

S
y
s
Introduction
e
m

A
d
m
i
s
t
r
a
t
i
o
U **n** *IX*

UNIX History

⌘ history

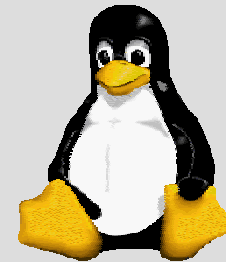
- ⌘ ultimate origins in 1969 as a platform for a “space travel” game/simulation
- ⌘ 1973 brought first C version
 - ⊗ *a novelty—all OSs until then had been in assembly*
 - ⊗ *reason for UNIX's portability*
- ⌘ 1978 saw an argument that lead to the two major versions of UNIX
 - ⊗ *System III (eventually System V) and BSD*
- ⌘ 1980s was a time of feverish development
 - ⊗ *messy*
 - neglect, innovation, political maneuvering
 - (see p. xiii, Frisch)
 - ⊗ '84: *rise of open source software movement*
 - ⊗ '86: *Mach*
 - ⊗ '87: *POSIX attempts to reconcile the different versions; X Window provides a networkable GUI*
 - ⊗ '88: *UNIX International and OSF/1 (based on Mach)*
- ⌘ 1991 Linux appears
- ⌘ 1995 saw UNIX's 25th birthday
 - ⊗ *UNIX clocks start at Jan 1 00:00:00 GMT 1970*

UNIX Status & Future

⌘ status

⏏ still messy

- ⊗ "At this point UNIX is hopelessly plural, and nowhere is this plurality more evident than in system administration."
- ⊗ (see p. xvi, Frisch)



- ⊗ has become the standard "enterprise operating environment"
- ⏏ was greatly harmed by feuding vendors
 - ⊗ SysV vs. BSD, UNIX Systems Labs., UI vs OSF, etc....

⌘ future

- ⏏ position is under attack by Windows NT
- ⏏ Linux breathing new life
 - ⊗ whether this actually amounts to anything remains to be seen

3
Monday, June 22, 2009

A
d
v
a
n
c
e
d
t
e
c
h
n
o
l
o
g
y

U
n
i
x

Unix '98

- ## ⌘ from the Open Group

 <http://www.unix-systems.org/version2/unix98.html>

📦 *deja-vu* all over again!

- ⌘ a brand that can be applied to systems conforming to V.2 of the "Single UNIX Specification."

- ⌘ organized as a family of product standards addressing different market needs:

📁 UNIX 98: the base product standard

- ☐ UNIX 98 Workstation: the base product standard plus the Common Desktop Environment (CDE)

📁 UNIX 98 Server: captures existing de-facto practice for internet/intranet services. The additional functionality over the base UNIX 98 functionality includes:

