

The Final Install

- and the final upgrade too?
Software, ready to run on AFS

AFS & Kerberos Best Practices Workshop

The Final Install

- The task we undertake
- The role of certain file system properties
- Why AFS is good or might be better
- What the outcome actually is



Software is a Problem

Software is a Problem

- Installation and updates (per host)

Software is a Problem

- Installation and updates (per host)
- Customization (per host)

Software is a Problem

- Installation and updates (per host)
- Customization (per host)
- Dependency resolution (per host)

Software is a Problem

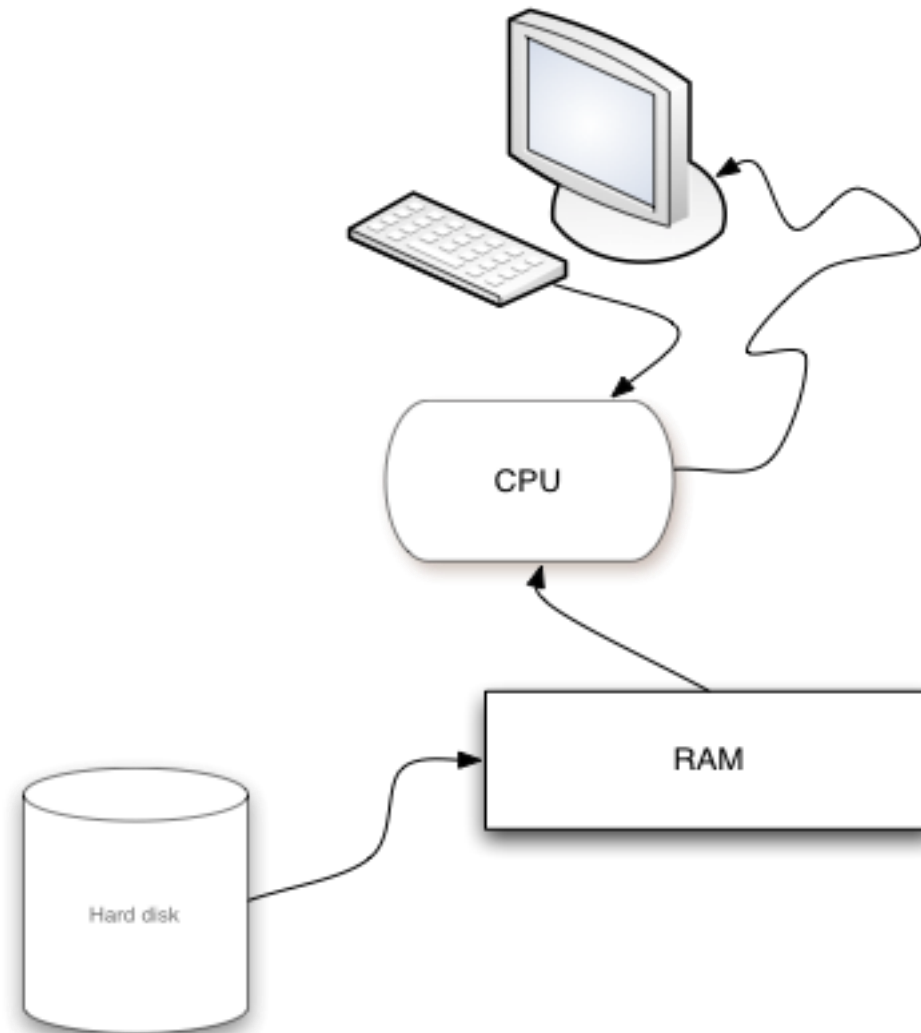
- Installation and updates (per host)
- Customization (per host)
- Dependency resolution (per host)
- The end user needs lead to clashes

Pains, continued

- Limited choice of software, because:
 - Local name space is limited
 - The dependency hell fills it up
 - Various needs => software clashes

