A Visual Introduction to Linux

Gerrit Muller
Philips Research
Prof Holstlaan 4 (WL01)
5656 AA Eindhoven
The Netherlands
gerrit.muller@philips.com

Version: 2.2

7th October 1999

Distribution

This article is evolving. It is distributed with the intention from my side to get feedback in order to improve this article. Further distribution is allowed if the document is not changed.

1 Introduction

This article makes an attempt to visualize a small number of basic Linux concepts and know how. Since Linux inherits nearly all of its characteristics from UNIX, most visualizations hold for UNIX in general.

Readers are urged to send feedback to: gerrit.muller@philips.com
in order to improve the visualization.

Main purpose of this visualization is to create overview and insight. Most diagrams are simplied on purpose, to enable the build up of overview. Complete information is abundantly available in how-to's, manual pages, info-files, web pages, faq's etcetera. The amount of information makes it difficult to maintain the overview.

2   Users, Desktops and Servers

2.1   Client server model in general

In the UNIX world the concept of servers has been used for a long time. Services are available transparently to all users, by means of the omnipresent network. The users access these services through a desktop (or client, workstation, console, . . .). Desktops exists in a spectrum of flavours from X-terminals, via diskless clients to full blown workstations. Figure 1 shows the spectrum of desktop flavours.

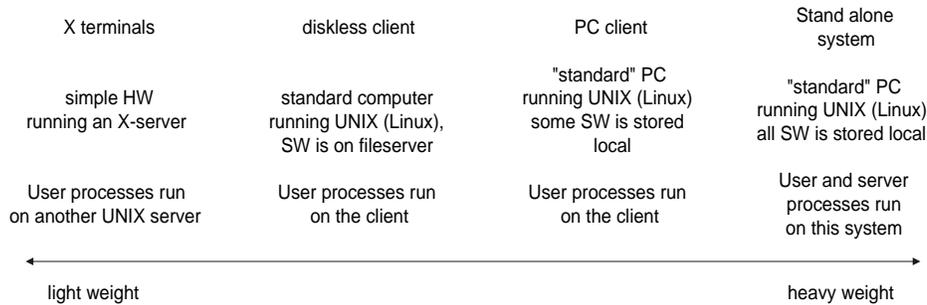| X terminals	diskless client	PC client	Stand alone system
simple HW running an X-server	standard computer running UNIX (Linux), SW is on fileserver	"standard" PC running UNIX (Linux) some SW is stored local	"standard" PC running UNIX (Linux) all SW is stored local
User processes run on another UNIX server	User processes run on the client	User processes run on the client	User and server processes run on this system

light weight ← → heavy weight

Figure 1: The spectrum of desktop flavours

Users and desktops are independent in the UNIX world, any user can use any desktop which is connected to the infrastructure. The desktop is an access point to information rather than a Personal Computer.
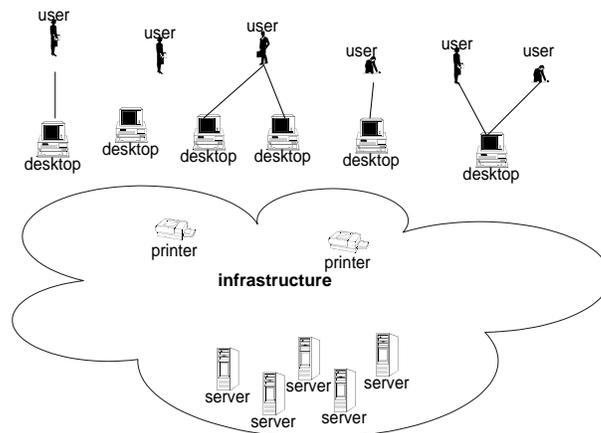

Figure 2: Users, Desktops and Servers

Figure 2 shows a simple view of the world with users roaming around and working somewhere on one or even several desktops. Part of the infrastructure, like printers, is in the offices, while the rest is invisible for instance in the basement or computer room.

