Linux Interfaces 1: CLI vs GUI

1. CLI vs GUI
   - OS interfaces: CLI, GUI
   - CLI advantages
   - shells
   - UNIX philosophy

2. GUI Elements
3. KDE
4. GNOME
5. Alternatives and 3rd-Party Applications

OS Interfaces: CLI vs GUI

There are two basic types of operating system interfaces:
- command line interface (CLI)
- graphical user interface (GUI)

CLI: users interact with the OS by typing commands.

GUI: users interact by using a mouse to point at icons-buttons and click on them.

UNIX began with a only non-graphical, CLI interfaces.

(As did Microsoft with DOS, but Apple had only GUI.)

OS Interfaces: CLI vs GUI (contd.)

Modern Linux (and UNIX) systems provide both CLI and GUI interfaces.

Most Windows/Mac users are familiar with only GUIs.

Most Linux users make at least some use of the CLI, and it is considered a standard skill for Linux users to learn.

Why the difference?

A GUI is considered easier to use (due to not having to memorize commands and their options), so better able to be used by those with limited computer experience.

In addition, the Windows CLI has been vastly inferior to the Linux/UNIX CLI.

OS Interfaces: CLI vs GUI (contd.)

The reality is that sometimes the CLI is the “better” way to accomplish a task, and sometimes a GUI is.

Potential CLI advantages:
- can require less effort:
  - type one command vs. open GUI, click, select, click, ...
- no need to move hands back-and-forth between mouse and keyboard
- can work on groups of files at a time by using file patterns
- ability to access OS functionality not available via GUI
- ability to rapidly produce new “commands” by combining existing commands
- ability to save custom commands for reuse and repetitive use
- requires less resources (e.g., no GUI/X11 at all in Linux/UNIX)
OS Interfaces: CLI vs GUI (contd.)

Thus, while the CLI is generally not required for OS users, those who do know how to use it can be more productive.

(Note that computer system administrators absolutely must be able to use the CLI—this applies to all OS's!)

Windows CLI:
- traditionally the “command-prompt” window (though very limited in power)
- recently can add PowerShell (powerful/useful, but also complicated to learn)

Mac OS X CLI: UNIX terminal (running Bash shell by default).

Shells

In Linux/UNIX terminology, the CLI is provided by a shell.

A shell is a command interpreter:
- users interactively type commands
- the shell parses and interprets these commands to determine what the user wants done
- it then interacts with the OS kernel (and other programs) to carry out the user commands

The original UNIX shell was the Bourne shell (sh).

The default Linux shell is Bash (bash), the Bourne Again SHell.

Bash is provided by the GNU project.

Shells (contd.)

A shell can be run interactively in either of two ways:
- in a console (a completely non-graphical OS interface)
- in a “terminal window” (which is running within the GUI)

There can be a variety of terminal applications available in a distro, and different desktop environments provide access to them in different ways.

E.g., in KDE you can just right-click on the desktop and select Konsole (the KDE terminal app).

Experienced Linux users generally have multiple shell sessions running at all times, to be able to quickly type a command.
**UNIX Design Philosophy**

Despite the fact that Linux/UNIX and Windows provide both CLI and GUI, there have been differences in the philosophy of how these interfaces should be designed and/or used.

**Linux/UNIX philosophy:**
- develop small components that can be combined (reused) to solve a very wide range of problems
- implement functionality in shell programs/filters and/or as part of reusable libraries
- GUIs are typically just a "front end" to a CLI program or library (so can also be accessed w/o the GUI)

**Windows philosophy:**
- develop "monolithic programs" that easily solve a fixed set of problems (but cannot be readily extended nor reused for unanticipated problems)
- nearly always accessible only via a GUI interface

**CD burning example:**
- Linux GUI burning programs are simply "front ends" to program such as cdrecord
- cdrecord can also be called from the CLI or a shell script to automate burning tasks
- Windows GUI burning programs allow access only via the GUI interface; cannot be scripted/automated
Linux Interfaces 2: GUI Elements

1. CLI vs GUI
2. GUI Elements
   - X Window (X11)
   - window managers
   - compositing WMs
   - widget toolkits
   - desktop environments
3. KDE
4. GNOME
5. Alternatives and 3rd-Party Applications

The Linux/UNIX GUI

In Linux/UNIX, the GUI is not an integral part of the OS.
Instead, the GUI is built in layers above the OS kernel.

