Open source software and Linux introduction

LECTURER: TRẦN NGUYÊN NGỌC

EMAIL: NGOCTN@SOICT.HUST.EDU.VN; NGOC.TRANNGUYEN@HUST.EDU.VN

OFFICE: 405-B1 HUST

Content

1. Open source software
2. Linux introduction
3. Other open source software
4. How to exchange open source software

Open-source software

- Operations on software
- Concept of free software or open source software
  - Software license
- Relationship between proprietary and open source software
- Business models for open-source software
- History of open-source software
- Some popular software licenses of open source software
  - Apache License
  - BSD License
  - GNU License
- Development models for open-source software

Operations on software

1. Software creation
2. Using software: personal, company, business, education, research
3. Modification, upgrading, improvement
4. Reverse enginering
5. Software distribution: built/binary version, source code, patches, upgrading.
6. Software management: allow/prohibit to execute the operations on software

What is open-source software (OSS)?

- Source code need to be provided publicly
- Some rights (copyright) normally belonging to the owner is provided with software license
  - Research, modify, upgrade
  - Distribution
- Examples of open-source software
  - Apache, Asterisk, Linux, Android Virtual Box
  - Open Office, Firefox, Hadoop, Scilab, Chromium (Web browser)
- OSS is usually free. Using OSS saved about 60 billions USD/year (Cnet, 2018)

Some open-source software

Distinguish OSS with other software types

- proprietary software
  - Can only use software some limited conditions
  - Can not modify or reverse engineering
- Freeware
  - You don’t have to pay to use the software
  - But they are not OSS
- Shareware
  - Being provided free but limited functions
  - Can only use full functions if paying fees

OSS definition of Open Source Initiative

Introduction
Open source doesn’t just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

1. Free Redistribution (Tự do phân phối)
The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources.

The license shall not require a royalty or other fee for such sale

OSS definition of Open Source Initiative (2)

2. Source Code (Mã nguồn)

The program must include source code, and must allow distribution in source code as well as compiled form.

Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge.

The source code must be the preferred form in which a programmer would modify the program.

Deliberately obfuscated source code is not allowed. Intermediateforms such as the output of a preprocessor or translator are not allowed.

OSS definition of Open Source Initiative (3)

3. Derived Works (Cho phép thay đổi)
The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software

4. Integrity of The Author's Source Code
The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time.

The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software

OSS definition of Open Source Initiative (4)

5. No Discrimination Against Persons or Groups (không phân biệt người hay nhóm)
The license must not discriminate against any person or group of persons

6. No Discrimination Against Fields of Endeavor (không phân biệt mục đích sử dụng)
The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research

7. Distribution of License
The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

OSS definition of Open Source Initiative (5)

8. License Must Not Be Specific to a Product
The rights attached to the program must not depend on the program's being part of a particular software distribution.

If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution

9. License Must Not Restrict Other Software (không hạn chế PM khác)
The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software

OSS definition of Open Source Initiative (5)

10. License Must Be Technology-Neutral (Trung lập về công nghệ)
No provision of the license may be predicated on any individual technology or style of interface.

– Open Source Initiative, http://opensource.org/docs/osd

→ Can be free or not but must be free of modifying and redistribution

Business models for OSS

- Cannot earn money using traditional methods
- How could we earn money or get benefits from OSS?
- The implementation of OSS in companies is technically difficult
  - Lack of official support
  - Training how to use software
  - Fast development and upgrading but can be no long-term development
- Provide services:
  - Training/ Solution (Redhat)
  - Technical support (Redhat)
  - Sponsorship/ advertisement (Firefox)
  - Make plugin for OSS (IBM – Eclipse)
  - Commercialise partly (2 versions parallel) – MySQL
  - Commercialise fully (only provide commercial version by closing OSS)
- Other benefits
  - Fame/reputation for developers joing OSS projects

Redhat services

Consultant

Training and certification

Technical support for software products

Technical management for software products

Provide OSS solution

Provide commercial products

Canonical services

http://www.canonical.com/

Education

Development

Certification

Technical support …..

Compare open/non-open software

Will OS destroy software market?