2.2 A Linux desktop

Figure 3 shows the desktop, detailing some of the entities visible to the user:

desktop machine The physical machine (computer, monitor etc) on which Linux and applications are running

virtual console A console is the access point to the running computer for a human user. By virtualizing the console the physical console can be used for independent applications at the same time, although of couse only one application at the time is connected to the real console.

session On the console the user starts a session by logging in, and ends it by logging out. Dependent on the installed window manager he will be working in a windowed desktop environment. Applications run within this environment.

tools UNIX systems make heavily use of small tools. The power of UNIX is that these tools can be connected together to form more powerful tools. In combination with a windowing User Interface which enables the user to work at multiple tasks a flexible and powerful desktop is offered.
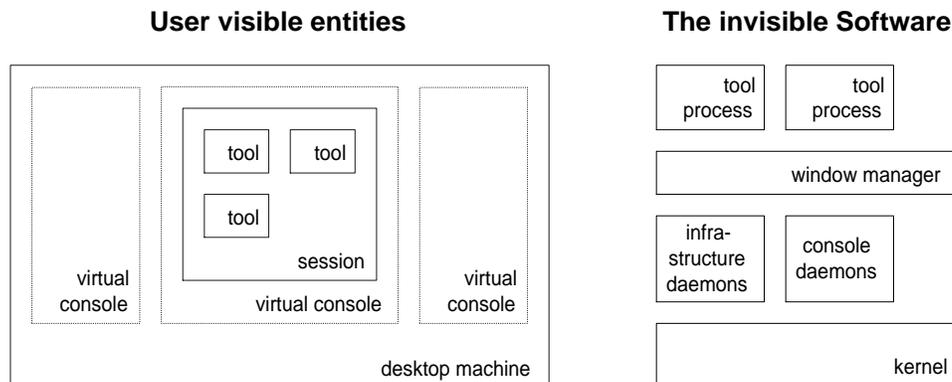
User visible entities   **The invisible Software**

Figure 3: The user visisble entities of a client and the enabling invisible software.

Naive users will not see the virtual console. The virtual console is an advanced feature of Linux enabling the user to use the physical desktop for multiple purposes, mostly used for troubleshooting in the installation phase. By using the ALT key in combination with a numeric key the visible console is selected, without disturbing the state of the other (virtual) consoles.

At the right hand side of the figure the software is shown which silently enables the desktop appearance. Kernel, infrastructure daemons and windowing manager are normally running. The applications and tools rely heavily on the forking mechanism: starting a new task is done by cloning a new process which will do the required task concurrently with the existing tasks. Every window has one or more processes running underneath to do the work.
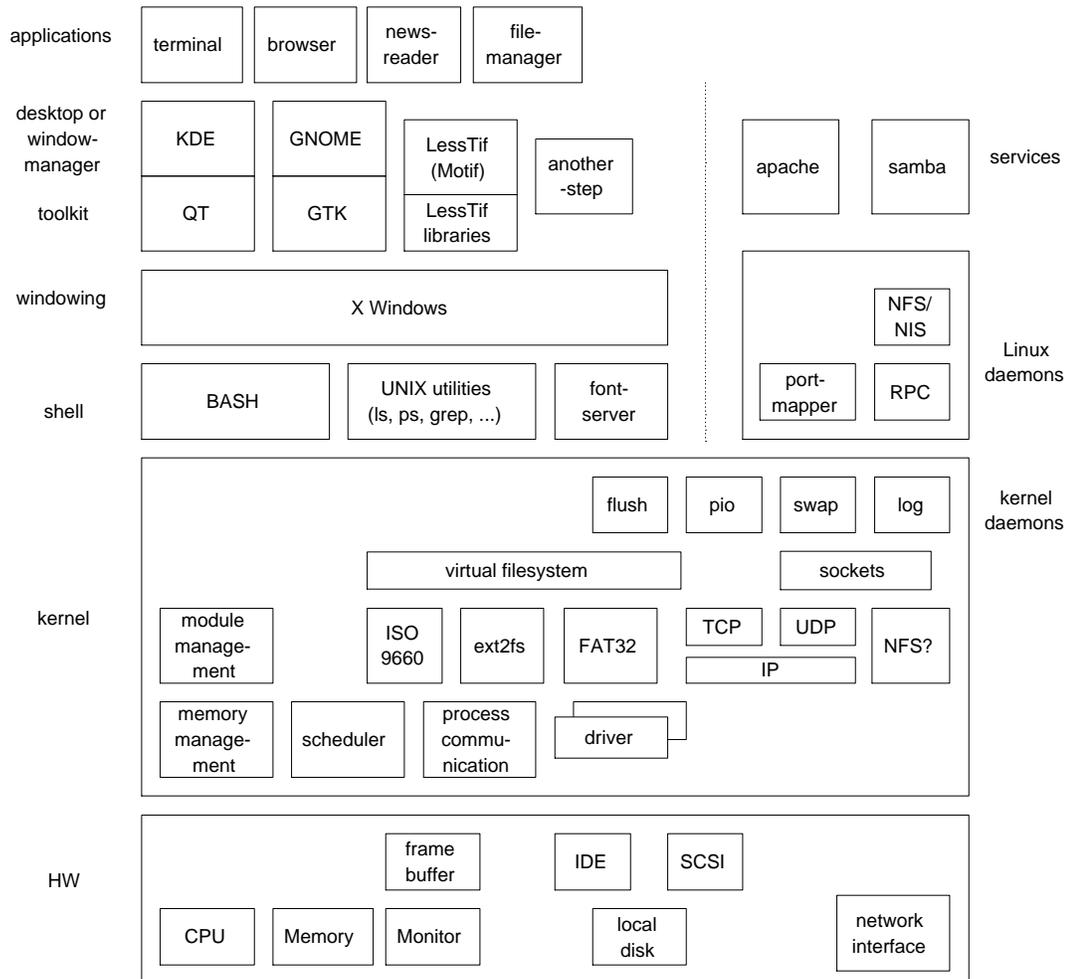
3 The map of Linux country

