Introduction to Linux

Rakesh Basavaraju

Agenda

- ✓ History and introduction
- ✓ Linux Variants
- ✓ Linux Features
- ✓ Linux Architecture
- ✓ Linux kernel
- ✓ Linux Modes
- ✓ Packaging

History and Introduction

Before Linux

- In 80's, Microsoft's DOS was the dominated OS for PC
- Apple MAC was better, but expensive
- UNIX was much better, but much, much more expensive. Only for minicomputer for commercial applications
- People was looking for a UNIX based system, which is cheaper and can run on PC
- Both DOS, MAC and UNIX were proprietary, i.e., the source code of their kernel is protected
- No modification is possible without paying high license fees

GNU project

- GNU is a recursive acronym for “GNU's Not Unix”
- GNU provides lot of tools, applications, libraries, License
- Established in 1984 by Richard Stallman, who believes that software should be free from restrictions against copying or modification in order to make better and efficient computer programs.
- Aim at developing a complete Unix-like operating system which is free for copying and modification
- Companies make their money by maintaining and distributing the software, e.g. optimally packaging the software with different tools (Redhat, Slackware, Mandrake, SuSE, etc)
- Stallman built the first free GNU C Compiler in 1991. But still, an OS was yet to be developed

Open Source Software

Open Source Software (OSS) generally refers to software for which the source code is available and which the licensing scheme permits the user to modify it and redistribute it in modified or unmodified form.

In a nutshell, the GNU General Public Licence (GPL) allows anybody to:

- use the software at no charge, without any limitations,
- copy, and distribute or sell unmodified copies of the software in the source or binary form,
- use the software with proprietary (e.g., your own) modifications, free of charge, as long as you do not distribute or sell the modified version,
- modify, and distribute or sell a modified version of the software as long as the source code is included and licenced on the same terms as the original you received (the GPL),
- sell support for the software, without any limitations.

Beginning of linux

- A famous professor Andrew Tanenbaum developed Minix, a simplified version of UNIX that runs on PC
- Minix is for class teaching only. No intention for commercial use
- In Sept 1991, Linus Torvalds, a second year student of Computer Science at the University of Helsinki, developed the preliminary kernel of Linux, known as Linux version 0.0.1

GNU/Linux Licensing

- Linus published first Linux under shared source license
- Most of tools are under GNU Public License
- Linux 0.99 is published under GNU General Public License (GNU GPL)
- Linus: “making Linux GPL'd was definitely the best thing I ever did.”

Message from Professor AndrewTanenbaum

" I still maintain the point that designing a monolithic kernel in 1991 is a fundamental error. Be thankful you are not my student. You would not get a high grade for such a design :-

(Andrew Tanenbaum to Linus Torvalds)

- Soon more than a hundred people joined the Linux camp. Then thousands. Then hundreds of thousands
- It was licensed under GNU General Public License, thus ensuring that the source codes will be free for all to copy, study and to change.

Linux Today

- Linux has been used for many computing platforms
- PC, PDA, Supercomputer, …
- Not only character user interface but graphical user interface is available
- Commercial vendors moved in Linux itself to provide freely distributed code. They make their money by compiling up various software and gathering them in a distributable format
- More than 3 major desktops

  GNOME, KDE, Xfce
- More than 5 major shells

  Bash, csh, tsh, …
- Complete set of compilers

  C, C++, java, Fortran, Python, Ada, …
- Many network services

  Web, Email, File Sharing, DNS, FTP, SSH, …
- Many user applications

  OpenOffice, Web browser, Latex, multimedia,

What is Linux

- Developed in 1991 by a University of Finland student Linus Torvalds.
- Basically a kernel, it was combined with the various software and compilers from GNU Project to form an OS, called GNU/Linux
- Linux is a full-fledged OS available in the form of various Linux Distributions
- RedHat, Fedora, SuSE, Ubuntu, Debian are examples of Linux distros
- Linux is supported by big names as IBM, Google, Sun, Novell, Oracle, HP, Dell, and many more

Linux variants

- Red Hat Linux : One of the original Linux distribution.

