LINUX - What, why, Where & how

- What?
  - An open source free Unix-like Operating System
- Why? … in a word Freedom – Unix is not free.
  - to modify & adapt, even to commercialise derivatives
  - from licence restrictions
  - From costs
- Where? … everywhere
  - Embedded micro to mainframe to HPC to net...
- How?
  - It's made and works … following course: CS2506
  - To make it work … this course : CS2503!

Why study this -

Forerunner of all modern Operating Systems

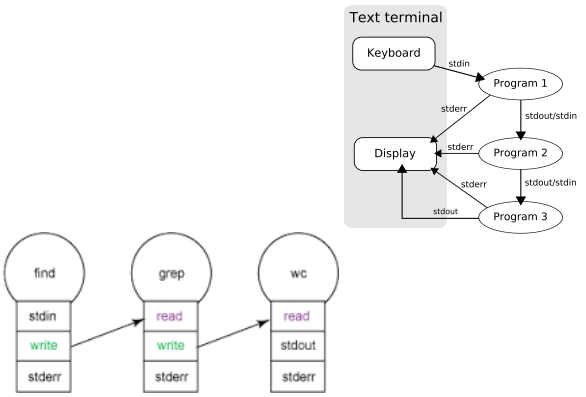
It underpins everything
so it overrules…everything

so if you want to get anything to work,
work on what runs everything.

Need to get files, processes, processors to play ball
Get the glue that molds, holds, and folds them all

And you can modify some of the ground rules!

Inter process comms via pipes



How did it all begin? - Unix

In early '60s, computers were expensive, had minimal memory and were single user, usually the operator, who manually selected and scheduled all jobs. (now an OS)

All activities were designed to make it easy for the computer

In the mid '60s, MIT, Bell Labs, and General Electric teamed to build, for a large computer, the GE-64, an OS:-

Multics (**Mult**iplexed **I**nformation and **C**omputing **S**ervice)

that would…

accommodate 1,000 simultaneous users

In March '69, Bell pulled out of the project

"Three people could overload it."

Unix Philosophy (affects culture, everything) – ultimate KISS

- Write programs that :-
  - do just one thing and do it well –
    - focussed, efficient,
      - runs fast on big dat, using low resources
  - work together, (jamming, mashing – way to go!)
    - So the output of one is the input of another
  - handle text streams, because that is a universal interface
    - So there are lots of really efficient text processing tools
- Don't hesitate to build new programs to do a job
  - But most jobs can be done by combining existing tools

Put text in your pipe and smoke 'em all!

(Disclaimer : I mean use Linux for ~~speed~~! - oops - efficiency!)

How did it all begin? Family tree - 1

- Bell-labs … AT& T : the US equivalent of Telecom Eireann, before deregulation…!
- After Multics … Unix
- Multics
  - programs tried to do too many things
- Unix - counter-revolution…
  - Just do one thing well
- Needed standards – else variations on variations…
  - see links on next slides

How did it all begin? Family tree – 2

- After Multics … Unix
- System V - mostly industry
- BSD - mostly universities
- Needed standards - POSIX in 1988
  - Portable Operating System interface
  - Partially adopted/implemented in Windows
- Linux - begun 91 - released v1.0 94
  - Begun as a student hobby project
  - Adopted due to commercialisation of others
- History of (some) distros
  - https://en.wikipedia.org/wiki/Unix
  - https://en.wikipedia.org/wiki/Linux
  - https://medium.freecodecamp.com/linux-is-25-yay-lets-celebrate-with-25-rad-facts-about-linux-c8d8ac30076d#.1elxgx77h

Comments on Wikipedia

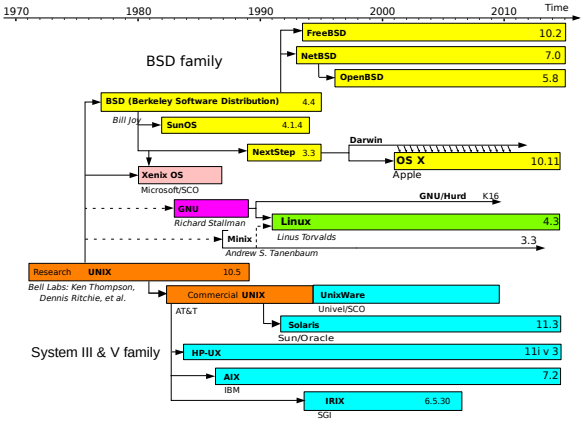
- Handy resource, but community sourced!
- May not be reliable, but convenient...for now
- Generally written by people with an interest, whether
  - Vested
  - Or just a passionate personal one
  - So over time, it is liable to be roughly right, as people with an interest will ensure that, until they lose interest
    - Which may be now!?
- Benefits
  - Great for a first port of call for
    - Basic outlines; explanations or diagrams
      - The best illustrations and explanations tend to remain
    - Comparisons of utilities