Linux uses the X Window System (also known as X11 or X) as its base graphical input/display system.
(Note that it is not "X Windows.")

GUI layers (from bottom up):
- X – provides low level window drawing and input capabilities
- widget toolkit – provides a higher level windowing API and a consistent “look and feel”
- window manager – uses the toolkit to provide a desktop and application windows

The Linux/UNIX GUI (contd.)

X was developed at MIT from 1984–1987.

X uses a client-server architecture:
- server: display plus input devices
- clients: X applications

The X implementation used in most Linux distros is X.org.

The Linux/UNIX GUI (contd.)

X is extremely flexible:
- the protocol is network transparent: the server (display) and clients (applications) can be on different machines
- allows for easy remote administration
- allows running applications on multiple machines from a single machine
- thus, it directly supports server-based, thin-clients
- the OS is not tied to a single window manager (look)
- GUI problems do not necessarily impact the OS kernel
- servers can run with no GUI, so CPU cycles are not wasted

The downside of X’s flexibility has been that it is less efficient if direct access to graphics hardware is required (as in games).
Window Managers

Window managers provide a desktop environment with window appearance and control (in X, the display server does the actual drawing/display of windows).

Many window managers are available for Linux.

Most are intended for use only with a single desktop environment, but some can be used with multiple DE’s.

Most work with only a single widget toolkit, but some can use multiple toolkits.

WM’s differ in their looks, capabilities, resource usage, and customizability.

Window Managers (contd.)

Here are some of the most important Linux window managers:
- **KWin** – KDE’s WM
- **Mutter** – GNOME 3’s WM
- **Metacity** – GNOME 2’s WM
- **Marco** – MATE fork of Metacity
- **Muffin** – Cinnamon fork of Mutter
- **Compiz** – Ubuntu Unity’s WM
- **Enlightenment** – Enlightenment DE’s WM
- **Xfwm** – Xfce DE’s WM
- **Openbox** – LXDE’s WM
- **IceWM** – themes to imitate Windows 95, XP, 7
- **Window Maker** – GNU WM that emulates NeXT

Compositing Window Managers

Compositing window managers are a key development in modern window managers.

A compositing window manager keeps each window’s data in a separate buffer and dynamically combines the multiple windows to draw the desktop.

Compositing allows for visual effects such as 3D, transparency, fading, scaling, and shuffling.

This requires use of the graphics card’s hardware acceleration.

Significant changes had to be made to X to support compositing.

Most Linux window managers are now compositing managers.
Compositing WM’s (contd.)

Key X extensions/systems required for compositing:

- **OpenGL** – cross-platform API for 3D graphics (alternative to Direct3D)
- **Mesa** – OpenGL implementation used in X.org
- **GLX** – allows OpenGL to be used in an X window
- **DRI** – allows direct access to graphics hardware (not via X)
- **AIGLX** – ties together X and DRI

Example 3D Desktops Cube (KWin):

![3D Desktops Cube (KWin)](image)

Compositing WM’s (contd.)

Widget Toolkits

A **widget toolkit** is a library of graphical elements (**widgets**) and functions, which are used to construct a GUI.

A window manager makes calls to a **widget toolkit** to draw windows and other GUI objects, and GUI applications make calls to the toolkit to draw/implement their GUI elements.

There are two main widget toolkits used in Linux GUIs:

- **Qt** – (“q-t” or “cutie”) cross-platform, open source toolkit and GUI application framework
- **GTK+** – (originally GIMP toolkit) cross-platform, open source toolkit from GNU

Most Linux DE’s/WM’s and GUI applications are based on one of these two toolkits.