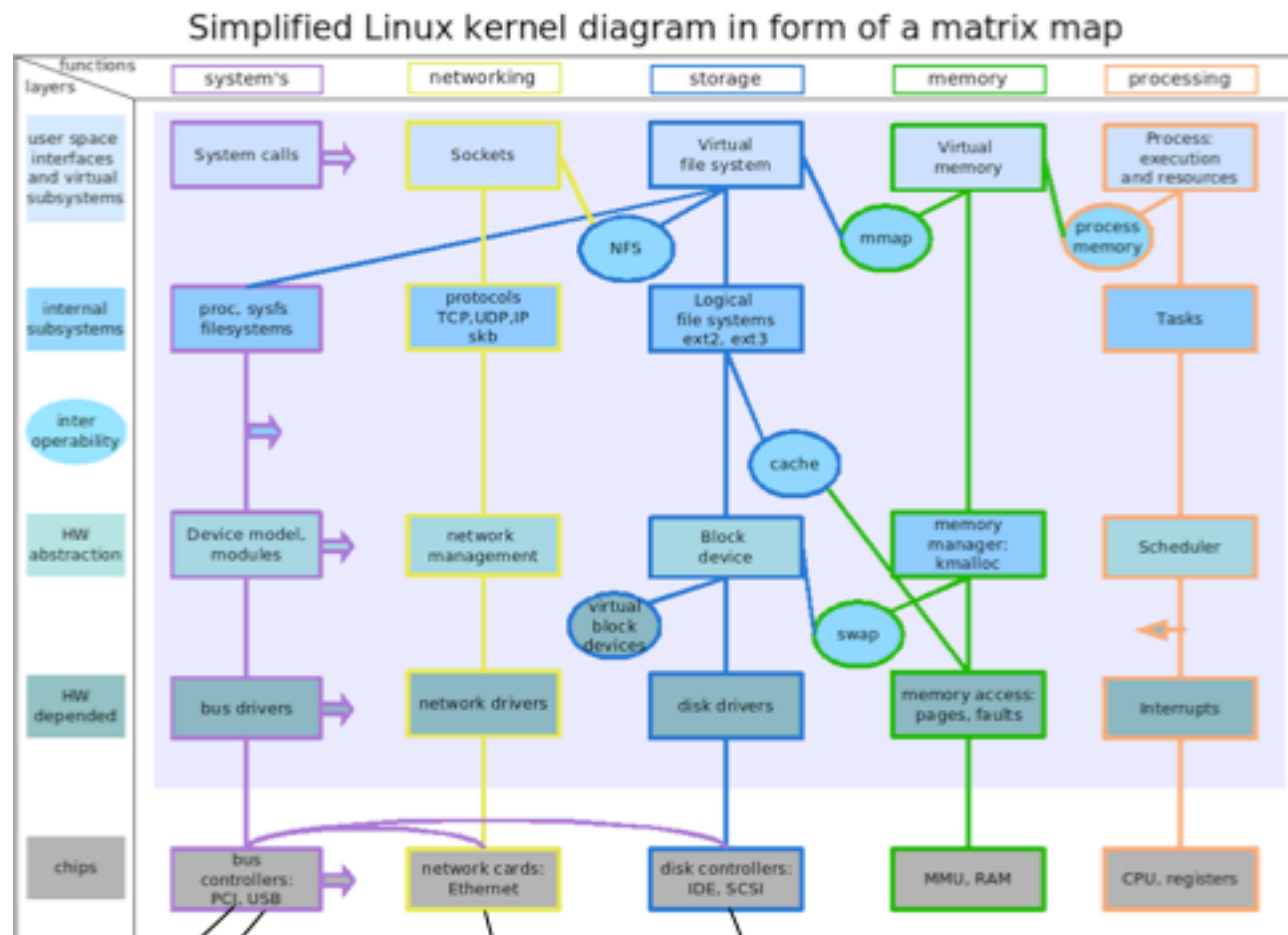
How software runs

(popular view)