1991 a college student in Finland shared his project on an email list:

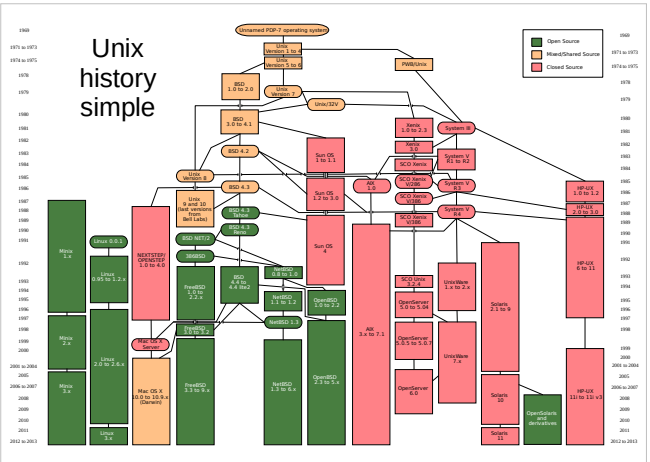
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Date: 25 Aug 91 20:57:08 GMT

Hello everybody out there using minix —
I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things). I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)
Linus (torv…@kruuna.helsinki.fi)
PS. Yes—it's free of any minix code, and it has a multi-threaded fs.
It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-(.

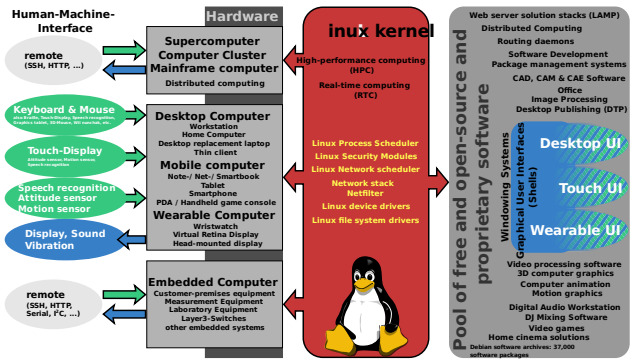
Unix timeline



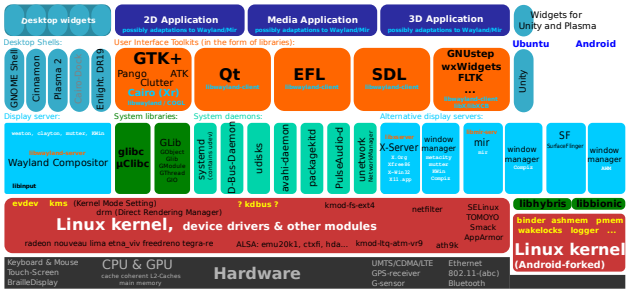
Unix history simple



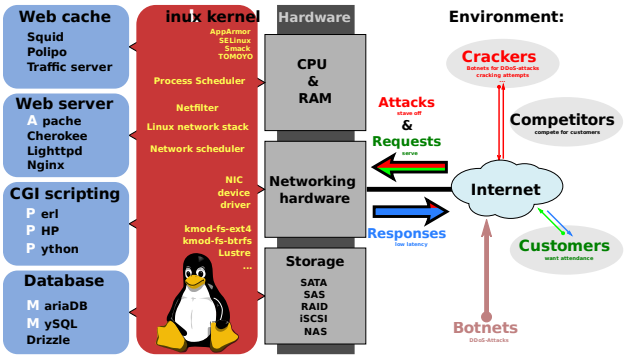
Linux kernel ubiquity



Free and open-source-software display servers and UI toolkits



LinuxApacheMySQLProg software bundle



Underpin = overrule

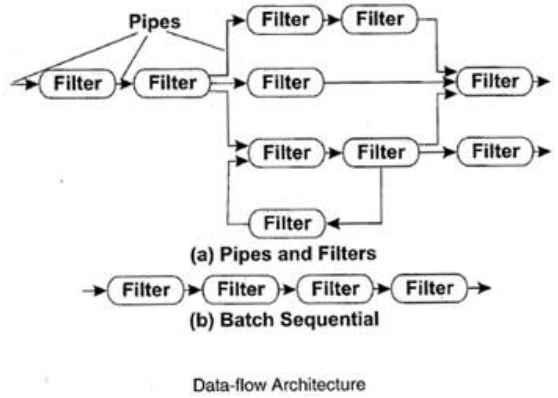
A program is only one link in the chain, or cog in the wheel

Think bigger, think systems programming!

