A large, faded, and slightly tilted Windows logo is visible in the background on the left side of the slide. It consists of four panes in a cross shape, with a grid of small squares extending from the top-left pane.

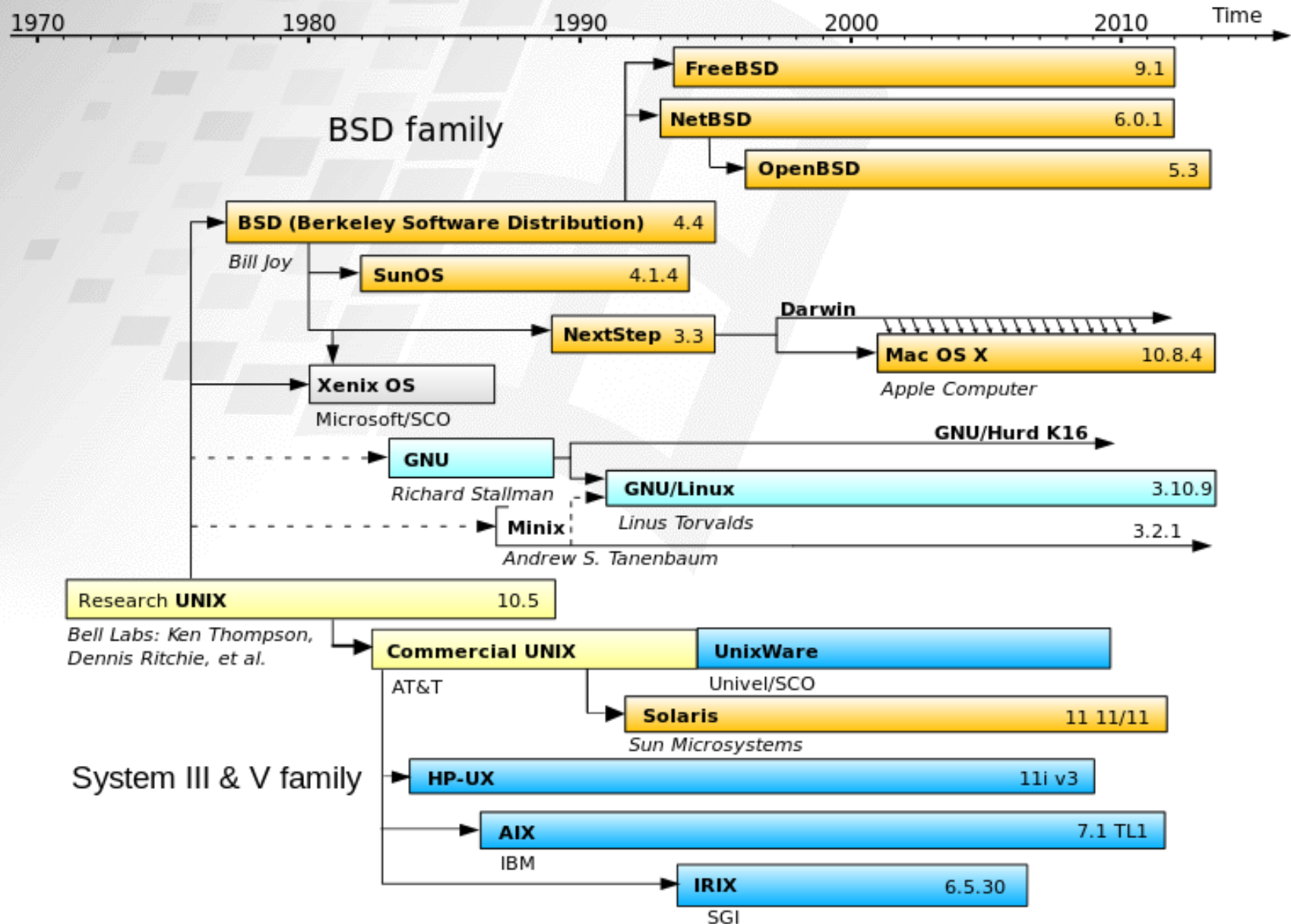
Windows vs Linux

Operating Systems

A large, faded, and slightly distorted Windows logo is positioned in the upper-left quadrant of the slide. The logo is composed of four panes, each containing a grid of squares. The squares vary in shades of gray, creating a 3D effect. The entire logo is semi-transparent, allowing the white background to show through.