Desktop Environments

A **desktop environment** (**DE**) is what provides the desktop experience that we are used to in modern OS GUIs.

A desktop environment has to provide a window manager along with icons, panels, window decorations, etc.

Most DE’s also provide a set of basic utility applications:
- file browsers, web browsers, email clients, editors, media players, configuration tools, etc.

These utilities not only share a common look and feel, but are generally integrated and share functionality.
Desktop Environments (contd.)

One of the most striking differences between Linux and Windows or OS X is that there are a wide variety of DE’s available for Linux.

Since the GUI/desktop is separate from the OS (kernel), Linux systems are not limited to using a single DE.

In fact, one of the key characteristics that differentiates distros is their primary DE and possible additional available DE’s.

Pro: choice, Con: confusion:
It can be difficult to use an unfamiliar distro because it looks and functions differently!

Many DE’s are also highly configurable/customizable by the user: number of mouse clicks, window focus, etc.

Important/popular desktop environments:
- KDE – K Desktop Environment, one of the most popular and comprehensive DE’s
- GNOME – one of the most popular and comprehensive DE’s
- Unity – Ubuntu’s DE for GNOME 3
- Cinnamon – Mint’s DE for GNOME 3
- MATE – fork of GNOME 2, traditional desktop
- Enlightenment –
- LXDE – fast/lightweight
- Xfce – fast/lightweight
- LXQt – Qt (toolkit) port of LXDE (was Qt-Razor)

Windows and Mac Equivalents

Windows:
- Aero, Metro – DE for Windows Vista/7, Windows 8
- Desktop Window Manager (DWM) – compositing WM for Windows Vista, 7, 8
- Windows API – Windows OS functionality including both graphics and UNIX system calls

Mac OS X:
- Aqua – DE
- Quartz Compositor – compositing WM and display server
- Cocoa – widget toolkit and GUI application framework

KDE and GNOME have traditionally been the most widely used DE’s with Linux.

 GNOME provides the underlying system for the popular DE’s Unity, Cinnamon, and MATE.

The KDE and GNOME projects both include a large number of applications, so provide nearly complete desktop systems.

The freedesktop.org project (originally X Desktop Group) has worked to provide a common technology base for DE’s such as KDE and GNOME.

This has helped to improve the interoperability of applications from KDE and GNOME.
Linux Interfaces 3: KDE

1. CLI vs GUI
2. GUI Elements
3. KDE
   - Qt toolkit
   - plasma desktop
   - desktop elements
   - KDE4 desktop
   - KDE applications
   - konqueror
4. GNOME
5. Alternatives and 3rd-Party Applications

KDE

Based on Qt toolkit, written mainly in C++.

Provides basis of desktop environment:
desktop with icons, panel (menu button, clock, application bar, application applets), window titlebars (max, min, close buttons), window focus, themes, virtual desktops (pager on panel).

KDE4 was completely rewritten, using more flexible technology.

Can provide traditional desktop layout/function or very different layouts.

Distros primarily using KDE: Mageia, PCLinuxOS, OpenSuse.

KDE

Key technology is plasma desktop:
- plasmoids: all desktop element (i.e., widgets/applets)
- containment plasmoids can contain other plasmoids (e.g., the panel and taskbar)
- folder view is a key plasmoid that provides traditional desktop icons/shortcuts
- many plasmoid/widget implementation languages are supported, plus 3rd party widgets (e.g., Google Gadgets)
- virtual desktops provide multiple desktops, each devoted to particular app, avoiding need to be constantly minimizing and unminimizing windows
- activities can be defined and used to change desktop layout for different purposes

KDE4 Desktop

Standard KDE Desktop:
KDE4 Desktop (contd.)