In 'nix, everything is either a file or a process.

- Programs run as processes which
- (consume &) generate data steams which are fed as
  - Files
    - for storage, subsequent processing or analysis –
      - via buffers, file system managers, disk controllers etc.
  - Pipes
    - temporary files created and afterwards destroyed by the system to allow processes to communicate
  - Sockets
    - basically like pipes except on different processors/machines
- Which are fed to other processes… (and so on_until job done)

Inter process comms via pipes



Data-flow Architecture

Some of the most...

• boring, • terse, • error prone, • standard-free, • Incomprehensible syntax, almost	• useful, • powerful, • jam-freeing, • flexible • Incomparably powerful tools

you're ever liable to encounter
Are here, like all of life!
And they're honed for speed, C, buffers...

And runs

- with almost any
  - language
  - Platform : processor maker & family
  - Size : embedded to Web & HPC
- Faster
- Fewer resource requirements
  - Less bloatware
  - Minimal versions for embedded systems
- Free !?
- Just made for (Internet of Things) IoT
  - And others are following...e.g. Win

And runs .. on and on...

Distributions of distributions …

https://en.wikipedia.org/wiki/Comparison_of_Linux_distributions

Free!?

Digital died → assimilated into Compaq → assimilated into HP….

Digital brought minicomputers to the world… computers that were much cheaper to buy and maintain, and did not fill a room, but did the work of those which did. PDP & VAX… real start of miniaturisation and ran proprietary Oses, but first UNIX & C on a DEC-PDP. DEC also helped develop Ethernet, X-windows, Clustering, RAID, etc. to the world

Where… well – everywhere!

- Top500 – June16 –
  - All run 'nix – no Win/ OS X
  - Only 3 run Unix, the rest run Linux
  - None of the 3 running Unix are in the Top250
- And the bottom500 too probably!
- And on everything in between – robots, drones, phones,IoT, embedded sys.
  - Latest dronekit with eyes can just act as a linux box when not flying
- Google, Amazon, even Microsoft Servers often run on and now support Linux, Win10 runs basic 'bash', and Windows apps can run on 'WINE' : Windows Emulation suite for 'nix.
- Of course, VM's can run just about anything on anything
- Containers (like a minimal VM)
  - just what is needed, not the full OS on the machine.
  - Resource efficient (time, space, energy) => cheaper
- OpenStack – private cloud
- and in 2020, all Top500 run Linux!
- Speculation that future Windows will be a layer on top of Linux!?

Simple view for starters

- Past
  - Unix → versions → proprietary
  - Teaching : minix → Linux → & GNU → Open Source
  - Unix Also
    - basis for
      - BSD → Mac OS X
      - Linux → basis for Android
    - influenced Windows : similar command names and function
      - DOS : with backslash instead or forward slash for filesystem paths
      - and more recently PowerShell : except Object Oriented...
- Present
  - Ubuntu...well Kubuntu … Ubuntu with KDE
- ~~Future~~ Recent past
  - Ubuntu moved to GNOME (2020), away from Unity convergent touch interface spanning phone through tablet to desktop (around 2010's)
  - Bottom line for now – all that matters to us : bash on Ubuntu...

All about info

- Text → files → filesystems
- commands
- Editors
  - Interactive
  - Programmed
  - Regular expressions
- Bash programs from commands
- Special
  - (g/m)awk :

Ways to manage info...mostly text.

- Files
  - Create
  - Move around
  - Find & edit
- Filesystems
  - Create (files and directories)
    - Creating filesystems as in FAT, NTFS, ext, zfs etc (don't worry, just more file systems) is a much bigger issue involving appraisal of:- apps e.g. databases, processors, RAM, busses disks & performance, reliability and recovery considerations
  - Move around
  - Find & edit… by file creation, deletion & moving
- Remember 'nix was designed with text in mind, so approaches highly focused and optimised over the years for text handling are used, such as regular expressions, an almost algebraic way of specifying text patterns.