August 30 2012

# Introduction to Linux

## Nuclear Engineering Voluntary computing seminar

**Ondřej Chvála** <ochvala@utk.edu>

## Lecture overview

- Tentative program for the seminar
- Very brief history of computing
- UNIX development
- GNU/Linux: free and open source

# Seminar plan for this semester

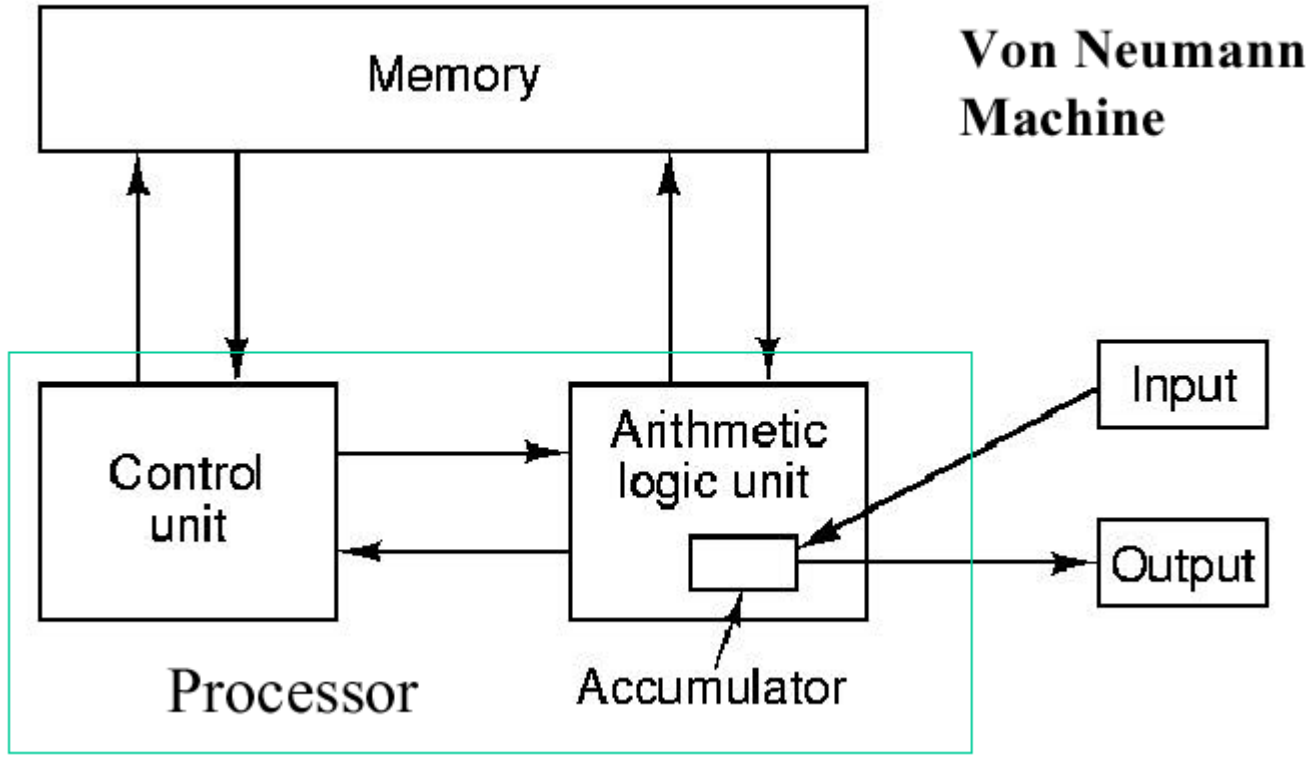
- Start with some historical and theoretical background.
  - Where we are and how we got here: todays lecture.
  - Introduction to binary representation, pitfalls of floating point.
- Get more technical and practical as time progresses.
  - Efficient use of ssh – practical guide.
  - Basics of bash shell scripting.
  - C/C++ programming and C++ frameworks.
  - Using common tools: gnuplot, sed, awk, perl, mc, etc.
  - High performance computing architectures and programming tools.
- Focus on student's individual exploration on their own.
- Questions and suggestions welcome.