The Linux OS

History of OS



History of Linux

- Open Source, Free Software
 - FOSS
- Linus Torvalds 1991
- Based on MiniX + GNU
- Still directs development of kernel
 - Kernel.org



Linux Architecture

Interactive User



Libraries

Commands

Application
Programs

...



Shell

OS System Call Interface

Trap Table

Device Driver

Device Driver

...

Device Driver

Driver Interface

Monolithic Kernel Module

- Process Management
- Memory Management
- File Management
- Device Mgmt Infrastructure

Linux Desktop GUIs

Sample Graphical Desktop Environments



GNOME Shell (GNOME 3)



KDE Plasma (KDE 4)



Unity



Xfce



LXDE



Enlightenment



Cinnamon



MATE (GNOME 2)



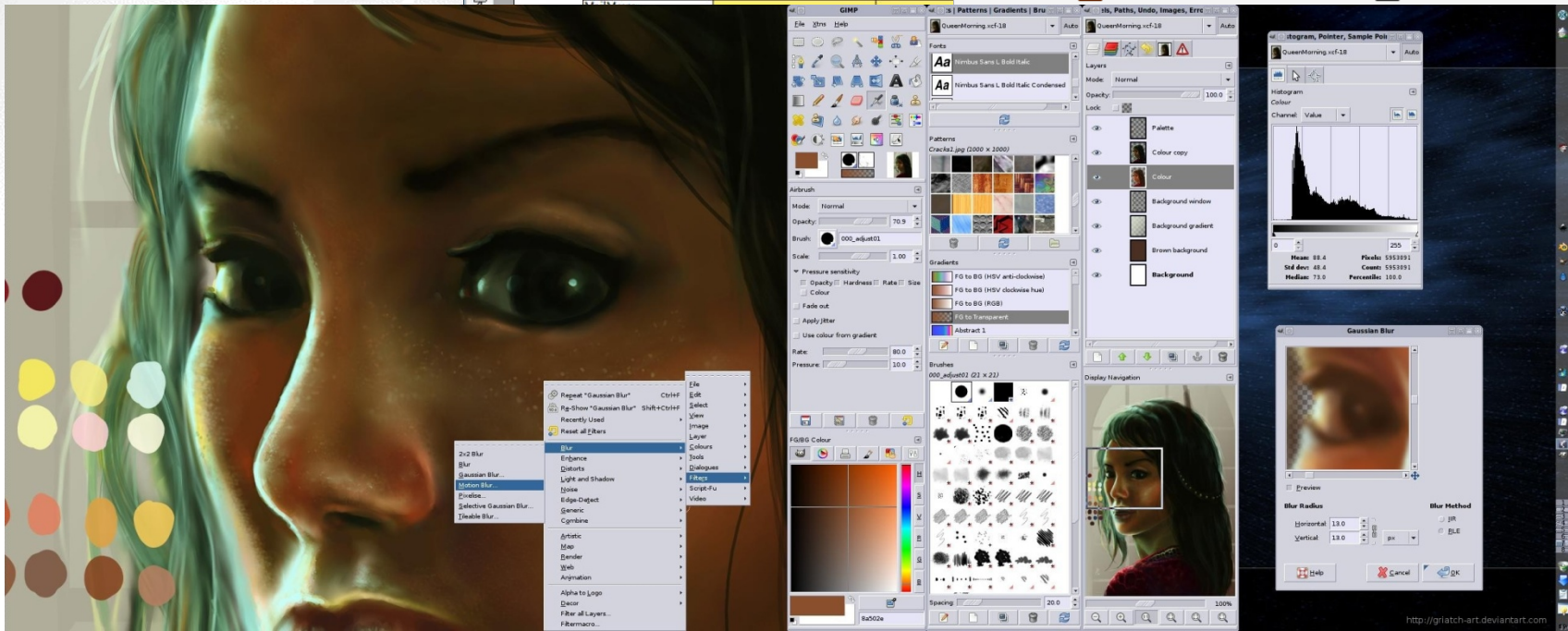
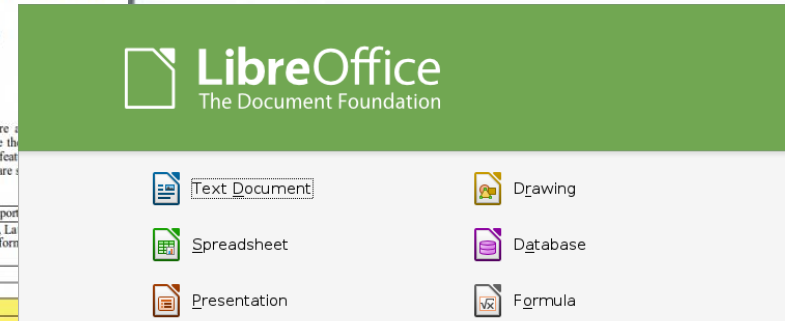
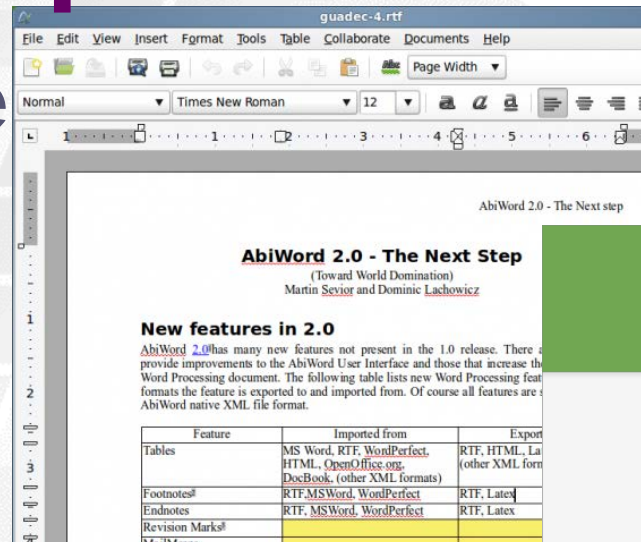
Trinity (KDE 3)