- Can earn money from OSS projects
- OSS can be an intermediary step to develop proprietary software
- Share development cost
- There is no economical obstruction (Ex: patches)
- There is no hidden mechanism

OSS can be develop

- According to user need
- No limit on creation
- Need law support

Advantages of OSS

Expand market

Establish industrial standards

Attract developers

Upgrade the development of technology

Provide reliable, stable, low-cost software

Flexible, creative, and innovative

No pressure from commericalisation

Disadvantages

Difficult to convince users except developers

No data about software functionalities

Hard to commercialise software

50-50 with hackers

History of OSS

1950-1960: free distribution because the limitation of hardware (peculiarity, low number)

End of 1960s: price cost of software increased → software cost had been counted in installed hardware

1980s: Software industry started to provide binary version, prohibit to modify software

1980: Copyright law extended to computer software

History of OSS (2)

- Unix (1971):
  - Multi-users, multi-tasking
  - Reliable and stable
  - Flexible, widely usage
  - Influence on developers, including Bill Gates, Steve Jobs, etc.
  - $$$$$$$$$$$$
- MINIX and Tannenbaum
  - Andrew Tannenbaum developed MINIX (Mimic Unix) for teaching and research but not allowing to redistribute modified versions (can redistribute the original one freely)
- 1983: Richard Stallman, MIT started GNU Project with the ultimate goal: build an GNU OS (GNU's NOT Unix) like Unix but free
- 1984: GNU project list software products needed to develop a complete GNU OS → develop those software

History of OSS (2)

1985: Free Software Foundation (FSF) was established to support GNU project, definition of Free Software: free to modify, redistribute or sell (Copyleft)

The beginning of 1990s, most of GNU OS parts are ready except OS kernel (because software had been develop on Unix environment)

1991: Linus Tovalds introduced Linux kernel (Like-UNIX) free and modifiable

1992: Linux kernel applied GNU General Public license (GPL) and was integrated to GNU project and became Linux OS

1998: Open Source Initiative (OSI) was found

Software copyright

- Software copyright is the application of copyright law to machine-readable sotware. It regulates the rights of developers/inventors/companies making the software, including
  - Copy right
  - Redistribution right
  - Modification right
- OSS has copyright or not?
- OSS has copyright for some reasons
  - Showing the contribution of authors/developers
  - Enforcing their terms

Software license

- A software license is a legal instrument (usually by way of contract law, with or without printed material) governing ***the use or redistribution*** of software
- With proprietary software, it is the regulation that end-user can use one (or some) copies of the software without violating software copyright
- EULA (End-user license agreement)
- OSS is not limited users in term of using and redistribution

Software license of proprietary software

- Likely protect the rights of software inventors: COPYRIGHT
- Certain rights regarding the software are reserved by the software publisher
  - Software redistribution
  - Software management
- User can only use one copy of the software, need to pay more if using multiple copies for different computers
- Can not modify and upgrade software
- Can not reverse engineering
- Example
  - MS Excel EULA
  - MathWork Mathlab

MS Excel EULA

- Can only use on a single computer
- Can not share through network connection
- Fair/Unfair?
- Need to use MS Excel on two computers → need to pay more

Software license of OSS

- Provide maximum rights to end users – Free
- Can modify, upgrade, and redistribute software, including source code: open source
- For most OSS license, OSS can be used in business purposes without paying fees for authors
- Some OSS licenses only allow redistribution for non-business purposes
- Limit software management to make sure software maintaining the free and open source (COPYLEFT)

Copyleft?