  The commercial, nonfree version is Red Hat Enterprise Linux, which is aimed at big companies using Linux servers and desktops in a big way.
- Debian GNU/Linux : A free software distribution. Popular for use on servers. However, Debian is not what many would consider a distribution for beginners, as it's not designed with ease of use in mind.
- Ubuntu: based on Debian Linux. Ubuntu claims to be most popular desktop version. Many applications and excellent “update mechanism” contribute to its sucess. Revenue is created by selling technical support.
- SuSE Linux : SuSE was recently purchased by Novell. This distribution is primarily available for pay because it contains many commercial programs, although there's a stripped-down free version that you can download.
- Mandrake Linux : Mandrake is perhaps strongest on the desktop. Originally based off of Red Hat Linux.
- Gentoo Linux : Gentoo is a specialty distribution meant for programmers.

Linux User interface

- Can be controlled through command-line (CLI) or Graphical User Interface (GUI)
- GUI run through Desktop Environments (DE)
- KDE, GNOME, Xfce, E17 are popular Des
- The GUI interface is easy-to-use and much like that of Windows and Mac OSX
- The CLI is similar to that of UNIX/BSD

Linux Features

- Linux is Free and source code is open and available
- Linux has high standard for source code quality
  - This feature makes Linux system more stable
- Linux kernel can be very small and compact
- Linux fully customizable with all its components
  - Source code can be modified and can be fit into any system to optimize the performance
- Linux runs on low-end hardware platform
  - Like less memory, low speed etc.,
- Linux is powerful
  - Linux systems are very fast, since they fully exploit the features of hardware components
  - The main linux target is efficiency
- Linux is compatible with many other common OSsystems API's
  - Linux can directly mount file systems.
- Linux is able to operate with many network layers like
  - Ethernet, token ring, fiber distributed data interfaces etc.,
- Linux can run programs written for other OSusing suitable libraries

Linux Features

- Linux is well supported
  - Thousands of developers are available around the world
- Linux includes all features of modern OS
  - Virtual memory
  - Virtual file systems
  - Lightweight processes
  - Reliable signals
  - IPC (Inter Process Communications)
  - Multiprocessor systems
- Monolithic and modular
- Kernel threading
- Can be independently scheduled
- Context switch between kernel threads are much less expensive
- may or may not associate usermode programs

Programming in Linux

- Modern languages are cross-platform, like Python, Ruby, Perl, Java
- Most Linux distros support these languages and have their runtimes pre-installed
- GTK+ and Qt are widely used to design applications for Linux
- IDEs like NetBeans, Anjuta, KDevelop, MonoDevelop, Eclipse are available for Linux too

On Desktop

- Linux is desktop computer ready
- Large number of distros targeted at Desktop users are available
- Linux desktop distros come with many commonly used pre-installed software
- The modern Linux interface is user-friendly and makes the interaction with computer easy

Server and Gaming on Linux

- Many native games are available, both 3D and 2D
- Popular games for Linux are: Quake, Unreal Tournament, Counter Strike, Doom, Cube, CodeRED, Wesnoth, OpenArena, SuperTux, Frozen Bubble, Medal of Honor, and many more.

- Linux is the most used OSon servers
- Linux is the cornerstone of the LAMP server-software combination (Linux, Apache, MySQL, Perl/PHP/Python) which has achieved popularity among developers