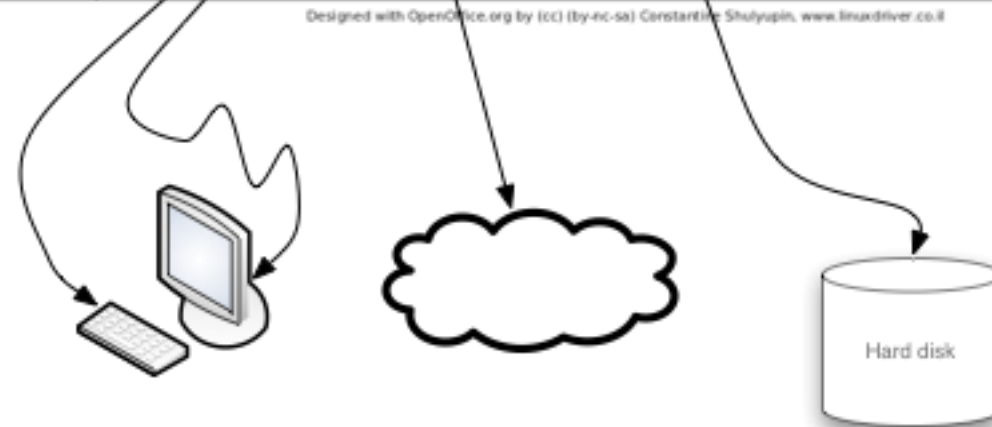


Running programs

(for those who know)