# What is Linux?

- **GNU/Linux** is a **Unix-like operating system** (OS) assembled under the model of **free and open source software** development and distribution. The defining component of Linux is the Linux kernel, an operating system kernel first released on October 5 1991 by Linus Torvalds.
- Runs on many hardware platforms: Alpha, ARM, AVR32, Blackfin, C6x, ETRAX CRIS, FR-V, H8/300, Hexagon, Itanium, M32R, m68k, Microblaze, MIPS, MN103, OpenRISC, PA-RISC, PowerPC, s390, S+core, SuperH, SPARC, TILE64, Unicore32, x86, Xtensa, IBM z/Architecture
- Leading OS on servers, mainframes, and supercomputers: >90% of today's top 500 supercomputers run Linux, including the 10 fastest.
- Runs on embedded systems such as phones, tablets, network routers, TVs, game consoles. The Android system is built on the Linux kernel.
- Recommended viewing: 2001 documentary "Revolution OS". (Also 2001 Finnish documentary about Linux "The Code".)

# Cherry-picking trough computing history

- **Charles Babbage** (1791 – 1871): first Turing-complete computer (1837) Mechanical, never built. https://en.wikipedia.org/wiki/Analytical_Engine
- **Alan Turing** (1912 – 1954): Universal Turing Machines (1936) theoretical computing model. https://en.wikipedia.org/wiki/Turing_machine
- **John von Neumann** (1903 – 1957): von Neumann architecture (1946) Stored program computer: both data and program are in the memory.



# Cherry-picking trough computing history (2)

- **ENIAC** (1946): first Turing-complete electronic computer.
- **FORTRAN** (1957): Practical alternative to assembly language.
- **MULTICS** (1964) (**Mult**iplexed **I**nformation and **C**omputing **S**ervice) : Early time-sharing OS by MIT, GE, Bell Labs to create commercial "computing utility" on mainframes. Many innovative and novel ideas.
- **UNIX** (1969, originally Unics) Ken Thompson & Dennis Ritchie (also C language) while working at AT&T's Bell labs, as a side project on little used PDP-7 machine. Adopted, streamlined, and enhanced MULTICS ideas: multi-tasking, multi-user, security, CPU time sharing, hierarchical filesystem, dynamic libraries, virtual memory, portability, command line interpreter (shell), pipes to pass data between processes, etc.
- 1970 – Unix rewritten for PDP-11/20 with AT&T funding for adding text processing functions. Then used internally for patent applications.

# DEC's PDP-7 & PDP-11

# Ken Thompson & Dennis Ritchie working on PDP-11

# UNIX philosophy

- Douglas McIlroy: "**Write programs that do one thing and do it well.** Write programs to work together. Write programs to handle text streams, which is a universal interface." **Everything in UNIX is a file, preferably a text file.**
- Eric Raymond: The "KISS" principle, from The Art of Unix Programming
  - Modularity – simple parts connected by clean interfaces
  - Clarity – better than cleverness
  - Composition & Separation & Least surprise
  - Simplicity – add complexity only when you must
  - Transparency – make debugging easier
  - Robustness – stems from transparency and simplicity
  - Silence – when a program has nothing to say, it should keep silent
  - Repair – when program must fail, fail loudly and as soon as possible

"Those who don't understand Unix are condemned to reinvent it, poorly." – Henry Spencer

# Short and abbreviated history of UNIX

- Following 1956 consent decree in anti-trust case, AT&T was required to freely license any non-telephone technology to anyone..
- 1972 – Unix rewritten in C => easy portability and wide adoption.
- 1975 – UNIX identified as well suited for ARPA network host, RCF#681
- 1983 – TCP/IP networking released with Berkeley (BSD) Unix 4.2.
- 1983 – US DoJ settled AT&T case by breaking up Bell systems. AT&T promptly commercialized Unix System V (which nearly killed Unix).
- 1984 – X Windows System, network transparent GUI system developed by MIT and released under free and open MIT license.
  - 1987 – current protocol version released (X11)
  - Note: there are X-servers for Windows, such as Xming and Cygwin/X Get one, if you have a Windows machine.