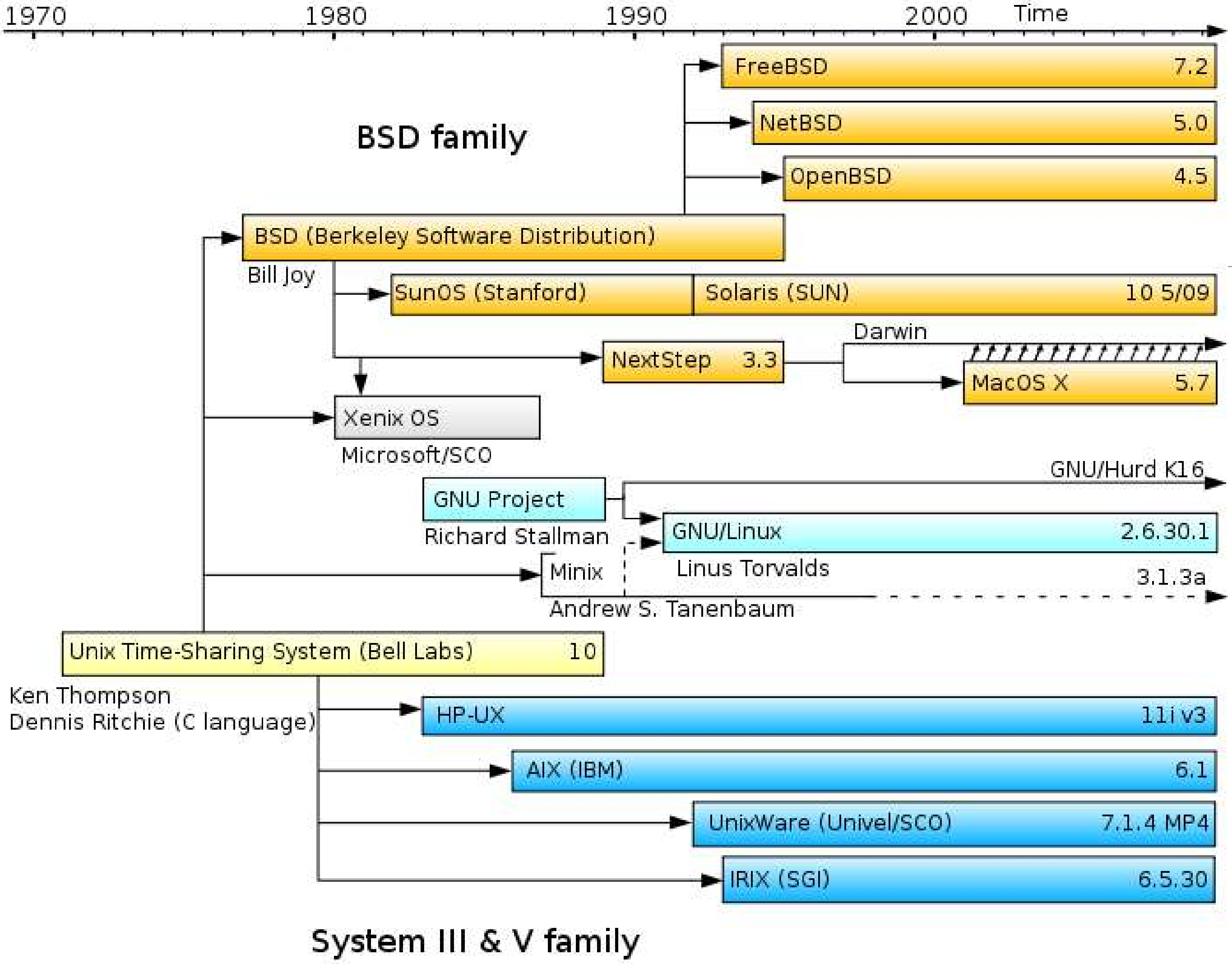
- Why do we need copyleft?

❖ Because without copyleft, OSS can be exploited to change to proprietary software → no more free

❖ Copyleft
  - Gives users/viewers the same rights as distributors (copying, distribution and modification)
  - Makes sure the freedom of software
  - Use copyright law to protect the freedom of the software

OSS licenses

- While authors want to provide source code for an OSS project, they likely provide it with a license
- There are more than 1400 OSS licenses
- Some licenses can only be used for a software product, while others can be used for multiple ones.
- Since 2008, OSS must follow copyright law as well (US law) → end-users must comply regulations in OSS licenses
- Some popular OSS licenses
  - Apache License
  - GNU General Public License, GNU Lesser General Public License - Copyleft
  - BSD License – modified versions can be redistributed with more limited restriction than the original version