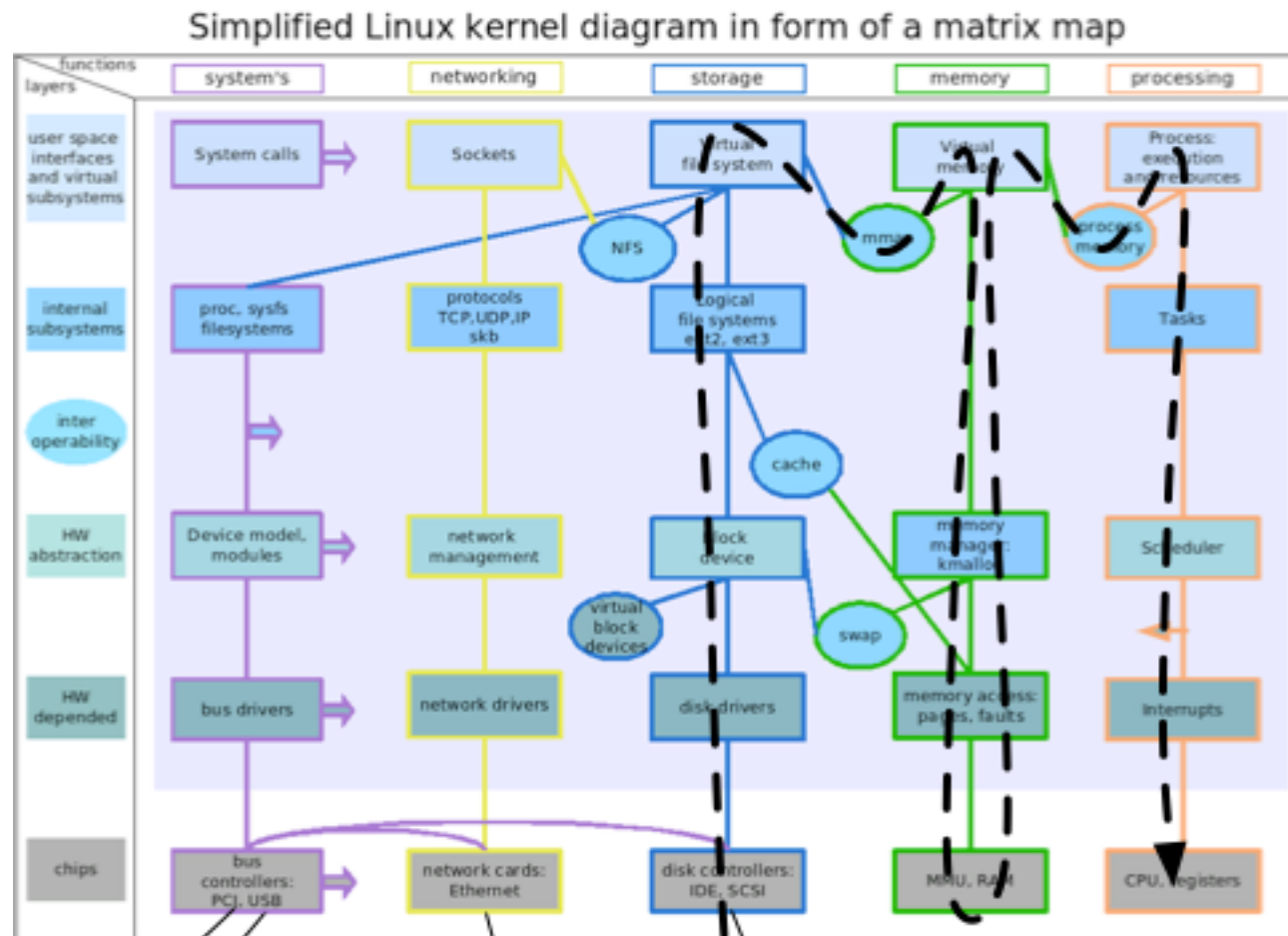


Designed with OpenOffice.org by (CC) (By-nc-sa) Constantine Shulyupin, www.bruxdriver.co.il

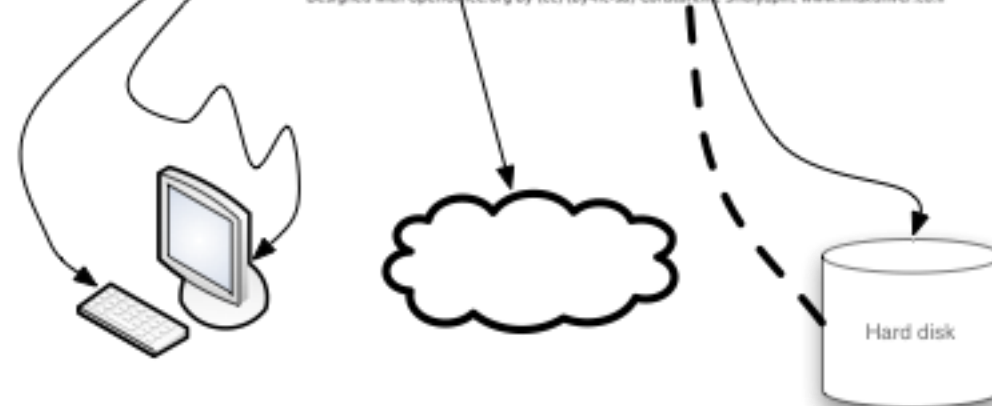


Running programs

(for those who know)