# GNU's Not UNIX

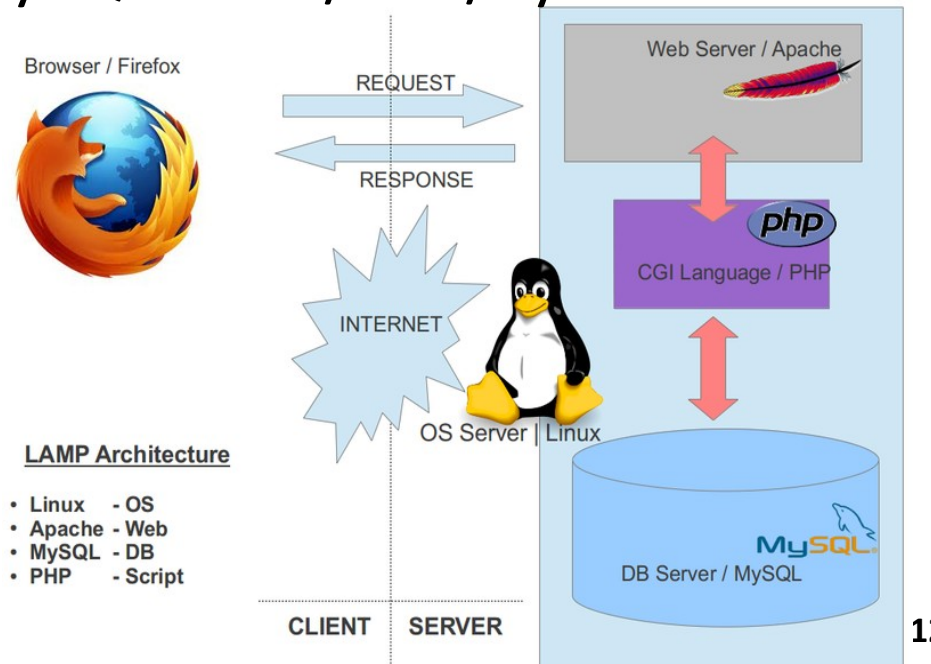
- 1983 – Richard Stallman announced plan for GNU operating system
  - GNU = GNU's Not Unix: new free (as in Freedom) Unix-like OS
  - 1985 – GNU Manifesto, 1986 – Free Software Foundation (FSF)
  - 1989 – GNU General Public License, first "copyleft" license ©: using copyright to require derivative works to preserve distribution right.
- Free Software Definition - matter of liberty not price. See Youtube talk.
  - **Freedom 0**: **to run** the program for any purpose.
  - **Freedom 1**: **to study** how the program works, and change it to make it do what you wish.
  - **Freedom 2**: **to redistribute** copies so you can help your neighbor.
  - **Freedom 3**: **to improve** the program, and release your improvements to the public, so that the whole community benefits.
  - Note: Freedoms 1 and 3 require source code to be available.

# GNU's Not UNIX

- 1984 to 1991: GNU system was developed by FSF by rewriting Unix part by part (remember modularity) and releasing it under GNU GPL. Also collecting existing parts which were licensed by other free licenses such as MIT or BSD licenses, and releasing GNU GPL versions of it.
  - Notable parts: Gnu C library (glibc), Gnu Compiler Collection (gcc), Gnu debugger (gdb), bash shell, tc shell, userland tools (ps, top, ls, mkdir, ...), vim and emacs editors, email clients, etc.
- It lacked a kernel: layer which talks to HW (GNU Hurd in development).
- Fortunately, Linus Torvalds released Linux kernel v. 0.99 under GNU GPL in 1992. Voilà, a free OS was born. See history of Linux kernel here.
- 1993: >100 developers work on Linux kernel. First GNU/Linux distributions: Slackware and Debian.
- 1994: X11 GUI ported. RedHat and SuSE distros. Linux kernel ver.1.0