1970
1980
1990
2000
Time
BSD family
FreeBSD 7.2
NetBSD 5.0
OpenBSD 4.5
BSD (Berkeley Software Distribution)
Bill Joy
SunOS (Stanford)
Solaris (SUN) 10 5/09
Darwin
NextStep 3.3
MacOS X 5.7
Xenix OS
Microsoft/SCO
GNU/Hurd K16
GNU Project
Richard Stallman
GNU/Linux 2.6.30.1
Linus Torvalds
Minix
3.1.3a
Andrew S. Tanenbaum
Unix Time-Sharing System (Bell Labs) 10
Ken Thompson
Dennis Ritchie (C language)
HP-UX 11i v3
AIX (IBM) 6.1
UnixWare (Univel/SCO) 7.1.4 MP4
IRIX (SGI) 6.5.30
System III & V family

Apache license

- The Apache License is a free software license authored by the Apache Software Foundation (ASF).
- Can use source code to develop free or non-free software
- Allow to distribute modified versions without the original license (không phải là copyleft)

BSD License – 4-clause license

Copyright (c) <year>, <copyright holder>
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution

BSD License – 4-clause license (2)

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the <organization>.
4. Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY <COPYRIGHT HOLDER> ”AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

BSD License (new) – 3-clause license

- Allow to distribute software without limiting purposes
- Copyright warning and the same content of denying responsibility of any damage while using (capitalised texts)
- Same as 4-clause license but removing clause 3 about advertisement
- OSI approved this license with the name The BSD License
- FSF approved this license compatible with GNU GPL. It was named Modified BSD License

FreeBSD License – 2-clause license

- Remove clause-3 and clause-4 in comparison with the original license
- Compatible with GNU GPL

So, BSD License only protect the copyright on the distribution version but the modified versions are not forced to be OSS

GNU General Public License

- The most popular OSS license : Linux, GCC, MySQL, etc.
- Approved by FSF
- Copyleft
- Can be used for many software products
- Three versions

GNU GPL v1 (1989)

Prevent the redistribution with limiting user freedom

Require the source code and binary versions go together

Modified or integrated software from the original one must be distributed by GNU GPL as well

No software price mentioned

GNU GPL v2 (1990)

Added clause: Liberty or Death

If the receivers (users) get the software with GPL license cannot redistribute software with full free rights, they cannot redistribute the software

GNU – Lesser GPL

- GNU LGPL is used for libraries
- Stands in between copyleft and BSD
- Software was “originated” from LGPL musted be allowed to modify source code, reverse engineering, and debug.
- Independent software using LGPL libraries don’t count as “originated” from those libraries

  → Can use GNU LGPL to write non-free software

GNU GPL v3 (2007)

- It states that GPL software can be sold with any price → no limitation of commercialisation
- Distributors cannot restrict rights provided by GPL

Development models for OSS

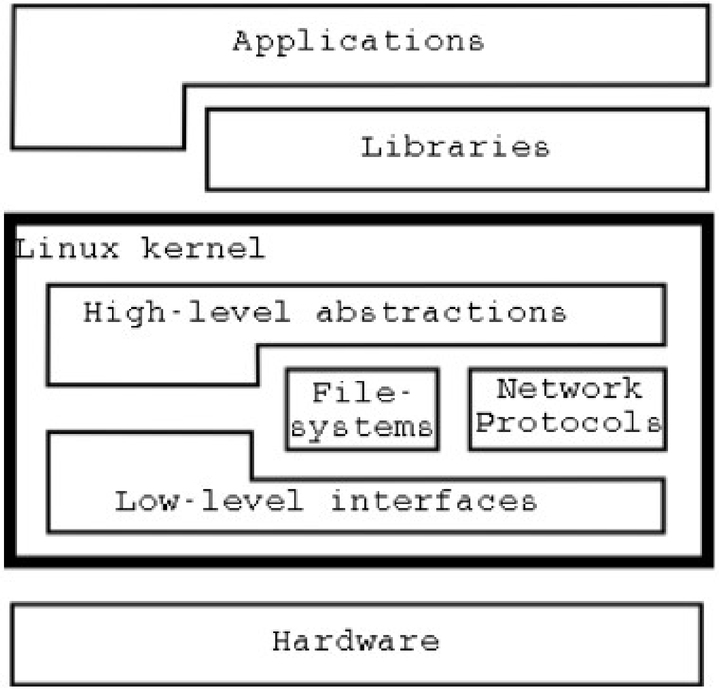
- Cathedral
  - Traditional model
  - Code developed between releases is restricted to an exclusive group of software developers
  - Freedom of members is limited
  - Examle: GCC, Linux kernel
- Bazaar
  - Over the Internet in view of the public
  - Users can be developers as well
  - High freedom
  - Early released version
  - Modules
  - Integrate new modules regularly
  - Three versions: stable, beta, night version
  - The inventor of this process: Linus Torvalds