Sugar

Linux Applications

- OpenOffice
- LibreOffice
- AbiWord
- GIMP



Linux Applications

Shooter Linux Games

Team Fortress 2



Join either the red or blue team and play different game modes in this free-to-play cartoony shooter.

[Go to](#)

Scorched 3D

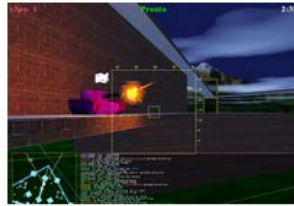


A 3D version of Tanks played out in a massive world with cool visual effects.

[Read More](#)

[Go to](#)

BZFlag



A 3D multiplayer tanks Linux game with less mountainous maps than Scorched 3D.

[Read More](#)

[Go to](#)

Simulator Games

Sports Linux Games

Extreme Tux Racer



Race a penguin down an icy slope to earn the fastest time.

TORCS



A fast and fun racing simulator with 3D graphics.

[Go to](#)

OpenBVE



An highly customizable train simulator for support with multiple maps and trains.

FlightGear



A highly sophisticated flight simulator with a worldwide map and custom planes.

Why Linux

- Used in many consumer device

- Toys / Household Gadgets

- Robotic Vacuum, Pets, Fridge, Cameras, Watches

- Navigation

- [TomTom GPS](#)

- Routers / Gateways

- [Linksys WRT54G series](#)

- Tablets / Smartphones

- [Motorola](#) EXZ series
 - [Openmoko](#)
 - [Nokia N900](#), [N9 cell phones](#)
 - [Android](#)

- The first device shipping with the Android operating system was the [HTC Dream](#), which was released on 22 October 2008.



