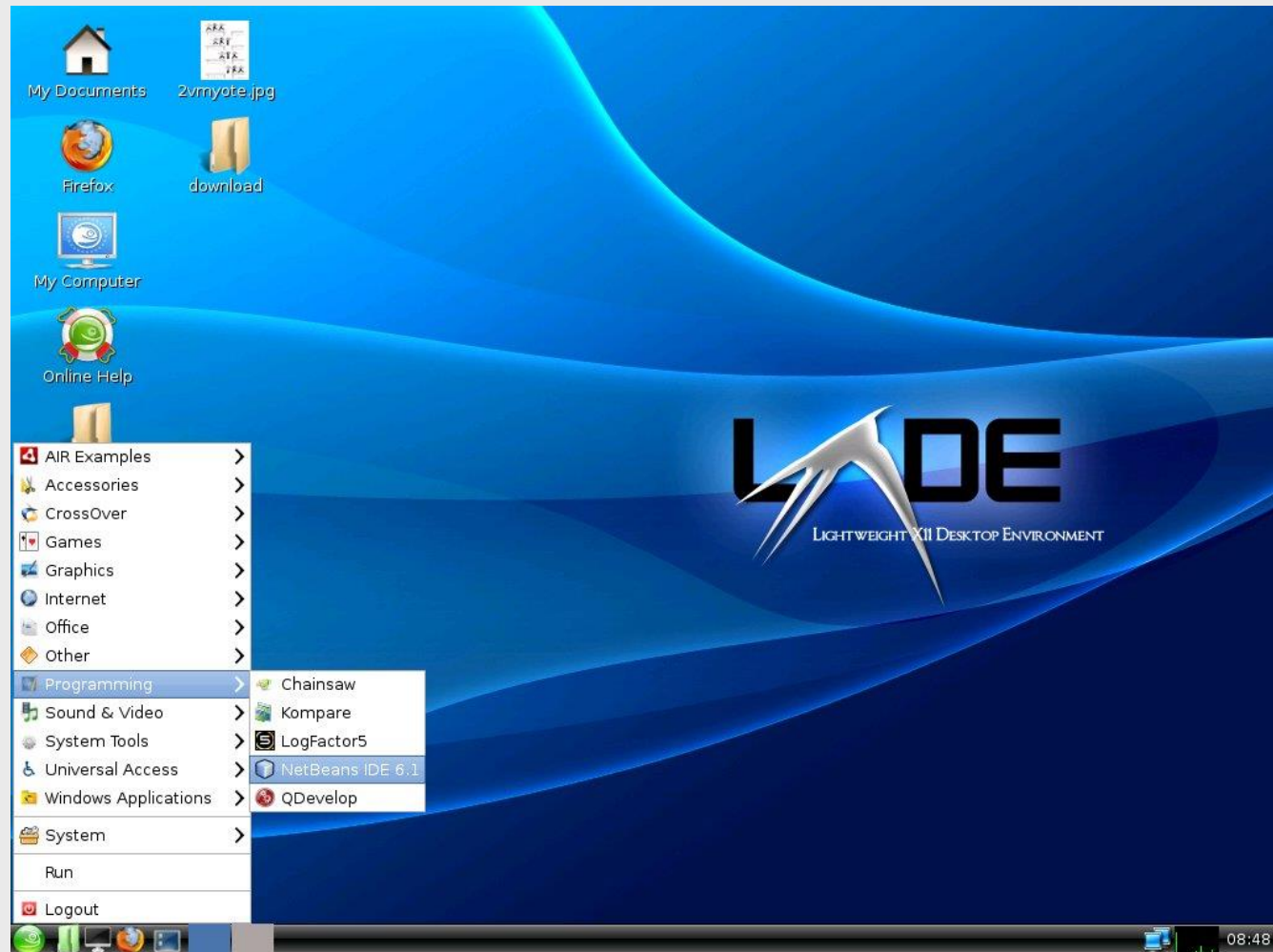
The “Cinnamon” fork of GNOME on Linux Mint