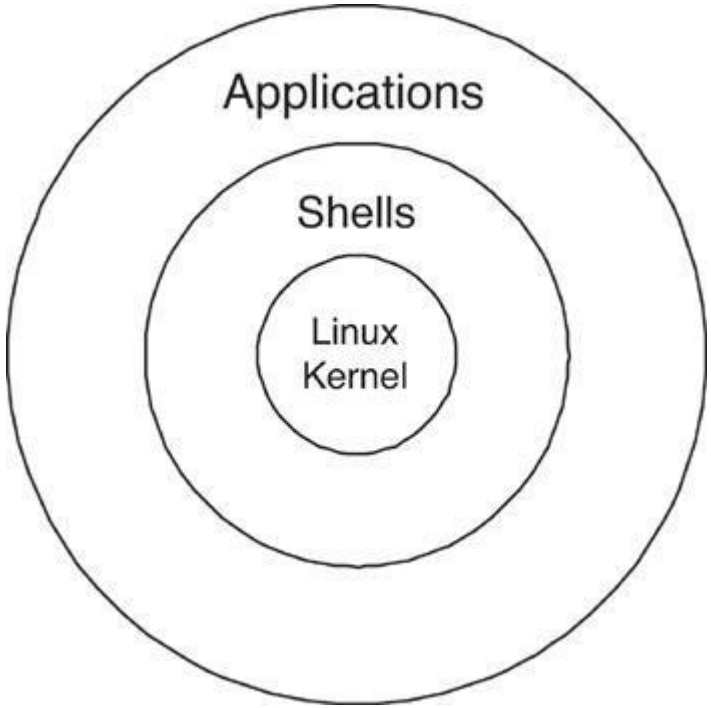
Why Should I use Linux

- No threat of viruses
- Linux systems are extremely stable
- Linux is Free
- Linux comes with most of the required software pre-installed
- Update all your software with minimum fuss
- Linux never gets slow
- Linux does not need defragmentation
- Linux can even run on oldest hardware
- Adding more software is a matter of a few clicks
- Most Windows-only apps have their either their native version or alternatives for Linux
- With Linux, you get the highest degree of possible customizability