Figure 4: The map of Linux country

Figure 4 shows a map of most well known parts of Linux. The diagram suggests a layering, modules higher in the diagram depend on modules lower, while no dependency should exist from lower to higher modules. The entities in this diagram are loosely defined, it is the larger picture which counts.

In [2] diagrams are shown of the Linux kernel as a result of an architecture extraction study. More details are shown in [1] and [3].

The Linux file hierarchy is defined in the FHS, Filesystem Hierarchy Standard [5]. This standard, edited by Daniel Quinlan, is very readable and clearly defines the hierarchy and the reasoning behind it.

Two basic concepts are mentioned in the standard to help understand its structure:
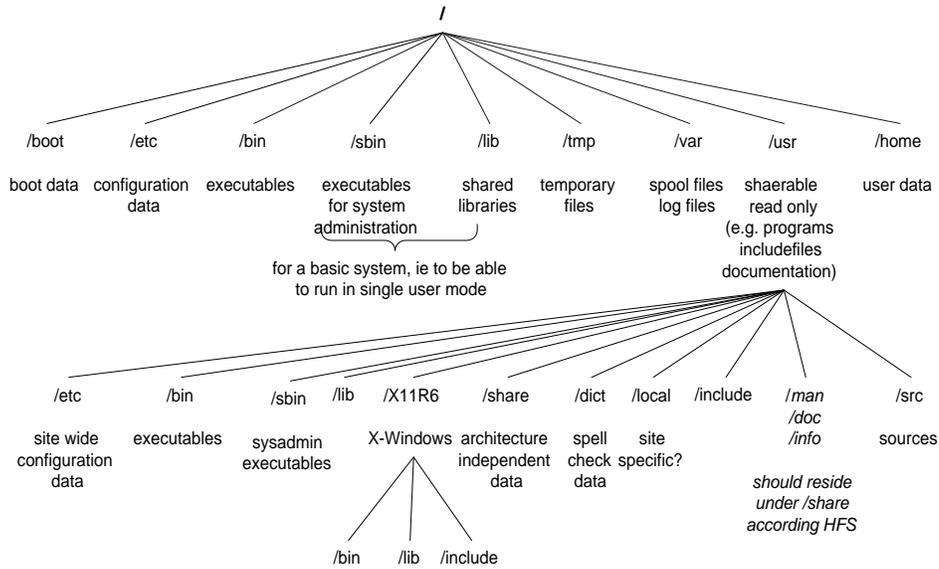
Figure 5: Typical Linux file hierarchy

	shareable	unshareable
static	/usr	/etc
	/opt	/boot
variable	/var/mail	/var/run
	/var/spool/news	/var/lock

Table 1: Summary of an FHS compliant system, table copied from FHS standard [5].

- shareable for multiple systems

- static versus variable

Shareable data will be stored on the network, easing maintenance and preserving diskspace, while unshareable data, which is system specific will be stored locally.

Static data, which is data which does not change often and is changed only by the system administrator, is mounted read only, which makes it more robust.

Table 1, taken from the FHS document, summarizes these two properties and shows a few examples of directories belonging to these categories.

Table 2 is an extension of Table 1 showing for a minimal client PC, the diskless client in figure 1, the file hierarchy. /it is this true? where reside /var/run and /var/lock in memory? uses a diskless client BOOTP during the BIOS boot? I would like to show here the configuration where the network aspect is maximized, while using "standard" PC's

	Minimal PC	Server	
	system specific	shareable, required for booting	shareable
"static"	/etc /boot	/bin /sbin /lib	/usr
variable	/var/run /var/lock	/var/log	/var/mail /var/spool /home

Table 2: The file hierarchy allocated to a minimal PC client and a server

Performance and gracefull degradation requirements can shift more directories the client. Most performance arguments tend to be too qualitative; due to the effectiveness of the UNIX (file) caching the performance is not very sensitive for the storage location of the original data.