# GNU/Linux rise to dominance

- End of 1990: Tim Berners-Lee at CERN: first web sever and browser.
- 1995: Apache HTTP server – free high performance & highly configurable reliable web server, killer app of Linux in 1990s.
- 1995: MySQL database – free full-featured relational database management system.
- 1995: PHP – simple server-side scripting language.
- The "LAMP" stack: Linux + Apache + MySQL + Perl/PHP/Python enabled free dynamic web development.
  - Free and robust alternative
  - Historical notes:
    - Perl: 1987
    - Python: 1991



# Internet circa 1995

```
 -o)
 /\\
 \_v-
```

Linux is like a tee-pee:
No gates, no windows, and an Apache inside.

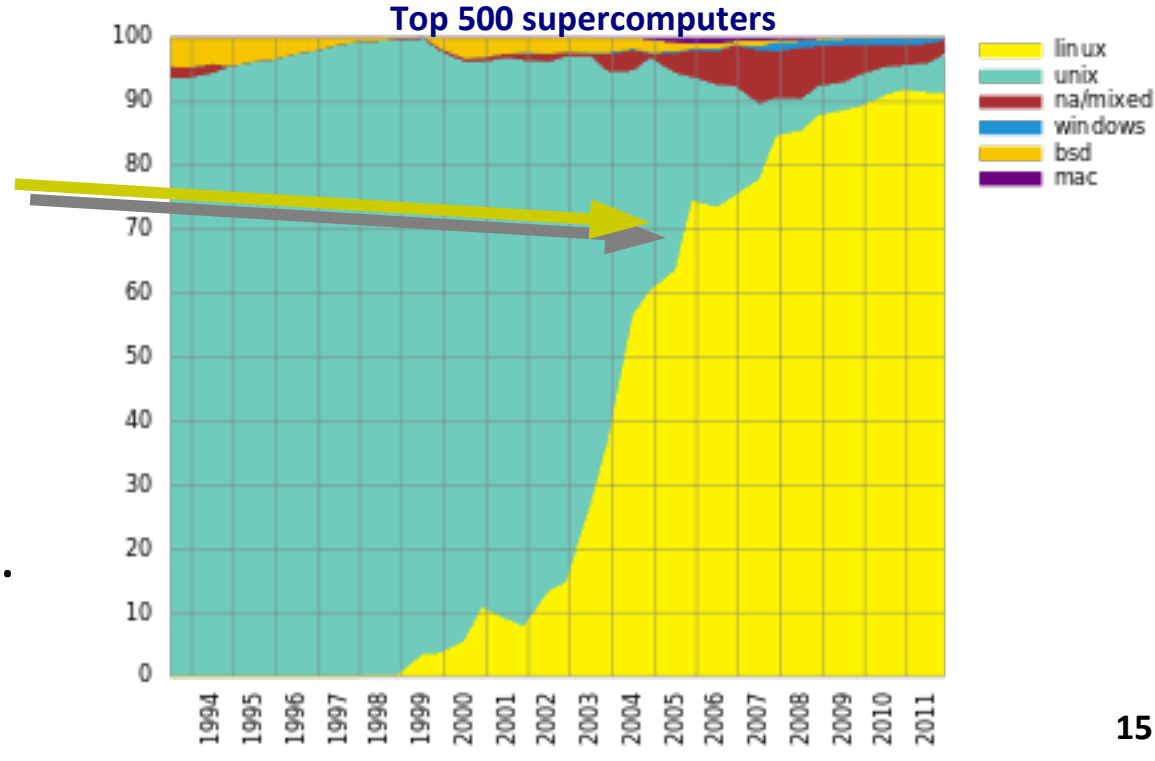
# GNU/Linux rise to dominance (2)