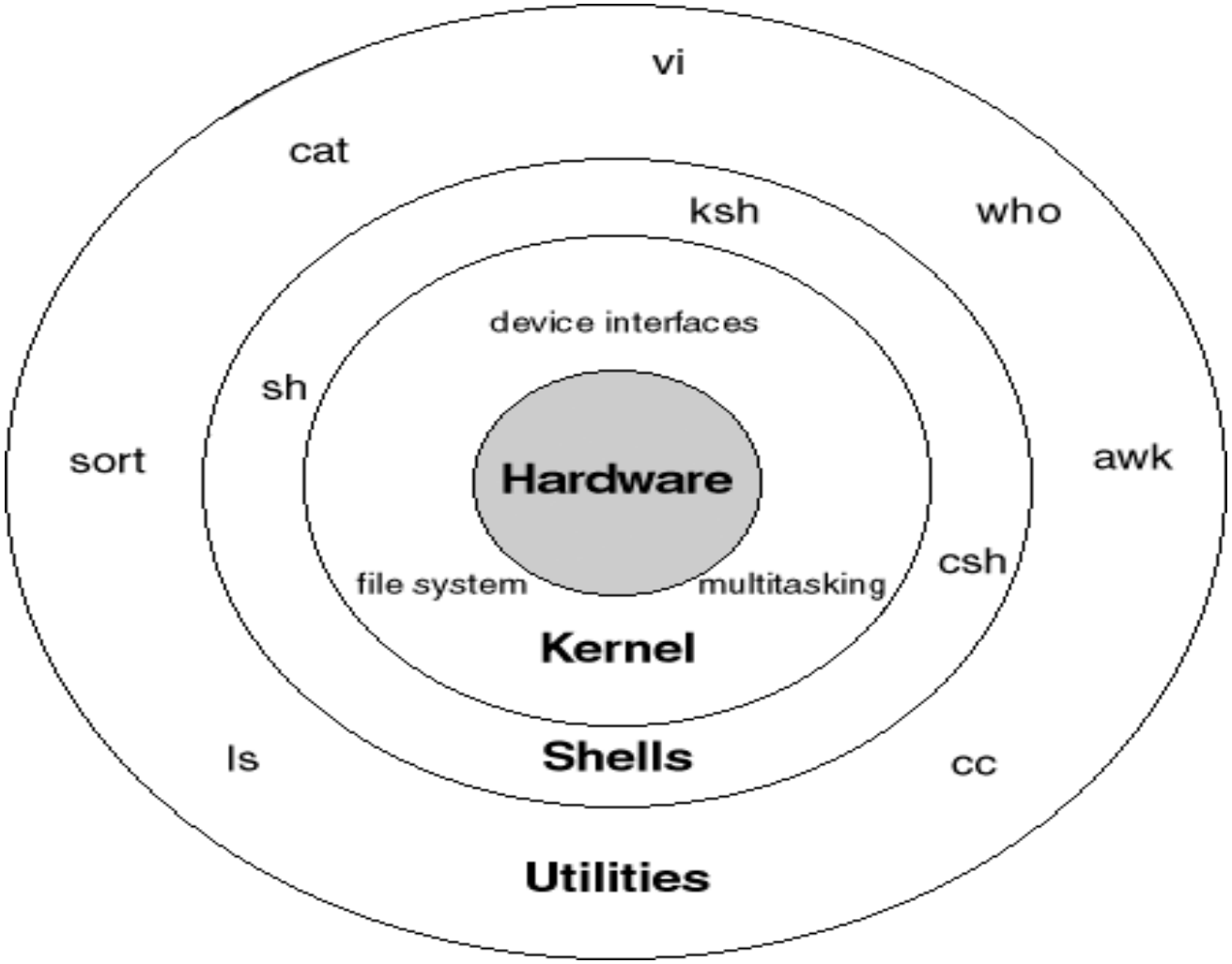
Linux Architecture

Linux architecture

- Components of Linux
  - Kernel
  - Shell
  - Application programs



Linux Architecture



Linux kernel

- The Linux kernel is currently maintained by Linus Torvalds and a few hundred other developers
- Releases are numbered in a very ordered fashion.

  Major.minor.patchlevel

Odd minor numbers are development kernels

Example: 3.10.101, 4.7-rc2,

The first number denotes the kernel *version*.