Linux introduction

- Originated from GNU project to generate an UNIX-similar OS but free and open-soure
- GNU components developed on Unix can work independently
- Linus Tovalds created Linux kernel, and then integrated other components to make the first version of GNU Linux

1.1. What is Linux?

- Kernel / system / distribution
- Broad definition / many interpretations
- GNU/Linux anyone?
- Strict definition: Linux is the kernel developed and maintained by Linus Torvalds.
- Linux kernel:
  - Provides core system facilities
  - Manages system through its lifecycle (next reboot)
  - Controls all hardware
  - Provides higherlevel abstractions to software



1.1. What is Linux? (cont.)

- Old kernel version identification scheme:
- Three digits: x.y.z
  - x.y => version number
  - z => release number
  - Linux 2.4.21 is release 21 of version 2.4
  - Linux 2.5.73 is release 73 of version 2.5
  - Even version number (ex.: 2.4) => stable tree
  - Odd version number (ex.: 2.5) => development tree
- Reallife kernel version identification schemes:
  - 2.4.18rmk4hh24 (Handhelds.org) / 2.4.208 (Rhat)

- Example of old scheme
  - 1994: 1.0
  - 1999: 2.2.0
  - 2001: 2.4
  - 2003: 2.6.0
  - 2009: 2.6.3

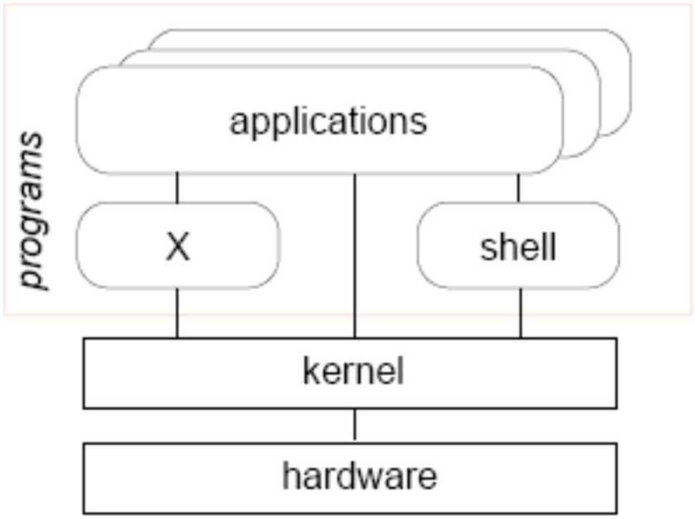
1.1. What is Linux? (cont.)

- Current version identification scheme
  - Four digits: x.y.z.a
  - x.y => version number
  - z => release number
  - a => stablization increment
  - These four digit releases are not maintained by Linus, but rather by other kernel developers
  - Goal: Provide stable/fixesonly releases.
  - Starting from 2.6.11.1

- Release cycle:
  - After major stable release, 2 week window for major features.
  - 2.6.x released, merge window opened for 2.6.(x+1)
  - Merge window close = 2.6.(x+1)rc1
  - 6 to 10 week bug fixing brings new 2.6.(x+1)rcN
  - When stable, 2.6.(x+1) release
  - Lather, rinse, repeat
- No 2.7, 2.9 or 2.(6+n) planned for now
- Maybe when PREEMPT_RT goes fully in = 3.0?

Linux components (including kernel)

- Kernel
- Drivers
- System software
- Application software
- X Windows
- GUI software



Linux functionalities

Open source

- Unlimited innovation?