Gracefull degradation is in daily work a very important consideration which requires the design of a network as a single system. Redundancy of the right information and services is the key to success. Often redundancy of data at all clients is not the most optimal solution, especially if you look at updates and synchronization.

4 Start up sequence

From power on to running applications is a stepwise process, called booting. Figure 6 shows this process as 6 major steps:

BIOS Boot brings the system from power up into a well defined state, to enable the booting of an OS

LILO Bootmanager enables the co-existence of multiple operating systems on a single machine. In this step the operating system to be booted and the parameters to be used are selected.

Kernel Boot This is the Linux specific start, where hardware is detected and the hardware and the kernel are initialized.

Init The Init process is the first process to be started, which in turn starts all other kernel, service and windowing processes.

Login The Login specializes the desktop configuration to the user's wishes according to the stored user profile. An environment is created, which defines the context for the applications.

Start Application Running an application in UNIX involves the forking of a new process which will execute the required application.

Every step in this start up process is an extension of configuration. These configuration extensions are defined in human readible files. Once the function and the place of these files is known this design eases problem searching (all control information is readable) and solving (because it is also editable).

Currently the trend is to provide graphical frontends for all of these textfiles, supporting a "naive" user, while the problem searching capability is preserved.
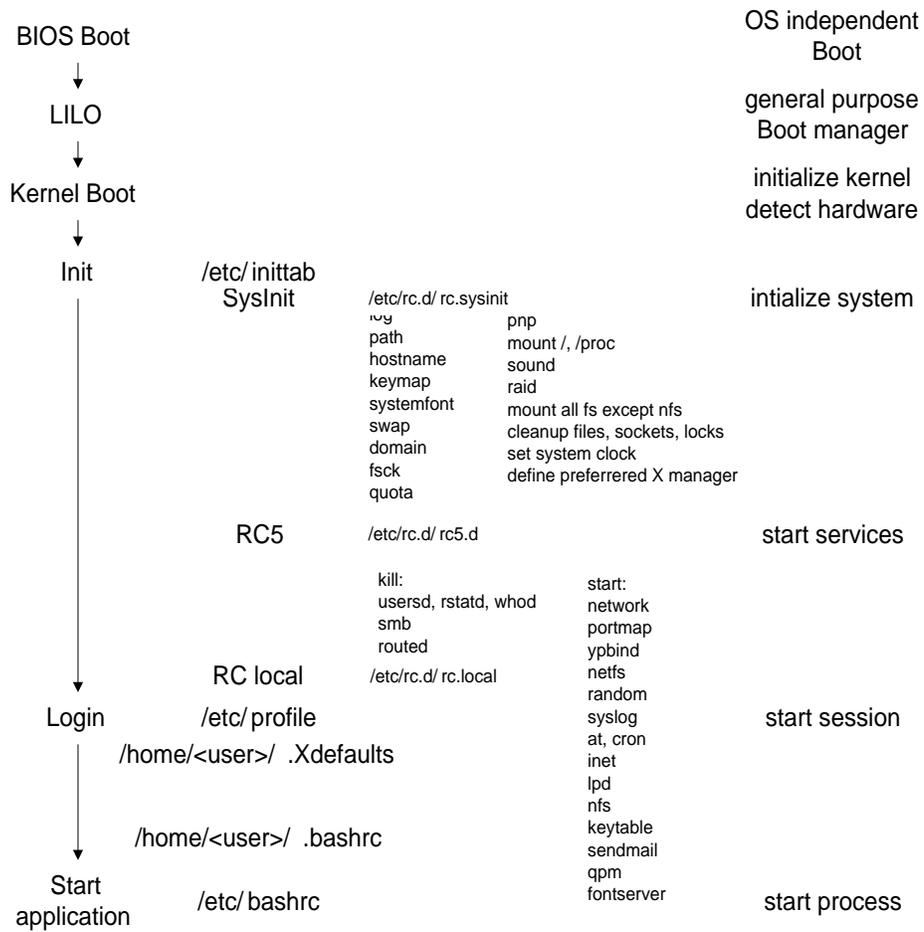
BIOS Boot   OS independent
  ↓   Boot

LILO   general purpose
  ↓   Boot manager

Kernel Boot   initialize kernel
  ↓   detect hardware

Init   /etc/ inittab
              SysInit   /etc/rc.d/ rc.sysinit   intialize system
                               log   pnp
                               path   mount /, /proc
                               hostname   sound
                               keymap   raid
                               systemfont   mount all fs except nfs
                               swap   cleanup files, sockets, locks
                               domain   set system clock
                               fsck   define preferrered X manager
                               quota