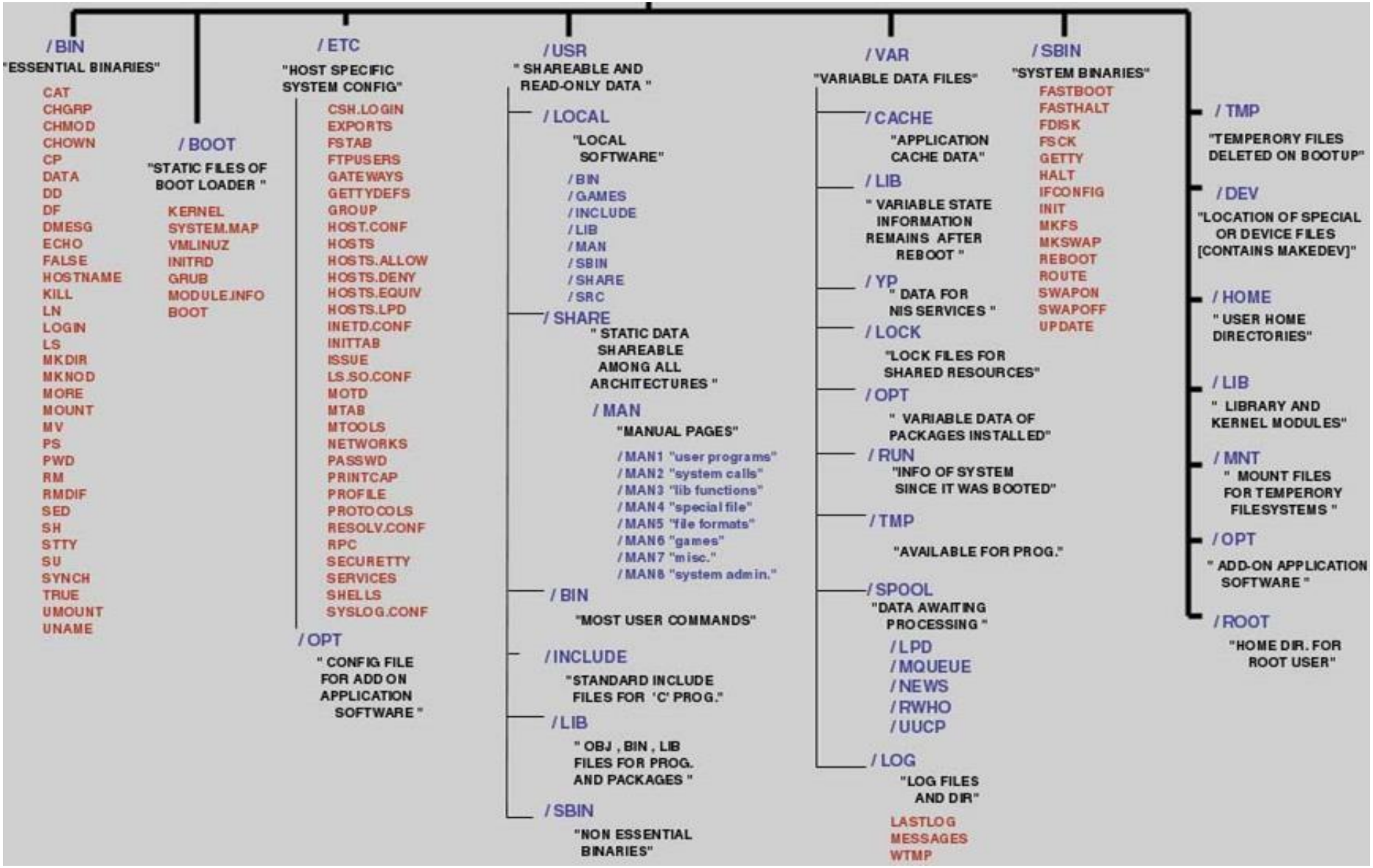
The second number denotes the *major revision* of the kernel version.

- odd numbers are development kernels.
- even numbers stable kernel

The third number indicates the minor revision of the kernel. It is only changed when new features or new drivers are added.

The fourth number represents corrections, such as security patches and bug (i.e., error) fixes. Sometimes the four numbers will be followed by several letters, such as rc1, ac, ck and mm. The letters rc refer to a release candidate and thus indicate a non-official release.

Linux File system



Linux modes

- Linux operates in two different modes
  - Kernel mode
  - User mode

Linux mode

- Kernel Mode
  - Kernel mode is also called as super user mode
  - All system programs work under kernel mode
  - User mode programs can not be executed directly under kernel mode
  - Kernel mode is also called as privileged mode
  - CPU must support privileged mode in the hardware level to support kernel

- User Mode:
  - User mode is a normal operation mode
  - All application programs runs under user mode
  - User mode programs can not directly access the kernel data structures
  - CPU provides special instructions to switch from user mode to super user mode and vice versa
  - Linux will transfer control from user mode to kernel mode using system calls. In other works entry points are called system calls.

Mode Switching

- A process in linux may always run under User mode or kernel mode
- If process runs kernel mode then the processor executes some kernel routine
- Transition between user mode and kernel mode may take place under following conditions
  - System calls
  - Scheduler
  - Interrupt handler

System call

- The System call
  - System call will transfer the control from user mode to kernel mode
  - User programs invoke the system call to switch to kernel mode
  - There are many system calls supported by linux for variety of operations
  - Depending on the service required proper system call handler must be used

Scheduler

- Scheduling the process
  - Scheduler is a part of Linux kernel, which schedules the process timings.
  - Timer interrupts are used to schedule tasks/ processes.
  - Timer interrupt swtiches the process from user mode to kernel mode by activating the scheduler

Interrupt handler

- Interrupts
  - Interrupts are the signal which causes the program interruption
  - Interrupts can be software interrupts or hardware interrupts
  - Interrupt switches the process from user mode to kernel mode
  - Under kernel mode interrupts service routine is serviced