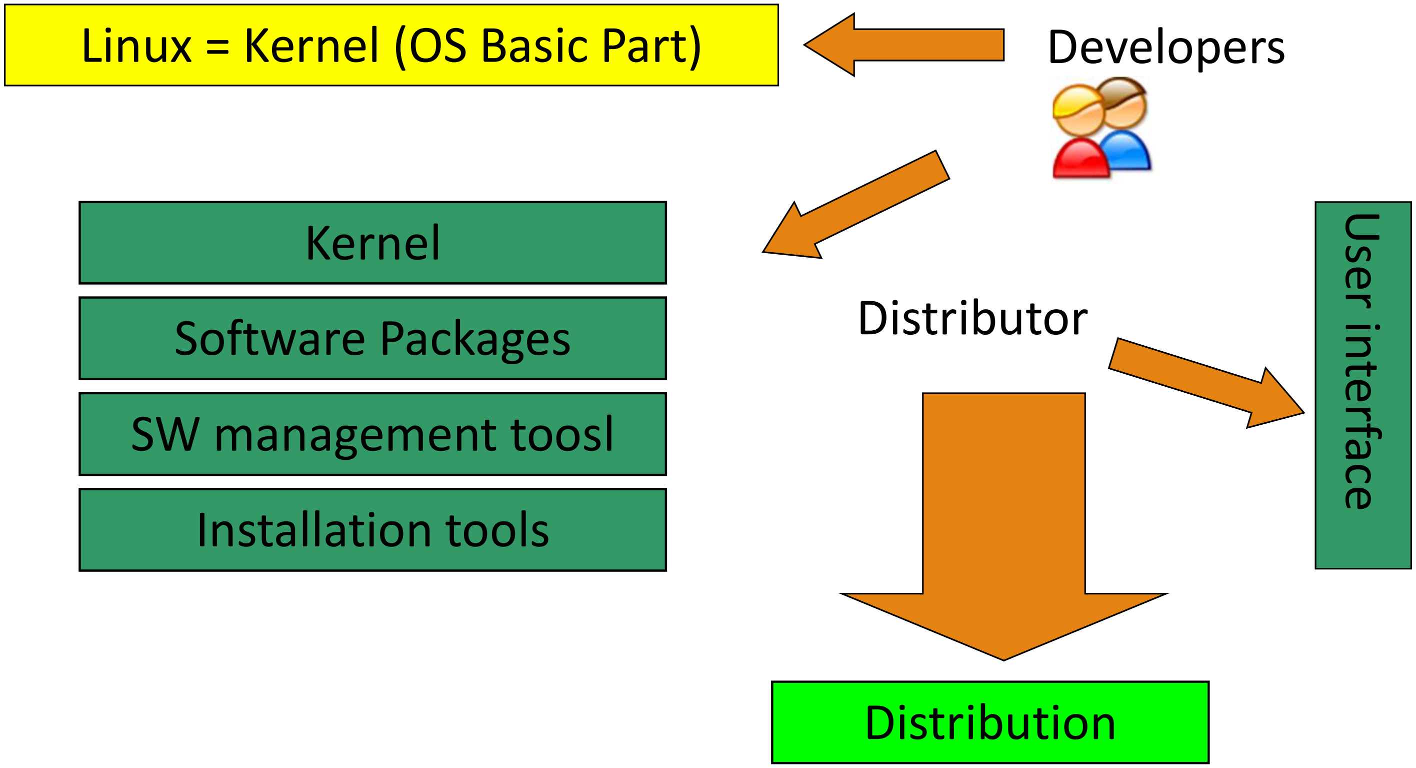
Support many hardware types

Many distribution

Inherit Unix characteristics

- Portability
- Multiusers, multi-tasking
- A single system file
- Shell
- Networking

V. Linux distribution versions



Linux = Kernel (OS Basic Part)