Window Managers & Desktop Environments



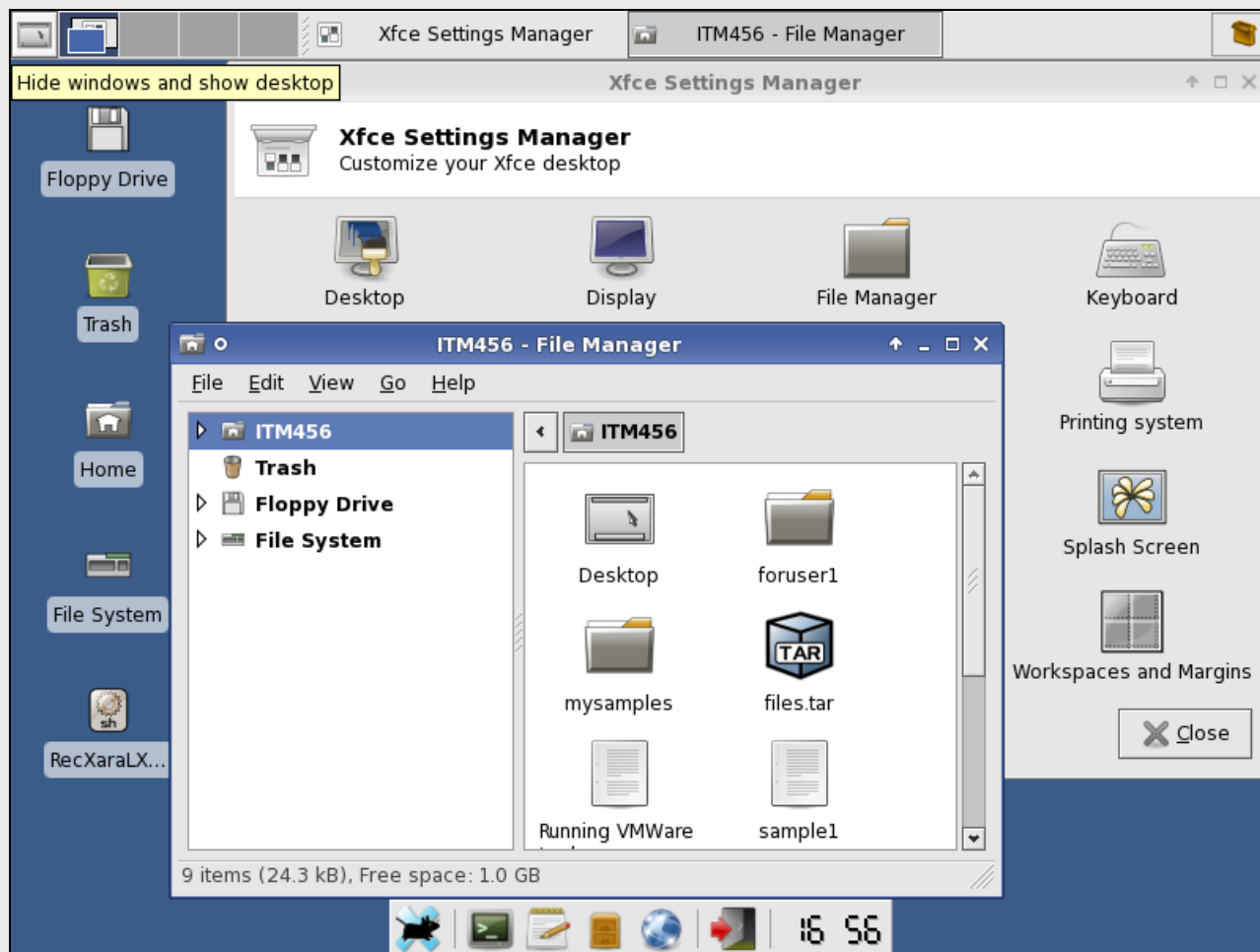
The “MATE” fork of GNOME on Debian

Window Managers & Desktop Environments



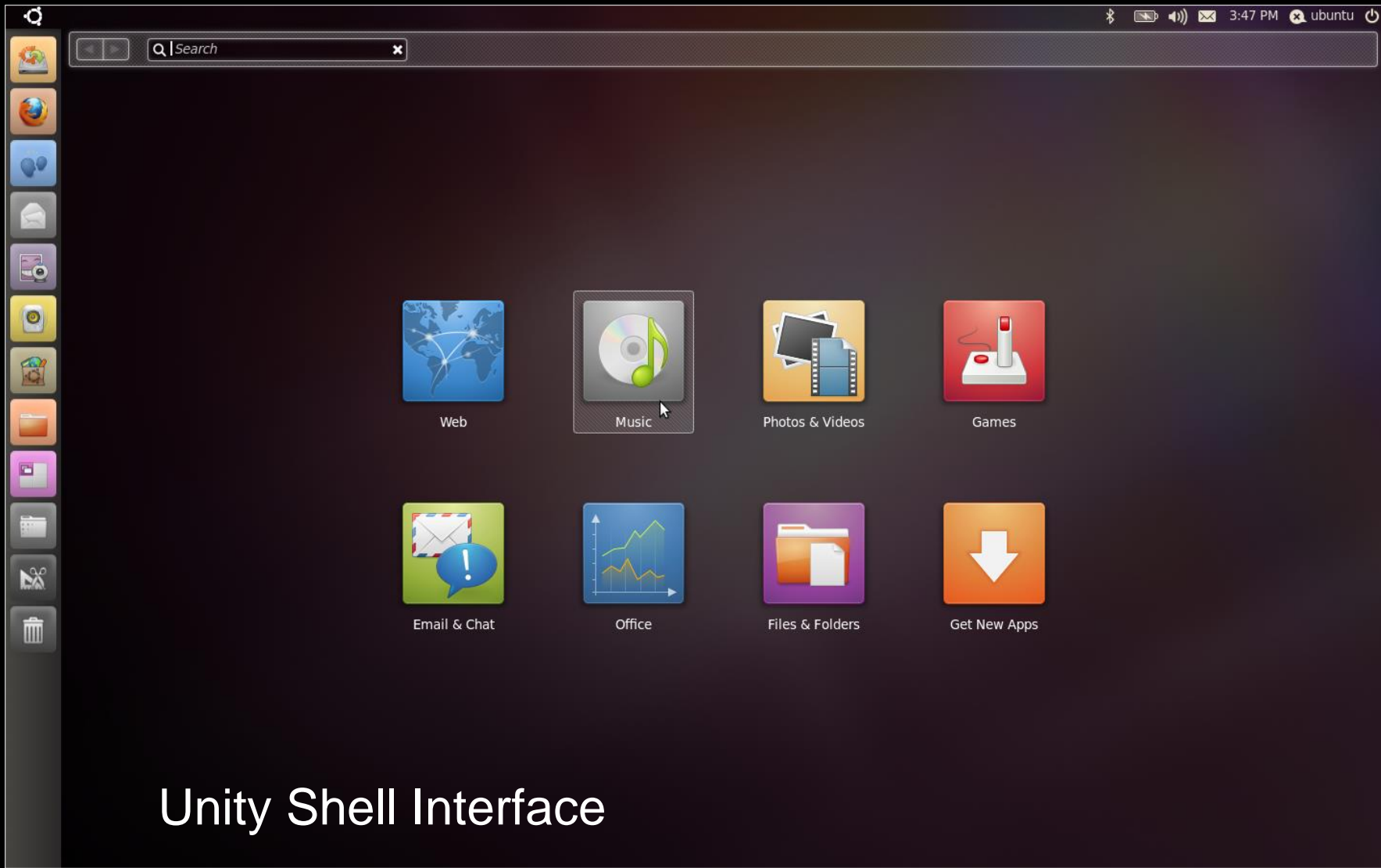
The LXDE desktop

Window Managers & Desktop Environments



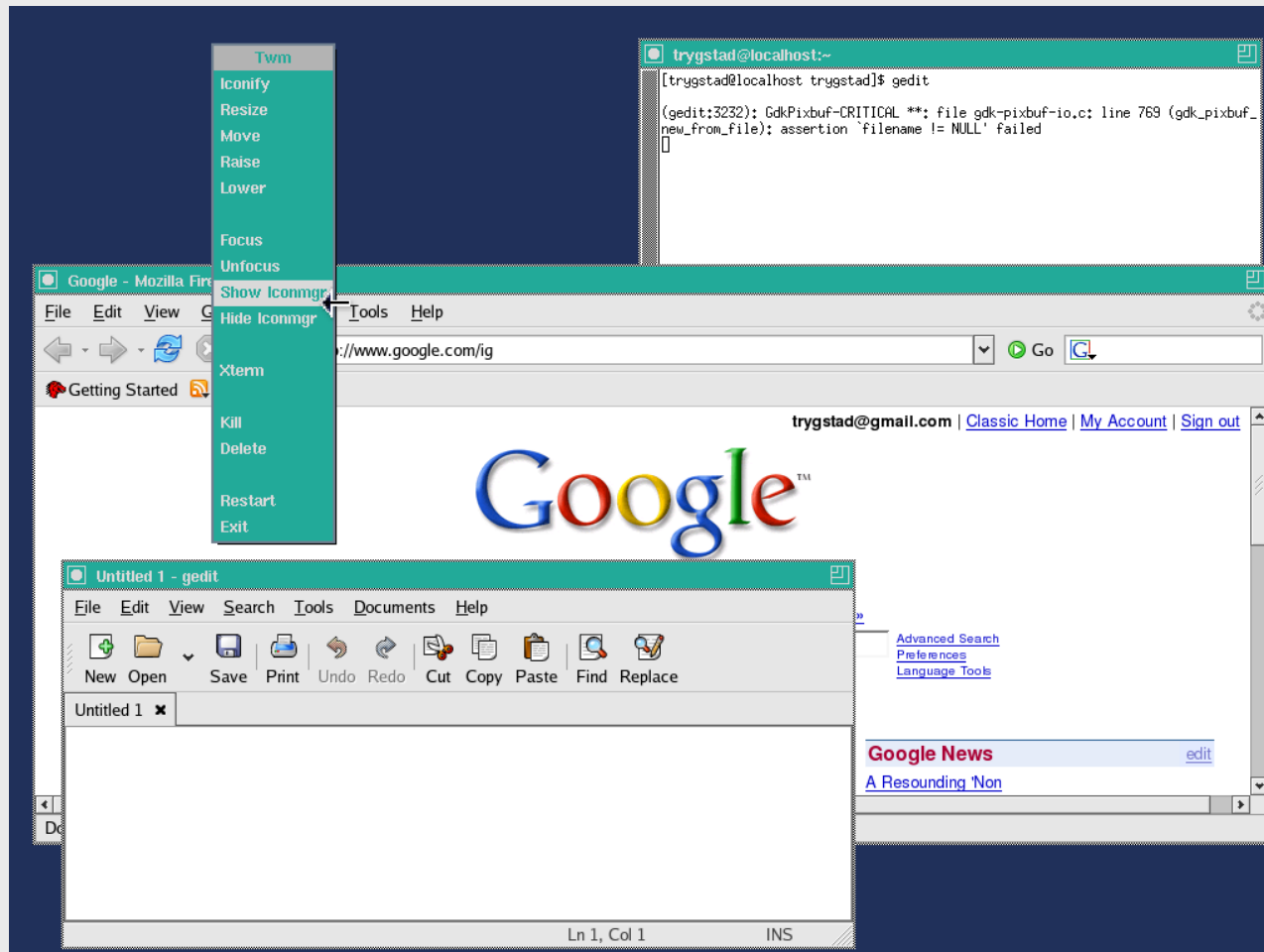
The Xfce desktop

Window Managers & Desktop Environments



Unity Shell Interface

Window Managers & Desktop Environments



The Tab Window Manager

Windows Managers & Desktop Environments

The Enlightenment Window Manager



Windows Managers & Desktop Environments

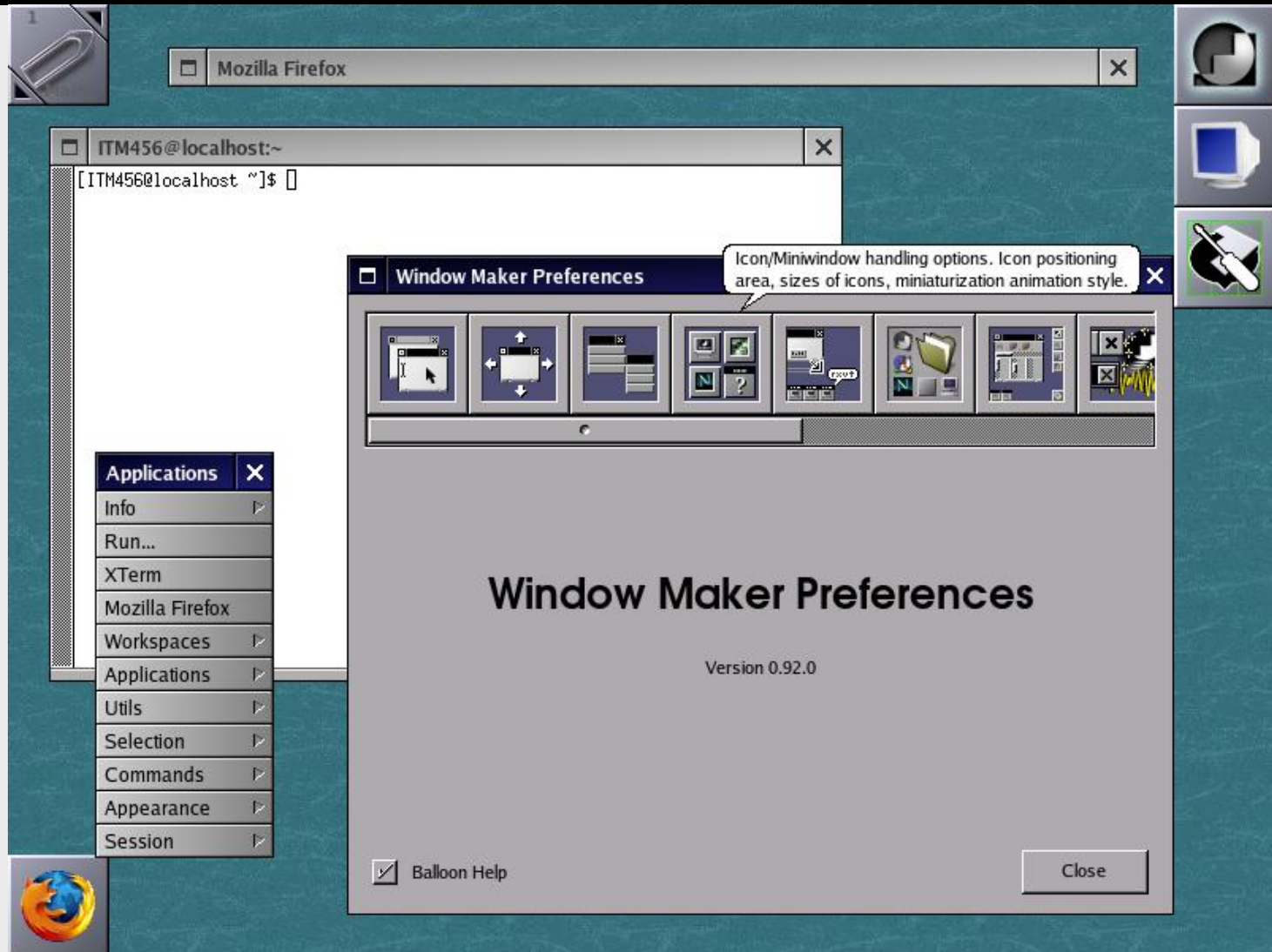
ITMO456

The Feeble
Virtual
Window
Manager
(FVWM)



Windows Managers & Desktop Environments

The
Window
Maker
Window
Manager



Windows Managers & Desktop Environments

Compiz
Window