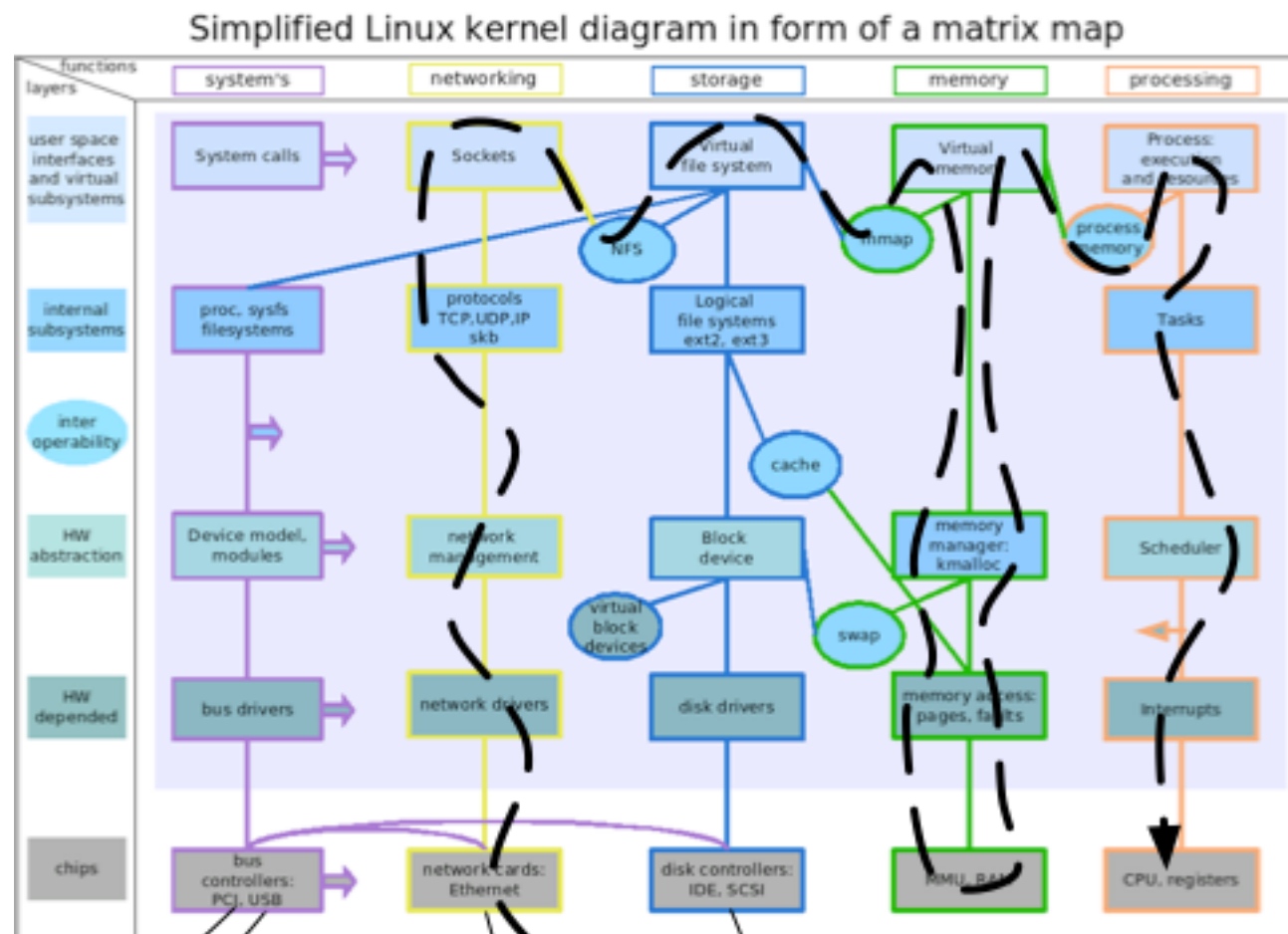


Designed with OpenOffice.org by (CC) (By-nc-sa) Constantine Shulyupin, www.linuxdriver.co.il

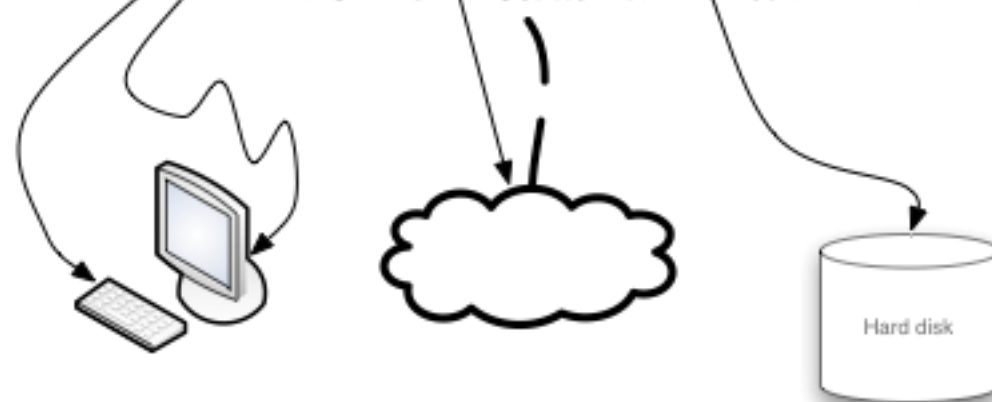


Running programs

(for those who know)



Designed with OpenOffice.org by (CC) (By-nc-sa) Constantine Shulyupin, www.linuxdriver.co.it



Why don't we ?..

- Installation would go fine
- Customization - ?
- Dependency resolution - ?
- All the conflicting needs - ?

Previous attempts

- ND
- NFS sharing of /usr and alike
- CMU/CS Depot

State of the art

- Packaging:
 - Hard for software industry
 - Painful for the users
- Per software installers/updaters
- Source build/install

User-centered?