Linux Today

■ Popular Distros

● Server

- RHEL
- CentOS
- OpenSUSE / SUSE
- Ubuntu

● Desktops

- Ubuntu
- Fedora
- Mint
- Arch Linux
- PCLinux

● Media

- XBMC Live
- Mythbuntu / UbuntuTV / Ubuntu Studio
- Arch Linux

● Gaming

● Rescue

- Puppy
- Ultimate CD
- Hiren
- SystemRescueCD

● Embedded

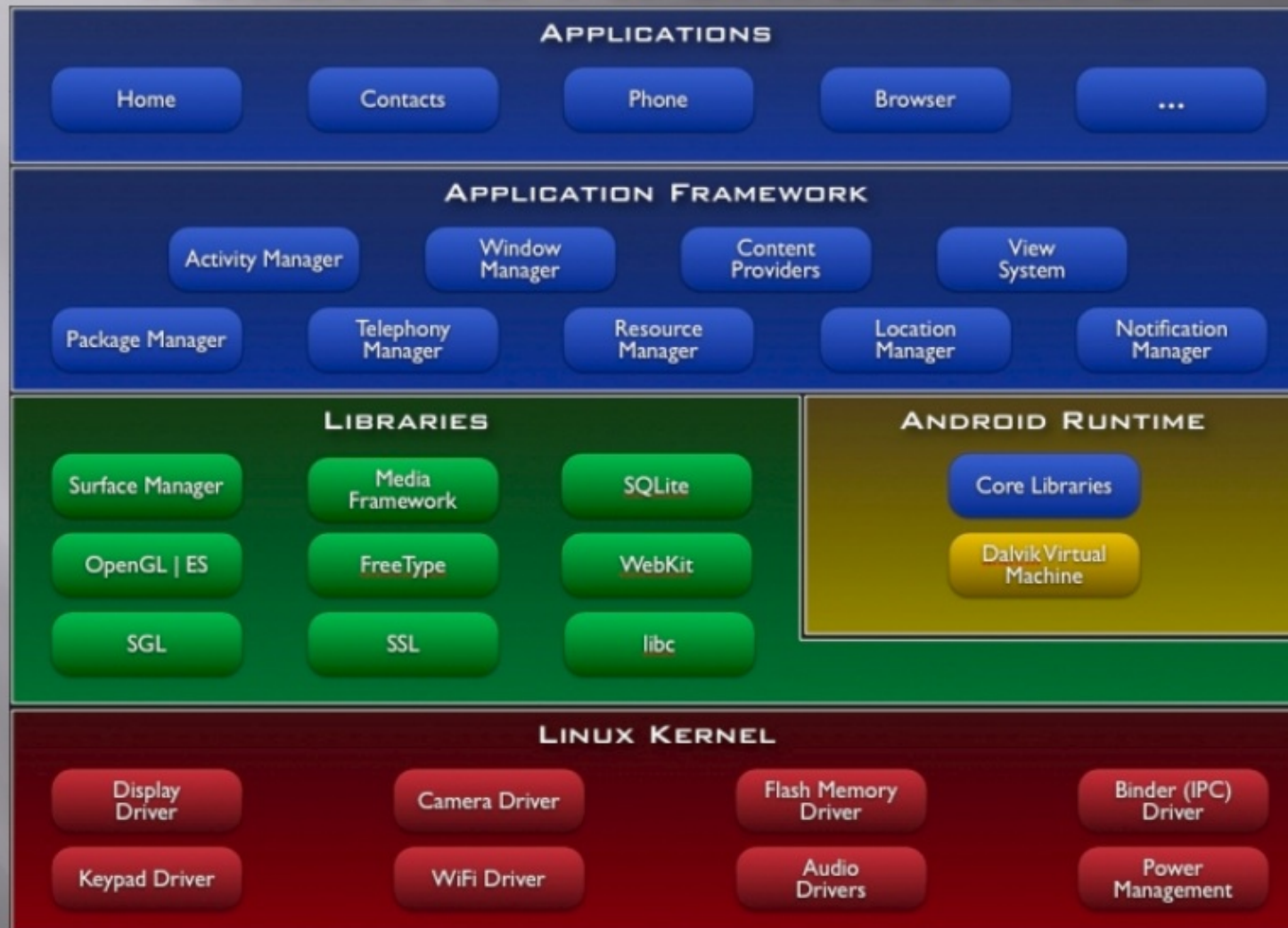
- Android

Linux Today - Embedded

- The Blackfin uClinux Distribution by Analog Devices - Blackfin processors
- Embedded Alley
- KaeilOS
- Lineo Solutions uLinux
- MontaVista Linux
- Pengutronix OSELAS.BSP()
- RidgeRun Linux
- TimeSys LinuxLink
- Ubuntu Mobile
- Wind River
- Little Blue Linux - MPC Data
- Digi Embedded Linux for Digi's ARM based modules
- Raspbian (Raspberry pi)