- 1997: Eric S. Raymond's "The Cathedral and the Bazaar" described why is open source superior to closed/proprietary source development.
  - Given enough eyeballs, all bugs are shallow.
  - Release early. Release often. And listen to your customers.
- 1998: Netscape Inc. persuaded by ESR arguments open-sourced the Netscape Communicator → Mozilla → Firefox browsers
- 1998: Open Source Definition by Bruce Perens – defined 9 rights. Business friendly & palatable version of Free Software Definition.
- 1998: Perens, Raymond et al. funded Open Source Initiative, IBM joined
  - IBM invested billions in Linux kernel development, funded >300 kernel developers, licensed several SW packages as open source.
- 1998: Oracle and Sybase do Linux ports, 1999: Linux @IBM mainframes

# GNU/Linux rise to dominance (3)

- Industry backing since ~1998 → support for more HW → wider popularity → more developers → more features and less bugs.
- Google, Amazon, Facebook, Wikipedia, web hosting, e-commerce, HW from Linksys routers to TV manufacturers adopted Linux and contributed code.
- Common code for all platforms (unlike Mac or Windows) → synergies in R&D such as multicore – originally developed for supercomputers, now even phones (Android) are multicore – just works.

- Adoption curve of Linux (yellow) among **Top 500 supercomputers** between 1994 and 2012, as the % share of the market.
- **Android** 59% of smartphone market share, 1 000 000 device activations daily (as of mid 2012).



# GNU/Linux distributions

- Linux kernel + GNU software + other software = distribution
- Allows easy installation and updating of the operating system.
- Variety = freedom to choose from hundreds. Each distribution has a niche.
- Major categories by package management system (ignoring Slackware):
  - **Redhat**-derived: RPM packages
    - RHEL, Fedora, CentOS, Scientific L., Oracle L., SUSE, Mandriva, ...
  - **Debian**-derived: APT system, deb packages
    - Debian, Ubuntu, Mint, Knoppix, SymphonyOS, Element, AuroraOS, …
  - **Source** based: portage, compiled with box-specific optimizations
    - Gentoo, Funtoo, Sabayon, VLOS, Toorox, Papug, ...
- Timeline graphs: http://futurist.se/gldt/

# Why is Linux more secure?

a) UNIX was **designed for** simultaneous work of many users, mandating high levels of **security**, from the **beginning**.