Linux Components

- The Kernel
  - Kernel is the core of Linux OS
  - Kernel directly interacts with hardware
  - Kernel performs all the operations related to hardware
  - Kernel is hardware platform independent
  - Kernel always operates at super-user level
  - Kernel acts as an interface between application program and hardware

Linux Components

- The application programs
  - Application programs performs the actual operation on the target system
  - Application programs cannot interact directly with the hardware
  - Application programs request kernel functions to perform hardware related tasks
  - Application programs are always run under user mode

Linux Packaging

Packaging distribution formats

- There is no standard package manager in Linux
- Packages Distributed in Binaries or Source Code form
- Main Package Management Standards
  - RPM (RedHat Package Manager) (.rpm)
    - Introduced by RedHat and has been adopted by many other distributions (Fedora, Mandrake, SuSe) .
    - The most popular Linux package format
  - DEB (Debian Package Manager) (.deb)
    - Introduced by Debian distribution
  - Tarball files (.tar.gz/.tar.bz2)
    - The old-fashioned way of distributing software in Linux/Unix
    - Compatible with all distros

Red Hat Package manager

- Using the command line, packages are installed using rpm utility program
  - Install a package
    - rpm -i <package_name>.rpm
  - Update an existing package
    - rpm –U <package_name>.rpm
  - Remove a package
    - rpm –e <package_name>

Installing software in Debian-based distros

- Three ways to manage software packages in Debian

- dpkg: Used on .deb files like rpm
  - Install: dpkg -i <package_name>.deb
    - If an older version of the package is installed it updates it automatically by replacing it with the new
  - Remove: dpkg -r <package_name>

- dselect: dpkg console front-end

- apt-get: The most frequently used way of managing software packages in Debian.
  - Install: apt-get install <package_name>
    - e.g. apt-get install kde to install KDE Window Manager
  - Remove: apt-get remove <package_name>