Linux Today – Embedded Android

Android Architecture



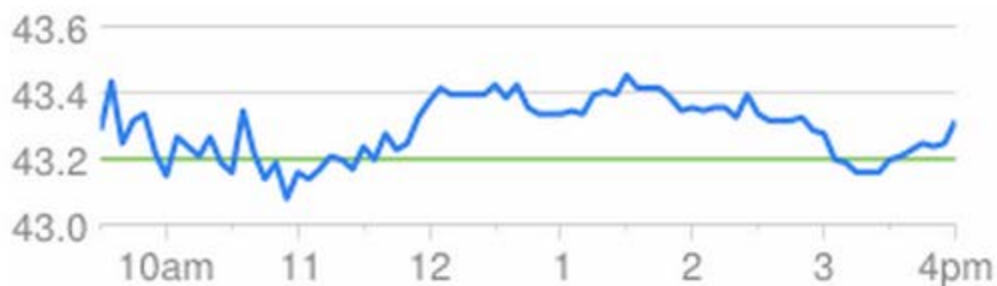
Why Red Hat?



Red Hat Inc

NYSE: RHT - Nov 8 4:02 PM ET

43.32 +0.12 (0.28%)



Open	43.29
High	43.47
Low	43.05
Volume	1,057,754
Avg Vol	2,166,000
Mkt Cap	8.21B

1 d

5 d

1 m

6 m

1 y

5 y

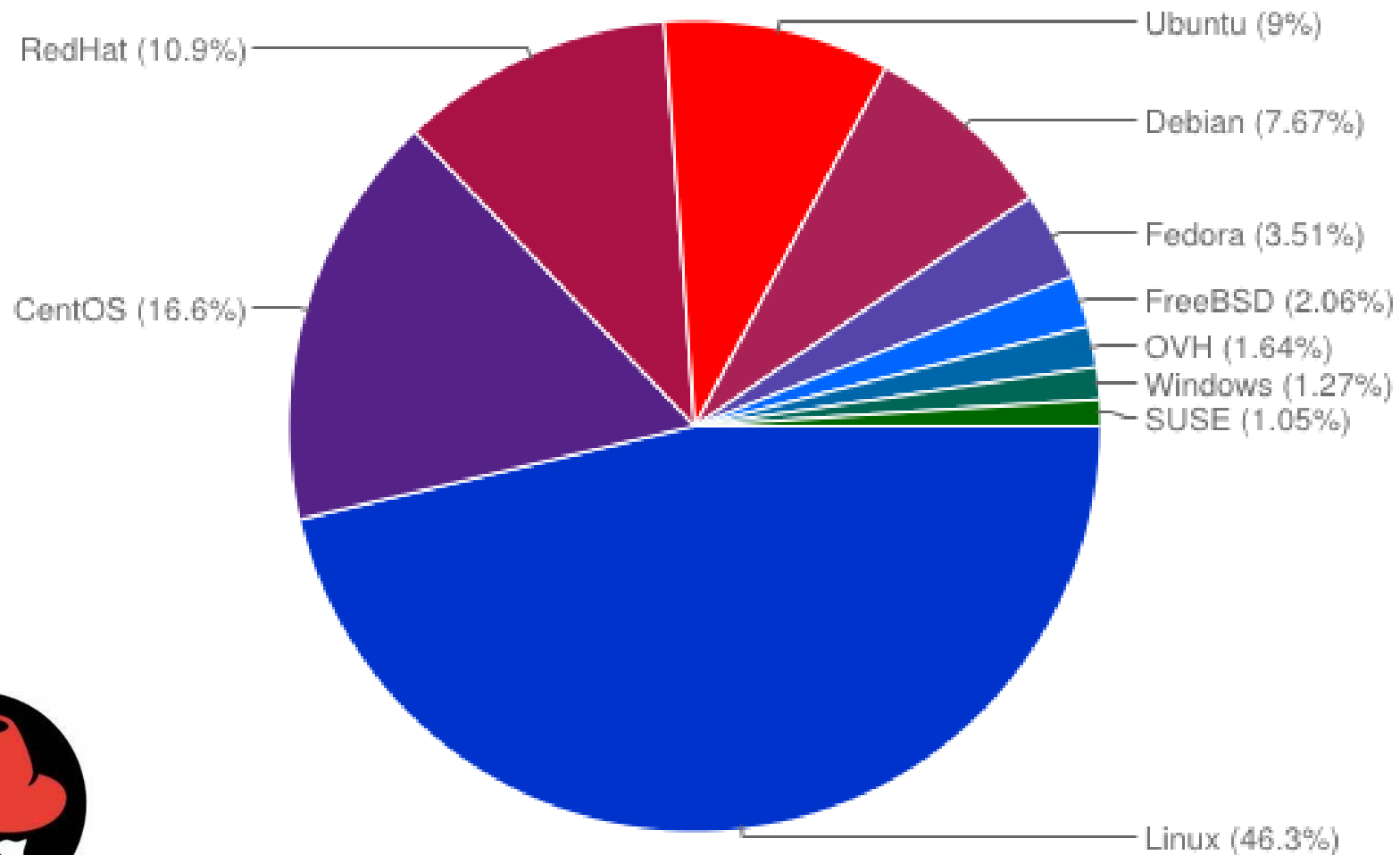
max

Red Hat



- Multi-Product Multi-Billion\$ Company
 - Operating Systems
 - JBoss Enterprise Middleware
 - Systems and Identity Management
 - Infrastructure products and distributed computing (Cloud + Data Centres)
 - Training, consulting, and extended support
- Fedora Core