- Configuration is efficient for groups
 - Wrongly approximated via hosts
- Installation in a home dir, even shared?
 - Still host-centric

User-centered!

- User centricity should mean that the software environment follows the user's needs, independently of which hosts are used.

➔ No host dependencies

- (at least as few as possible)

But how?

- A global file system could house all the software there is!
- But namespace usage must change

Which host dependencies do we have to deal with?

- Hardware
- Kernel
- Installation
- Configuration
- Data

Hardware

- Hardware dependencies will remain
- Fortunately most hardware has the same basic characteristics, although implemented differently
- Current trend for mobile devices increases hardware diversity, though

Kernel

- Most kernels export a POSIX API
 - ✓ (Even Windows)
- Not a big problem
- Mobile devices exhibit diversity again

Installation dependencies

- Let's eliminate installation
- A good network based file system with a global namespace helps a lot

Configuration

- Config can be both user, group and host centered
- We must eliminate unnecessary host centered config
- User and group centered config must live in the network
- A good network based file system with a global namespace helps a lot

Data

- A good network based file system with a global namespace helps a lot

A good file system?

- Needs to implement the most essential file access operations
- i.e. open,close,read,write,exec,stat etc.
- Have a global way of naming files:
 - Same name for the same file, irrespective of host
- Caching for performance

More things needed

- Authentication server->client
- Integrity protection
- Nice to have:
 - Client->Server authentication

Not much choice

- DCE/DFS
- Coda
- AFS (with dynroot)

Chips (1997-2001)

- Chalmers University of Technology
- DCE/DFS

Dapty

- A mostly complete software environment.
- Initially built on top of Coda
- Now built also for AFS
- `/afs/0l.se/dapty/{alpha,beta,gamma}`

Dapty

- About 1800 programs in PATH
- Requires:
 - an ia32-compatible processor
 - a Linux or BSD kernel
 - AFS or Coda client
 - X11 server for GUI apps

Dapty

- Supports multiple versions of the same software simultaneously
 - even on the same host
 - no installation
 - ready to run
- Follows the user, not the host
- Config may live in the network FS as well

Dapty includes e.g.

- Full gcc/g++ build environment
- All the binutils
- IceWM based desktop
- Iceape, OpenOffice etc
- ...

Disclaimer

- Current AFS implementation is a proof of concept
- Just one server
- Limited bandwidth (under 100 MBit/s)
- Some software may be aged

How does AFS perform?

starting Ooo 2.3 on a workstation - ping rtt to the server, seconds with cold and warm cache to open a text document

FS	Ping	Cold	Warm
Coda	0.35ms	29s	~4s
Coda	8/20ms	205s	~4s
AFS	18ms	154s	~4s

How was it implemented?

- Software units
- Nonsensical paths
- Eliminates name clashes
- Build templates
- Auto tools considered harmful

Hard to crack issues

- Hardware configuration
- Software trying to be intelligent, behaving stupid
- Software trying to be reliable, asking for trouble

Commercial apps

- How to put commercial apps on Dapty
- Requires another licensing mindset

Did we succeed?

- Installation and updates
- Customization
- Dependency resolution
- The end user needs / clashes

Did we succeed?

- ✓ Installation and updates
- Customization
- Dependency resolution
- The end user needs / clashes

Did we succeed?

- ✓ Installation and updates
- ✓ Customization
- Dependency resolution
- The end user needs / clashes

Did we succeed?

- ✓ Installation and updates
- ✓ Customization
- ✓ Dependency resolution
- The end user needs / clashes

Did we succeed?

- ✓ Installation and updates
- ✓ Customization
- ✓ Dependency resolution
- ✓ The end user needs / clashes

The net result

- Not long ago – heresy

The net result

- Not long ago – heresy
- "Cloud Computing"!

Making bucks

- Many products can be built upon this
- Huge potential market for software maintenance and more
- Customers can be hard to reach

<http://www.aetey.se>

[mailto: see-the-web-page](mailto:see-the-web-page)

(a moving target!)

Rune Ljungbjörn, Christer Bernéus