Installing from tarball files

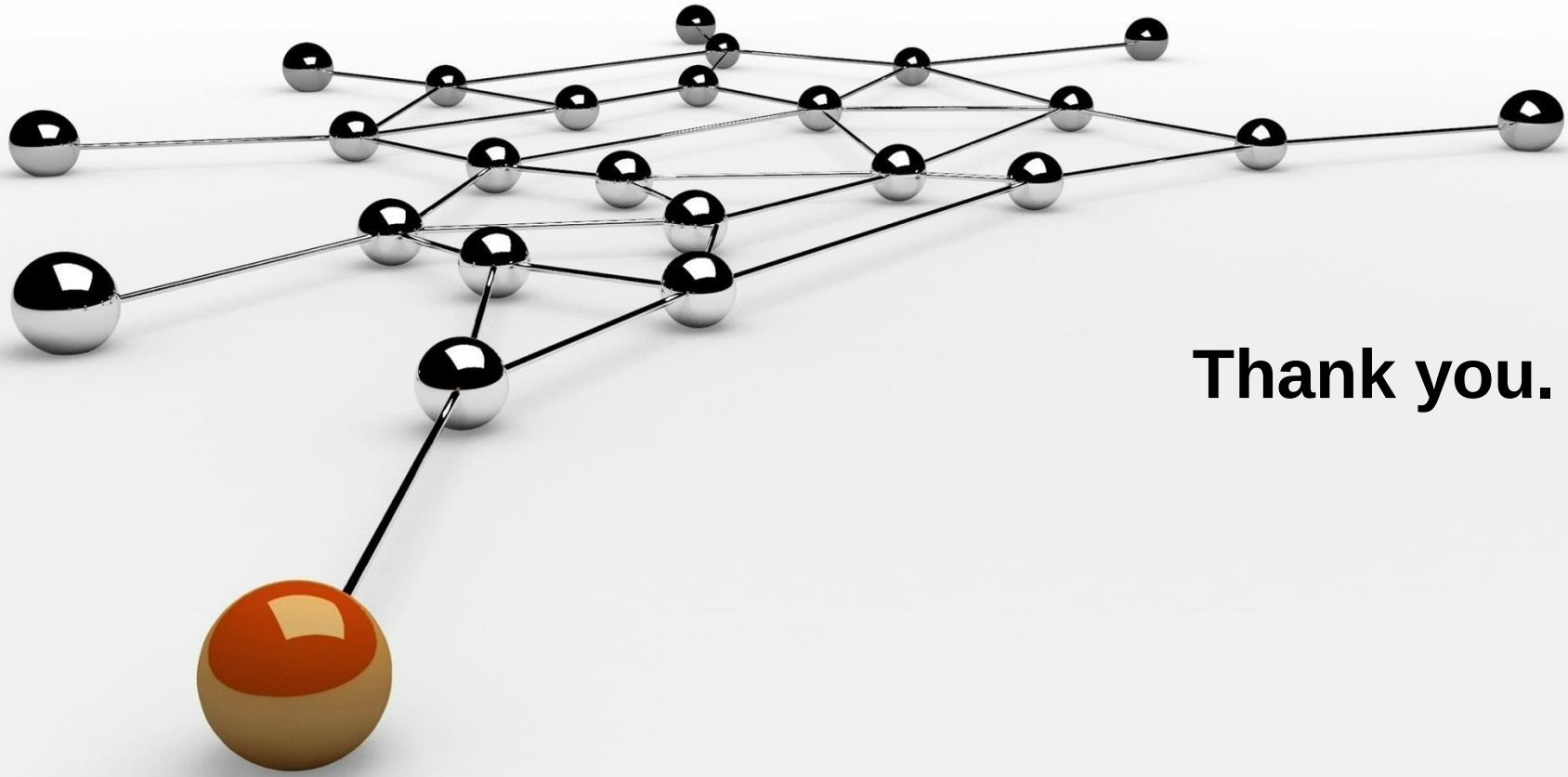
- Compatible with all Linux distributions
- Contains a bunch of files of the application, packed in a .tar archive and compressed using GNU Zip (.gz) or BZip2 (.bz2).
  - Format : <filename>.tar.gz or <filename>.tar.bz2
- Can be unzipped and unpacked on a directory using the tar command:
  - tar xvzf <filename>.tar.gz
  - tar xvjf <filename>.tar.bz2
- "INSTALL" or "README" files are also exist in this directory giving application-specific usage information

Installing from Source

- Software Packages coming in source code archives have to be compiled before installed
- Usually come in .tar.gz/.tar.bz2 archives
- Typical compilation/installation steps
  - Unpack the archive:
    - tar xzvf <package_name>.tar.gz
    - tar xvjf <package_name>.tar.bz2
  - Change to the extracted directory
    - cd <extracted_dir_name>
  - Run source configuration script as follows:
    - ./configure
  - Build the source code using the GNU Make utility as follows:
    - make
  - Install the package as follows:
    - make install

Aricent®

Engineering excellence. Sourced.