 - Open, community-supported, proving grounds for technologies which may be used in upcoming enterprise products
- Customised Kernel
- Virtualisation Hypervisor (Embedded Virtualisation)
 - Xen (up to 2010)
 - KVM

Red Hat



Red Hat Certified System Administrator (RHCSA)

- Diploma Plus (Dip+)
- ITA311 + ITA312
- 8 + 8 days (during semester breaks)
- Subsidised Exam Cost

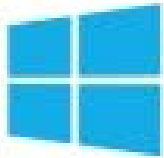


The Windows OS

Objectives

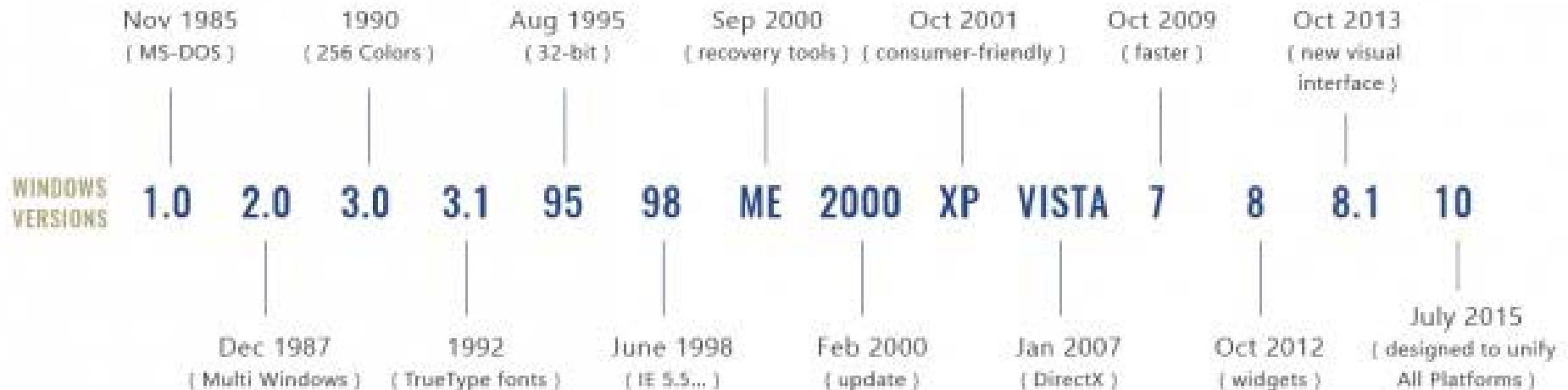
- Brief History of Windows
- Examining the structure of the Windows OS
- Processes and Threads
- API, Library, DLL