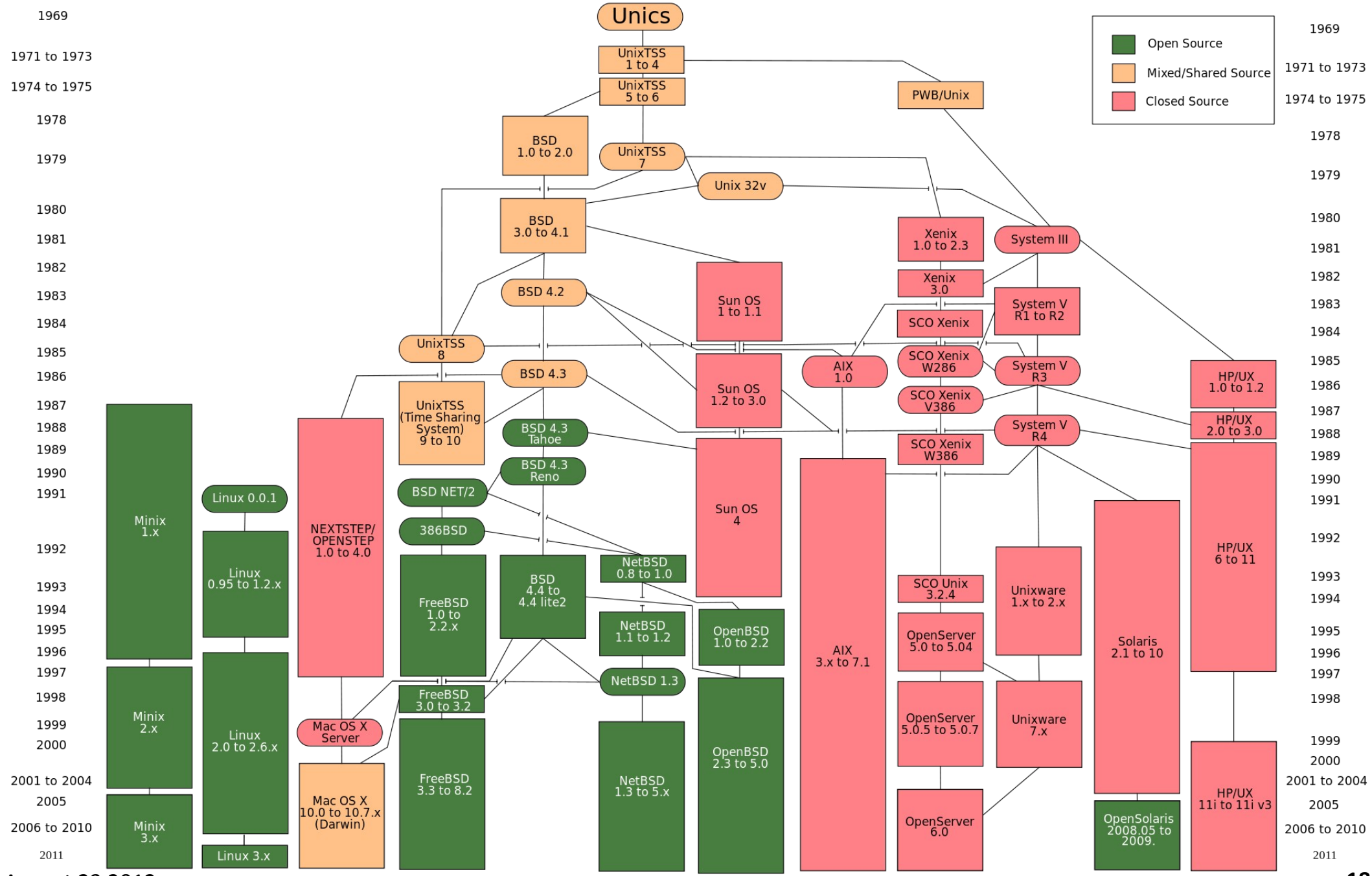
b) Linux is **open source** and widely **popular** in academia, industry, and govt. agencies → more bugs found and fixed more rapidly. Also new security features are introduced rapidly: Never-eXecute bit support on data structures, running services in insulated environment at native speed, Security Enhanced Linux developed by NSA, etc.

- Security by obscurity does not work.

c) Linux distributions have easy tools to keep all installed programs **up to date**

d) Linux is **highly heterogeneous** – almost every distribution has slightly different version of packages, with its own patches, and compiled with different optimization flags & included features. Therefore the binary images of executables are different, complicating exploitation of potential bugs. Every MacOS or Windows binaries are the same for the specific OS version.

# Other Unices - some of the many dozens



# Summary

- Design of computer architectures was studied for almost a century.
- First UNIX was developed off MULTICS in 1969, and spread due to portability (C language) and shared-source licensing.
- UNIX improved over 20 years of academic & industrial development.
- GNU project started to keep freedom in computation in 1984. OpenSource: more practical & less political PR strategy in 1998.
- Linux kernel was the missing piece in GNU system, added in 1992.
- Once got enough momentum (LAMP stack), Linux became used widely and supported by large players in the industry, which made Linux even more popular, developed, and applied to larger variety of hardware. Virtuous cycle continued, improving and expanding Linux.
- Other Unices are still in use, both free/open and proprietary.
- **There are many Linux distributions, try some!**