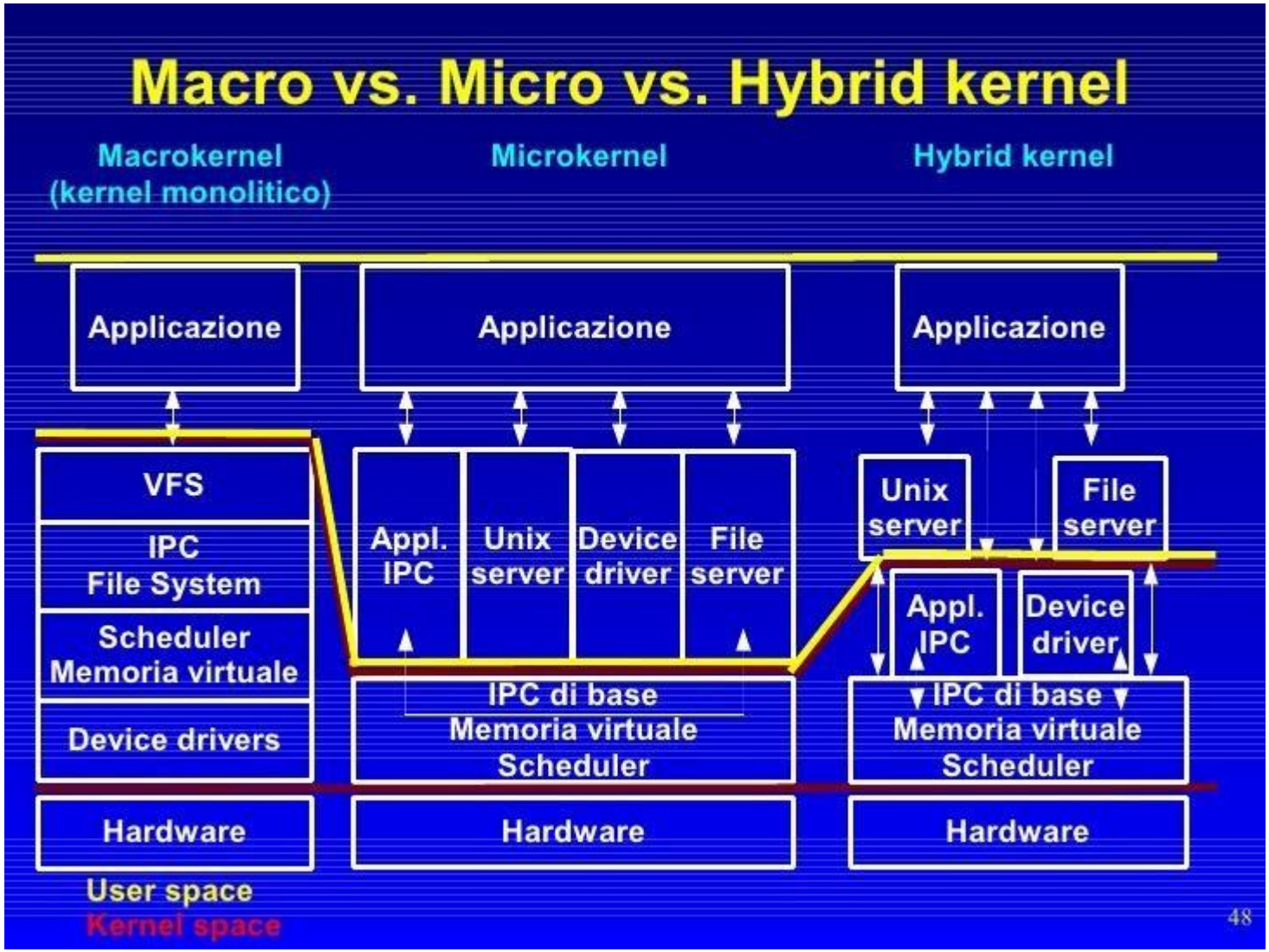


Thank you.

www.aricent.com

Additional slides

Different kernels



Different kernels

- Monolithic Kernels function with all of the kernel and device drivers inside one address space running in kernel mode.

Example: Linux, MS-DOS, Windows 9x, Mac OS versions below 8.6

- A Microkernel tries to run most of its services and device drivers in user space. This can result in an increase in stability and possibly security on machines with a Memory Management Unit.

Examples: AmigaOS, Amoeba, Mach, Minix, L4, QNX

- A hybrid kernel combines the concepts of both monolithic kernels and microkernels. It is generally implemented by having a monolithic kernel with a more microkernel like design. When properly implemented it is hoped that this will result in the performance benefits of a monolithic kernel, with the stability of a Micro kernelity on machines with a Memory Management Unit.

Examples: NT kernel (used in Windows NT, 2000, XP, Vista and Windows 7), Darwin (Mac OSX's kernel), DragonFly BSD, BeOS, Plan 9