History



Windows OS History Timeline

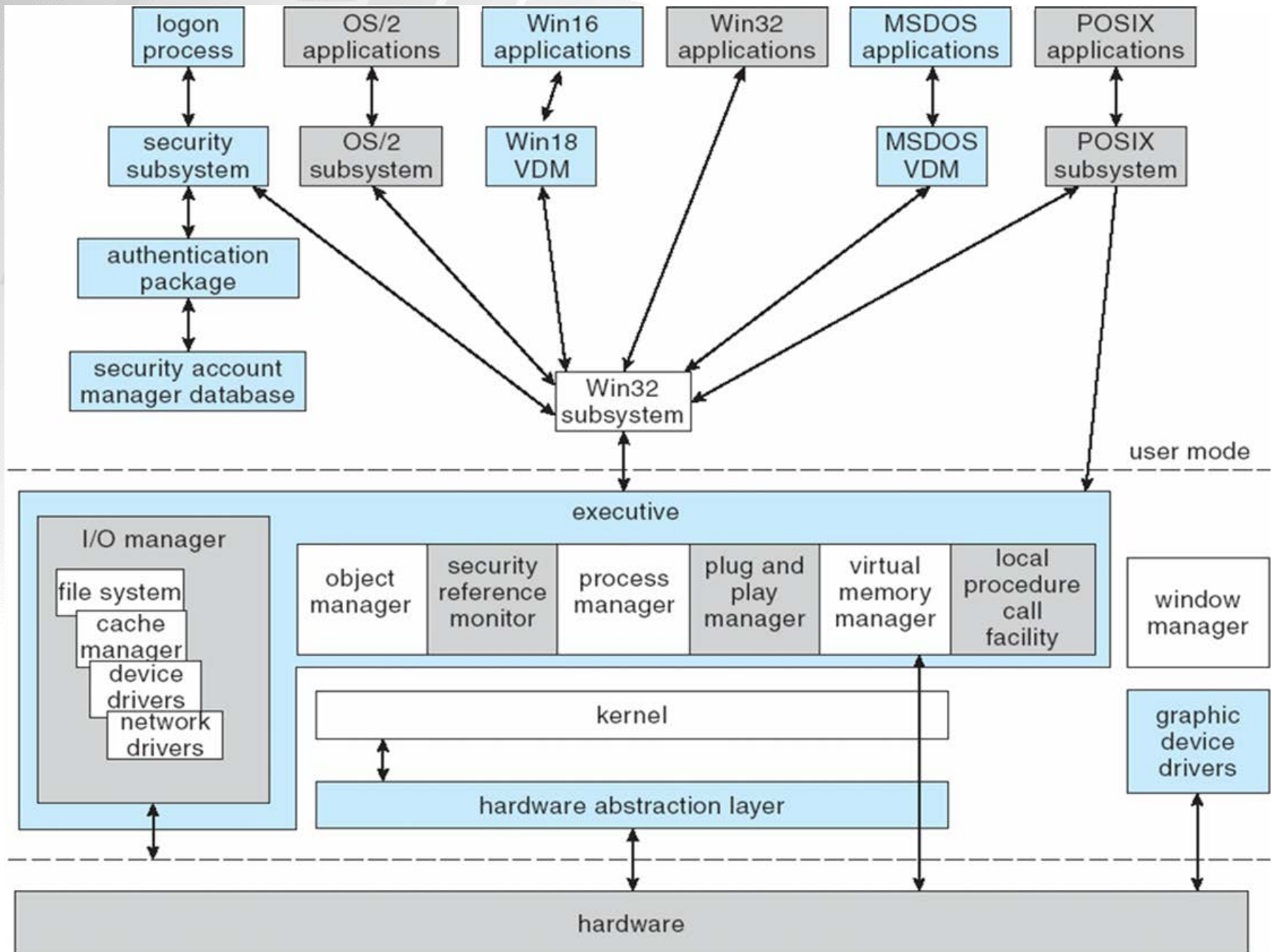
Microsoft was founded by Bill Gates and Paul Allen on April 4, 1975, to develop and sell BASIC interpreters for Altair 8800. Microsoft introduced an operating environment named Windows on November 20, 1985.