Important desktop elements:

- **menu** – find and launch apps
- **launcher** – easily find/start popular apps
- **task manager** – select among running apps (right-click for options), classic vs icon-only managers available
- **pager** – select active virtual desktop
- **system tray** – for select apps, iconified
- **panel** – contains all of the above plasmoids
- **applets/widgets** – special apps, can be on desktop or panel
- **widget tool box** – add, lock/unlock, control applets/widgets

KDE System Settings

KDE System Settings:

KDE Applications

Important/popular KDE applications:

- **Internet**
  - Konqueror – web and file browser
  - Kmail – email program
  - KTorrent – BitTorrent client

- **Graphics**
  - Okular – document (e.g., PDF) viewer
  - GWenview – photo/image viewer/browser
  - digiKam – photo downloading and management
  - Krita – bitmap image editing (i.e., paint)
Important/popular KDE applications (contd.):

- **Audio & Video**
  - Amarok, Juk – audio/music players
  - Dragon Player – video player
  - Kaffeine, KPlayer – media players
  - KsCD – CD player
  - KAudioCreator – CD ripper

- **Personal Information Management (PIM)**
  - Kontakt – personal information manager
  - Kmail – email program
  - KOrganizer – calendar
  - KAddressBook – manager contacts

---

**KDE: Konqueror**

The KDE web browser and advanced file browser:

- understands many protocols via **kio slaves**: http, file, sftp, smb, etc.
- can browse both local and remote files
- highly configurable in how files are viewed
- can bookmark both web sites and local/remote files
- can associate actions with **mime types** (e.g., show in text previewer, open in editor, play in media player)
- right-click for service menus (actions for types)
- can change change permissions and other file properties

---

**Konqueror (contd.)**

Example Konqueror, browsing local + remote files:
Linux Interfaces 4: GNOME

1. CLI vs GUI
2. GUI Elements
3. KDE
4. GNOME
   ● GTK+ toolkit
   ● GNOME shell
   ● activities overview
   ● applications view
   ● dissatisfaction with GNOME 3
   ● classic mode
   ● GNOME applications
5. Alternatives and 3rd-Party Applications

GNOME

Part of the GNU project.

Based on GTK+ toolkit, written mainly in C.

 GNOME 2 provided a traditional desktop environment.

 GNOME 3 was a complete redesign.

 GNOME shell is the GNOME 3 interface:
   ● activities overview component used for switching between
tasks and workspaces
   ● dock allows quick app switching and launching

Distros primarily using GNOME: RedHat/CentOS, Fedora,
Ubuntu (underlies Unity), Mint (underlies Cinnamon)

---

GNOME Shell

GNOME 3 interface is called GNOME Shell:

---

GNOME Shell (contd.)

Important desktop elements:

   ● panel – along screen top, contains the below items
   ● Activities – switch to activities overview
   ● app menu – menu of actions for active (in-focus) application
   ● system menu – logout/shutdown, etc.
** GNOME Shell (contd.) **

** GNOME 3 Shell, annotated:**

** GNOME Activities Overview **

Important **activities overview** elements:

- **dash** – laucher for select apps
- **search** – search for apps
- **workspaces** – select workspace (virtual desktop)
- **notifications/message tray** – displays notifications, as overlay

** GNOME Activities Overview (contd.) **

** GNOME 3 **activities overview, annotated:**

** GNOME Applications View **

** GNOME 3 applications view:**
GNOME Dissatisfaction

Though GNOME claims redesign of GNOME 3 focused on enhancing usability, the new interface has been controversial:

- users found it unfamiliar, not always as usable as the traditional desktop style
- not very configurable, many options were removed
- requires use of compositing window manager, which requires hardware accelerated graphics

Bottom line: users don’t always like what they are told by “experts” they should like, and no one interface is likely to be best/preferred for different types of users.

Sound familiar? Windows 8??

GNOME Classic Mode

Beginning with GNOME 3.8, GNOME provides **Classic Mode**, for a traditional desktop experience (similar to GNOME 2):

GNOME Classic Mode (contd.)