              RC5   /etc/rc.d/ rc5.d   start services

                               kill: start:
                               usersd, rstatd, whod network
                               smb   portmap
                               routed   ypbind
              RC local   /etc/rc.d/ rc.local   netfs
                                                     random
Login   /etc/ profile   syslog   start session
                                                     at, cron
        /home/<user>/ .Xdefaults   inet
                                                     lpd
                                                     nfs
              /home/<user>/ .bashrc   keytable
                                                     sendmail
Start   qpm
application   /etc/ bashrc   fontserver   start process

Figure 6: Typical Linux start up sequence

8

5 Networking

Network protocols form a layered structure. Figure 7 shows 3 layers of network adressing and the mapping to translate one address in the address of the next layer.
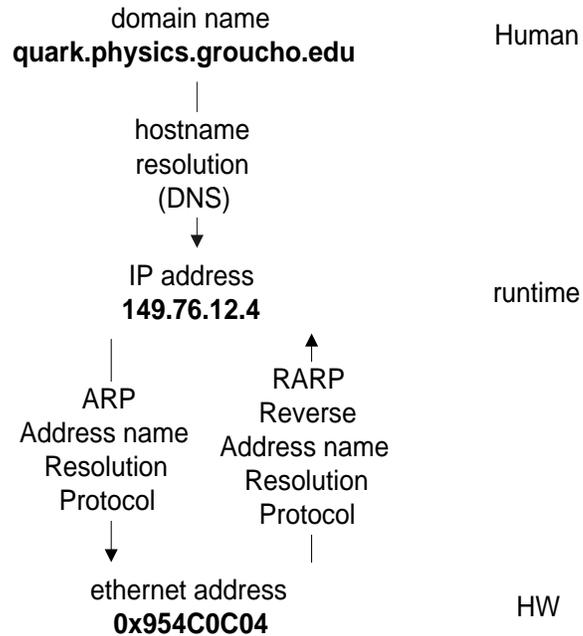
domain name
quark.physics.groucho.edu   Human

hostname
resolution
(DNS)

IP address
149.76.12.4   runtime

ARP   RARP
Address name   Reverse
Resolution   Address name
Protocol   Resolution
              Protocol

ethernet address
0x954C0C04   HW

Figure 7: Layering of network addressing and the required translations

6   Acknowledgments

Linux documentation is widely available and useful for the reconstruction of the underlying concepts. Most important contributants are therefor the numerous authors of how-to's, manual pages, etcetera.

Lacking myself most specific Linux knowhow the input of Auke Jilderda is indispensable.

References

[1] Ivan T. Bowman. Conceptual architecture of the linux kernel. URL: http://plg.uwaterloo.ca/ itbowman/CS746G/a1/, 1998.

[2] Ivan T. Bowman, Richard C. Holt, and Neil V. Brewster. Linux as a case study: Its extracted software architecture. URL: http://plg.uwaterloo.ca/ itbowman/papers/linuxcase.html, 1999.

[3] Ivan T. Bowman, Saheem Siddiqi, and Meyer Tanuan. Concrete architecture of the linux kernel. URL: http://plg.uwaterloo.ca/ itbowman/CS746G/a2/, 1998.

[4] Olaf Kirch. *Linux Network Adminstrator's Guide*. O'Reilly, Sebastopol, California, 1995.

[5] Daniel Quinlan. Filesystem hierarchy standard. URL: http://metalab.unc.edu/pub/Linux/docs/fhs/, 1999.

[6] Matt Walsh and Lar Kaufman. *Running Linux*. O'Reilly, Sebastopol, California, 1996.

[7] Lars Wirzenius. The linux system administrators' guide. URL: http://www.iki.fi/liw/linux/sag, 1998.

History

Version 2.2, october 10th 1999

- minor editorial changes

- Minor adaptions to the figures to enable re-use in presentations.

Version 2.1, august 12th 1999

- Figures scaled to textwidth

- Reference added to Bowman reverse architecting articles

- Figure 1: "Linux" replaced by "Unix (Linux)"

Version 2, august 11th 1999

- Section "Users, Desktops and Servers": decomposed in 2 subsections

- Figure 2: 2 users added which use the same desktop

- Figure 3 ("Desktop and server side by side"): replaced by a client only figure. The text is adapted to match the new figure.

- Figure 4:

    - added a label "services"
    - added toolkits
    - added LessTif
    - some minor cosmetical improvements

- table 2 changed from a general mapping into a mapping for a minimal PC configuration; The text is adapted to match the new table.

- appended the History

Version 1, august 6th 1999
Initial version.