Developers

Kernel

Software Packages

SW management toosl

Installation tools

Distributor

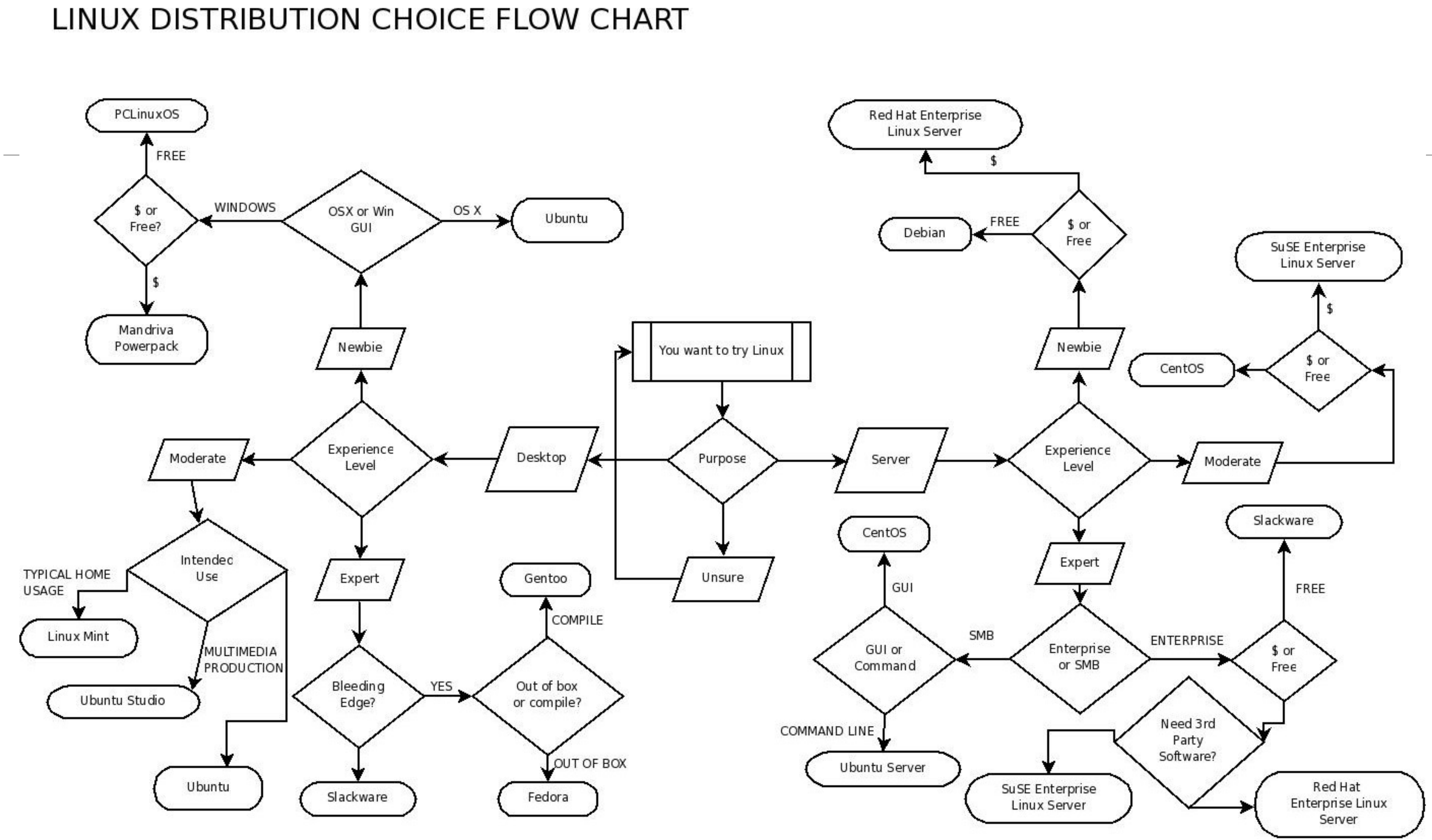
User interface

Distribution

Some popular distributions of Linux

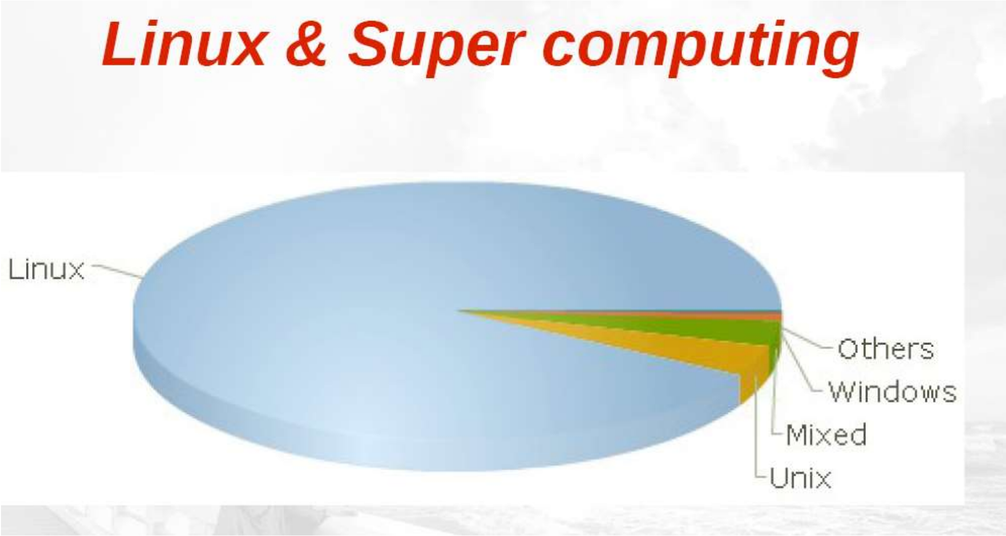
- Debian
  - Knoppix
  - Ubuntu
- Fedora (sponsord by Red Hat)
  - Fedora
  - CentOS
  - Red Hat Enterprise Linux