**GNOME Classic Mode** applications menu:

GNOME Applications

Important/popular GNOME applications:

- **Internet**
  - **Epiphany** – web browser
  - **Evolution** – email client

- **Graphics**
  - **Evince** – document viewer
  - **GIMP** – GNU Image Manipulation Program (like Photoshop)
  - **Inkscape** – vector drawing program (like Adobe Illustrator)
  - **Eye of GNOME (eog)** – image viewer
  - **gThumb** – image viewer/browser
  - **Shotwell** – photo manager
Important/popular GNOME applications (contd.):

- **Audio & Video**
  - Rythmbox – music player
  - Totem – video player
  - Banshee – media player (like Window Media Player)
- **Personal Information Management (PIM)**
  - Calendar – calendar appl
  - Evolution – email client

- **Office**
  - Abiword – word processing
  - Gnumeric – spreadsheet program

- **Tools**
  - Nautilus – file manager
  - Gedit – text editor
  - Terminal – terminal/shell
  - Brasero – CD/DVD burning

- **Development**
  - Gedit – text editor
  - Anjuta – integrated development environment (IDE)

### GNOME Core Applications

GNOME 3 includes a set of so-called **Core Applications** that are tightly integrated with the GNOME Shell, including:

- Calculator – calculator
- Contacts – manage contacts
- Documents – manage document (including cloud)
- Files – file browser
- Mail – email client
- Media – media player
- Music – music player
- Photos – access and organize photos
- Web – web browser
Linux Interfaces 5: Alternatives and 3rd Party

1. CLI vs GUI
2. GUI Elements
3. KDE
4. GNOME
5. Alternatives and 3rd-Party Applications
   - GNOME alternatives
   - Ubuntu Unity
   - non-DE applications
   - third-party software

Dissatisfaction with GNOME 3’s interface led (at least partly) to the development of several alternative interfaces:

- **Unity** – Ubuntu desktop for GNOME 3
- **Cinnamon** – traditional desktop for GNOME 3
- **MATE** – fork of GNOME 2 (to retain GNOME 2)

Ubuntu developed Unity in part because of issues with GNOME 3’s interface, but also for technical reasons and to make its interface more uniform across different screen size devices.

Negative reaction to *Unity* then helped to push the popularity of Ubuntu-based alternative distros like *Linux Mint*, which uses Cinnamon and MATE as its DE’s.

Ubuntu Unity

Ubuntu’s Unity desktop is an alternative interface to GNOME 3:

Ubuntu Unity Dash

The top launcher button invokes what is called the **dash**, which can be used to locate applications:
Non-DE Application Software

While much of the basic GUI application software supplied with a distro will be from a DE like KDE/GNOME, a large amount of non-DE GUI software is also supplied with most distributions.

(Of course, distributions supply vast amounts of library and utility software that is required by the GUI “front end” applications.)

Some important non-DE software:
- Infrastructure: Java (machine)
- Web Browsers: Mozilla Firefox, Google Chrome
- Office Suites: Apache OpenOffice, LibreOffice
- Multimedia Players: Adobe Flash Player, VLC player
- Editors: Emacs, Vi
- Development: Eclipse IDE
- Emulators: VirtualBox

Third-Party Software

In the context of Linux, the term “third-party software” refers to software that is not provided by the distribution.

Use of third-party software has many disadvantages:
- cannot be installed using distro software tools
- will not be automatically updated by distro
- will be more work to download and install, may have to be built first
- can compromise system security unless precautions are taken

This is why it is good to use large, popular distros, since they are more likely to supply all the software you will need.

Third-Party Software (contd.)

Distributions with large software repositories:
Ubuntu (and its derivatives like Mint), Debian, Fedora, Mageia, OpenSUSE.

Another advantage of using a popular distro is that third-party software is more likely to be tested and even packaged for that distro.

E.g., VirtualBox has packages for several different distros (in addition to a general installer for all Linuxes).

VBox officially supports their Guest Additions for particular distros only: Oracle Linux, Fedora, RedHat, SUSE/openSUSE, and Ubuntu (though it works for many others).