Manager



Starting & Stopping X Windows

- ◆ When the init daemon boots to runlevel 5 or graphical.target, GNOME Display Manager (GDM) starts
 - Displays graphical login screen
 - Allows user to choose the desktop environment
- ◆ If you use runlevel1 (or rescue.target) or runlevel 2-4, the GDM is not started by default
 - Type **startx** at a character terminal to start X Windows and the default window manager

Configuring X Windows

- ◆ X Windows interfaces with video hardware
 - Requires information regarding keyboard, mouse, monitor, and video adapter card
- ◆ Attempts to automatically detect required information
 - If automatic detection fails, user needs to specify correct hardware information manually

Configuring X Windows

- ◆ User-configured settings are stored in files under the `/etc/X11/xorg.conf.d` directory
- ◆ Common settings such as the display resolution can be modified using the Displays utility within the GNOME desktop environment
- ◆ You can manually run the **system-config-keyboard** command to configure keyboard
- ◆ Use **xvidtune** utility to fine-tune the vertical refresh rate (vsync) and horizontal refresh rate (hsync)

Summary

- ◆ Boot loaders are typically loaded by the system BIOS from the MBR/GPT or the first sector of the active partition of a hard disk
- ◆ After the boot loader loads the Linux kernel, a system initialization process proceeds to load daemons that bring the system to a usable state
- ◆ There are two common system initialization processes: UNIX SysV and Systemd

Summary

- ◆ UNIX SysV uses seven runlevels to categorize a Linux system based on the number and type of daemons loaded in memory
- ◆ Systemd uses five standard targets that correspond to the seven UNIX SysV runlevels
- ◆ The init daemon is responsible for loading and unloading daemons when switching between runlevels and targets
- ◆ Daemons are typically executed during system initialization via rc scripts

Summary

- ◆ The **service** command is commonly used to start, stop, and restart UNIX SysV daemons, and the **systemctl** command is commonly used to start, stop, and restart Systemd daemons
- ◆ Use the **chkconfig** or **update-rc.d** commands to configure UNIX SysV daemon startup at boot time
- ◆ The Linux GUI has several interchangeable components: X server, X clients, window manager, and optional desktop environment

The End...

◆ Questions?