Pick the right distribution for your needs



Linux applications

- LAMP
  - Linux, Apache, MySQL, Perl/PHP/Python
- Gaming (PS3)
- Embedded systems
  - Set top boxes, mobile phones, routers and switches, dreambox, TiVo
- Handhelds
  - Android
- Supercomputers
  - 34 fastest running Lnux, 92% of top500



3. Some other OSS

Linux

- Webserver
- Mail server
- KDE, GNOME, …..

Other OSs

- Open Office
- Gimp
- FireFox
- LaTeX

How to exchange/support OSS development

- CVS (Concurrent Versions System) to manage versions
- SVN (Apache Subversion) to manage versions
- Use repositories containing source code on websites, allowing OSS project to host and manage the progress of OSS development. Mostly support CVS and SVN
  - Example: Sourceforge, github, gitlab, etc.

Embedded system

The combination of

- Hardware
- Software
- Mechanical structure
- Other components

  To execute (a) specialized function(s) (*Michael Barr, Programming Programming Embedded System in C and C++)*

An application contains at least a programmable computer (microcontroller, microprocessor,…)that normal users cannot aware a computer-based system. (*Michael J. Pont, Embedded C)*

Examples of embedded systems

- In Cars:

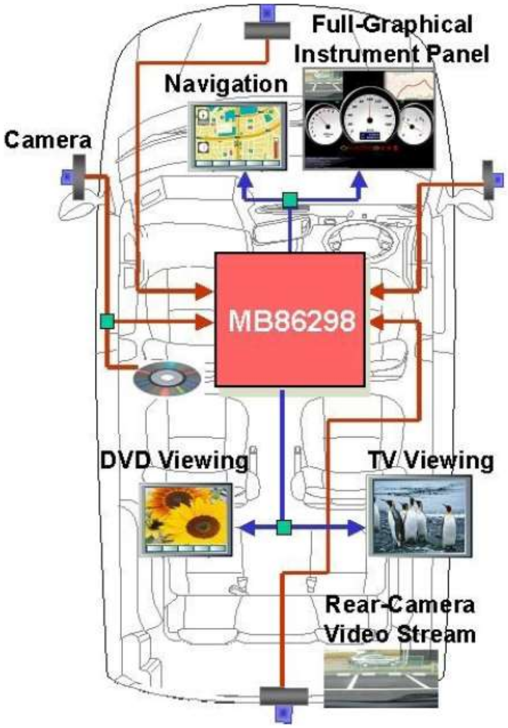
From Computer Desktop Encyclopedia
© 2008 The Computer Language Co. Inc.

High-Speed Control

- Antilock Braking
- Central Electronics
- Electronic Throttle
- Engine Control
- Steering Wheel
- Transmission Control

Low-Speed Control

- Audio
- Climate Control
- Driver's Door
- Driver Information
- Passenger Door
- Phone
- Power Seat
- Rear Electronics
- Sun Roof
- Supplemental Restraint System
- Upper Electronics



Examples of embedded systems (2)

Automatic seller machines

- Chipset: 16-bit Hitachi H8/300H processor
- Mechanical components: motor,

Toothbrush

- 8-bit processor

Digital cameras
- DIGIC II Image Processor

Questions

Which OS do we need for embedded system?

Why Linux for embedded systems?