Versions and Editions

Versions	Editions
Windows XP	Starter, Home, Professional, Tablet, Media
Windows 7	Starter, Home Basic, Home Premium, Professional, Enterprise, Ultimate
Windows 8	Windows 8, Professional, Enterprise, Windows RT

Win 7 Architecture



Design Principles

- Layered Architecture
 - Kernel layer runs in protected mode and provides access to the CPU by supporting threads, interrupts, and traps.
 - Executive runs in protected mode above the Kernel layer and, provides the basic system services
 - On top of the executive, environmental subsystems operate in user mode providing different OS APIs
 - Modular structure allows additional environmental subsystems to be added without affecting the executive
- Portability —Windows 7 can be moved from one hardware platform to another with relatively few changes
 - Written in C and C++
 - Platform-dependent code is isolated in a dynamic link library (DLL) called the “hardware abstraction layer” (HAL)

Windows Architecture

- Layered system of modules
- Protected mode — **hardware abstraction layer (HAL)**, kernel, executive.
 - Executive includes file systems, network stack, and device drivers.
- User mode — collection of subsystems, services, DLLs, and the GUI
 - Environmental subsystems emulate different operating systems
 - Protection subsystems provide security functions
 - Windows services provide facilities for networking, device interfaces, background execution, and extension of the system
 - Rich shared libraries with thousands of APIs are implemented using DLLs to allow code sharing and simplify updates
 - A graphical user interface is built into Win32 and used by most programs that interact directly with the user

Architecture – implications

- Windows XP and newer architecture is the key to its:
 - Reliability
 - Scalability (Professional, Server, Advanced Server, Datacenter Server)
 - Security
 - Portable (runs on Intel AND other platforms)
- Windows Me, 9x, and 3.x *do not* have this type of architecture

So how does it all work?

- Let's start by defining some terms...
 - Program
 - Process
 - Thread

Definitions (program)

- Program
 - Also known as an *application*
 - It is...
 - The software stored on disk or other media
 - Here we mean the program “Microsoft Word” (i.e., the one you could buy)

Definitions (process)

- Process

- A *program* that has been **loaded** from long-term storage (e.g., hard drive) into memory by the OS and is being run
- It includes...
 - System resources it needs to run (e.g., RAM, etc.)
 - One or more *threads*

Definitions (thread)

- Thread

- A component (or part) of a *process*
- Or, a single unit of executable code
- The C programs you are writing in IPC are an example of a single threaded program
 - Larger programs tend to use multiple threads.



So it's built in layers and there are lots of threads, but how does the OS actually make my programs work?

Answer: APIs and Libraries

Definitions

- Let's define some more terms:
 - API (Application Programming Interface)
 - Library
 - DLL (Dynamic Link Library)

API

- Application Programming Interface
- A set of pre-made programming functionalities and tools (a library) for building software applications.
- APIs make it easier to develop programs by providing all the building blocks a programmer needs to create complex programs.
- Because all programmers use these APIs, users get programs that look and feel like each other.
- The Windows APIs are stored in libraries

Example API: English vs. XP

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y
Z				

Alphabet

All words
must have
one vowel

**Rules for
Making Words**

apple
Cat
woman
is

Words

subject
verb
object

Capitalization

punctuation
rules

Grammar

Novel

News-
paper

Web
Page

Writing

Microkernel

Native API
(Low-level
API)

Executive
Services

Win32 API
(High-level API)

32-bit
Windows
Applications

Libraries

- We've all been to a library, but what is a library in programming?
 - A collection of precompiled routines or functions that a program can use.
- We put commonly used routines in a library so we don't have to re-write them
 - Example: sorting a list of numbers
- Windows uses a special kind of library called Dynamic Link Libraries

Dynamic Link Libraries (DLL)

- A DLL is: A **library** of executable **functions** or **data** that can be used by a Windows application. Example: user32.dll, kernel32.dll
- DLLs provide one or more functions that a Windows program accesses by creating a link to the DLL.
 - The word “Dynamic” means that the link is created whenever the function or data is needed (i.e., while the program is running) instead of being linked at compile time
- Other DLLs are written for a particular application and are installed with the application (this is why we need to install!)
 - Spellchecker in MS Office is the same for Word, Excel, Power Point, etc. The DLL that contains this functionality is msp232.dll.

APIs and DLLs

- We said the Windows APIs were stored in libraries. There are 4 main library files:
 - The Native API (kernel level functions) is stored in a file called **ntdll.dll**. The Win32 API libraries make use of this file to do things with hardware
 - The Win32 API is split between 3 files:
 - **kernel32.dll** - File I/O (CreateFile()), thread management, etc.
 - **user32.dll** - Window (e.g., CreateWindow()) and Event Messaging (e.g., mouse-clicks) functions
 - **gdi32.dll** - Drawing functions to actually draw the windows we see on the screen (e.g., LineTo())