✖ the Internet Protocol Suite

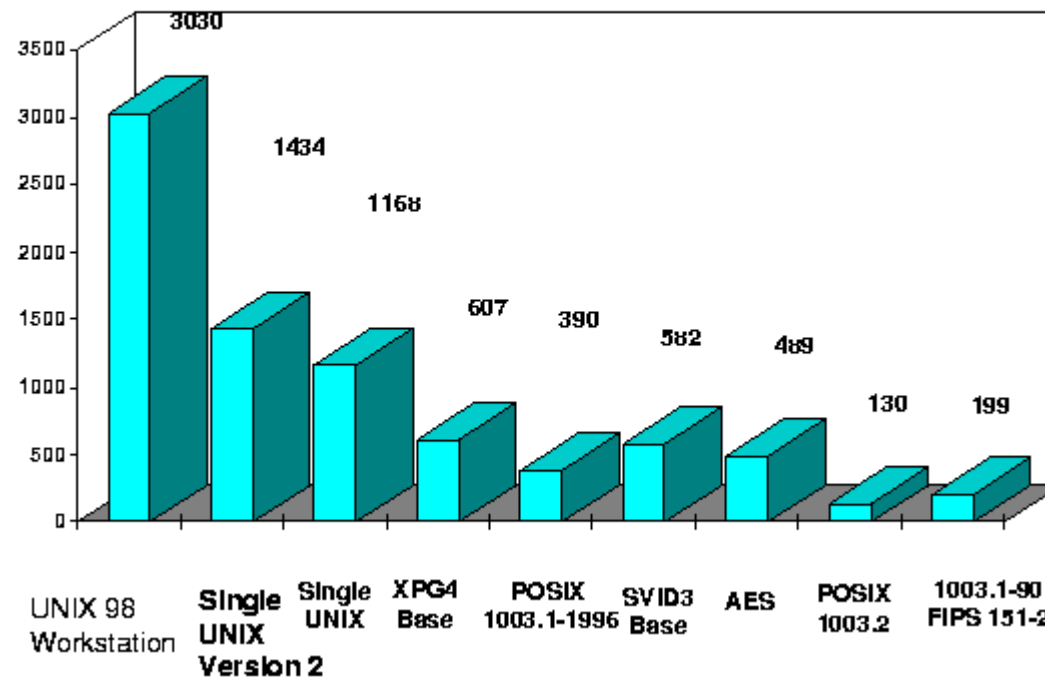
✘ *java Support*

- ☒ *internet capabilities to support network computing*

More Unix '98

- ⌘ "For those of us who count interfaces, the Single UNIX Specification Version 2 now has over 1400 programming interfaces. The UNIX 98 Workstation profile which includes CDE (and thus X11 and Motif) increases this number to over 3000 programming interfaces."

Portability functions



Linux History & Status

⌘ history

- ⏏ announced in 1991, Linus Torvalds' 'hobby' while a student
 - ⊗ *for his '386 PC*
- ⏏ based on Minix
 - ⊗ *but completely free*
 - ⊗ *a better platform for GNU software*

⌘ status

- ⏏ grown beyond anyone's expectations
 - ⊗ *runs on various architectures*
 - even the PalmPilot!
 - runs on Sun hardware better than Solaris!
- ⏏ seen as a threat to the 'establishment'
 - ⊗ *has Microsoft worried*
 - the 'halloween' leaks
 - ⊗ *has caused Sun to offer Solaris for free*
 - for personal & limited use only
 - ⊗ *is causing people to seriously re-evaluate Open Source Software*

Linux Future

⌘ moving towards POSIX compliance

⌘ increasing popularity

⌘ needs to make progress before it is acceptable as a 'mainstream' OS

⌘ *limited plug-and-play support, for example*

⌘ *a better GUI is badly needed*

- GNOME, KDE, etc.

⌘ high-end features

⌘ Beowolf clustering

⌘ *from NASA*

⌘ RAID support

⌘ still, the honeymoon period will end

⌘ rate of development is slowing down

⌘ *no more "chasing taligates"*

⌘ OSS mechanisms may not provide the certainty/support that a large enterprise needs

Wisdom

"Linux is Only Free
If Your Time is Worthless"

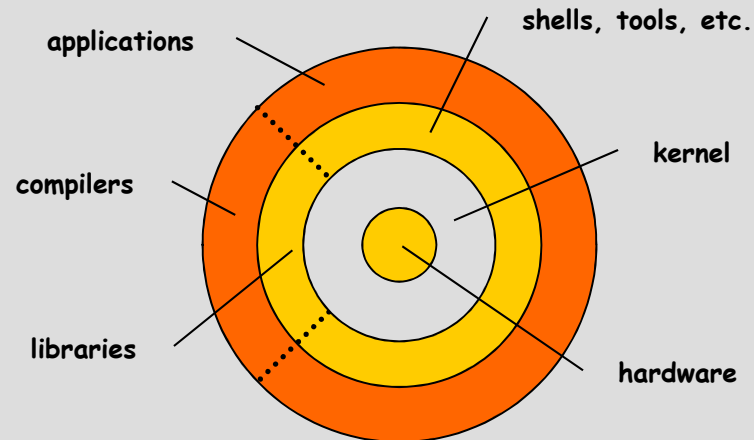
SVEN

ABE-EN-THO

UNIX

Architecture & Design

⌘ logically organized as a series of rings



⌘ divided into two major functional areas

⌘ processes

⌘ filesystem

⌘ neither of these are directly controllable

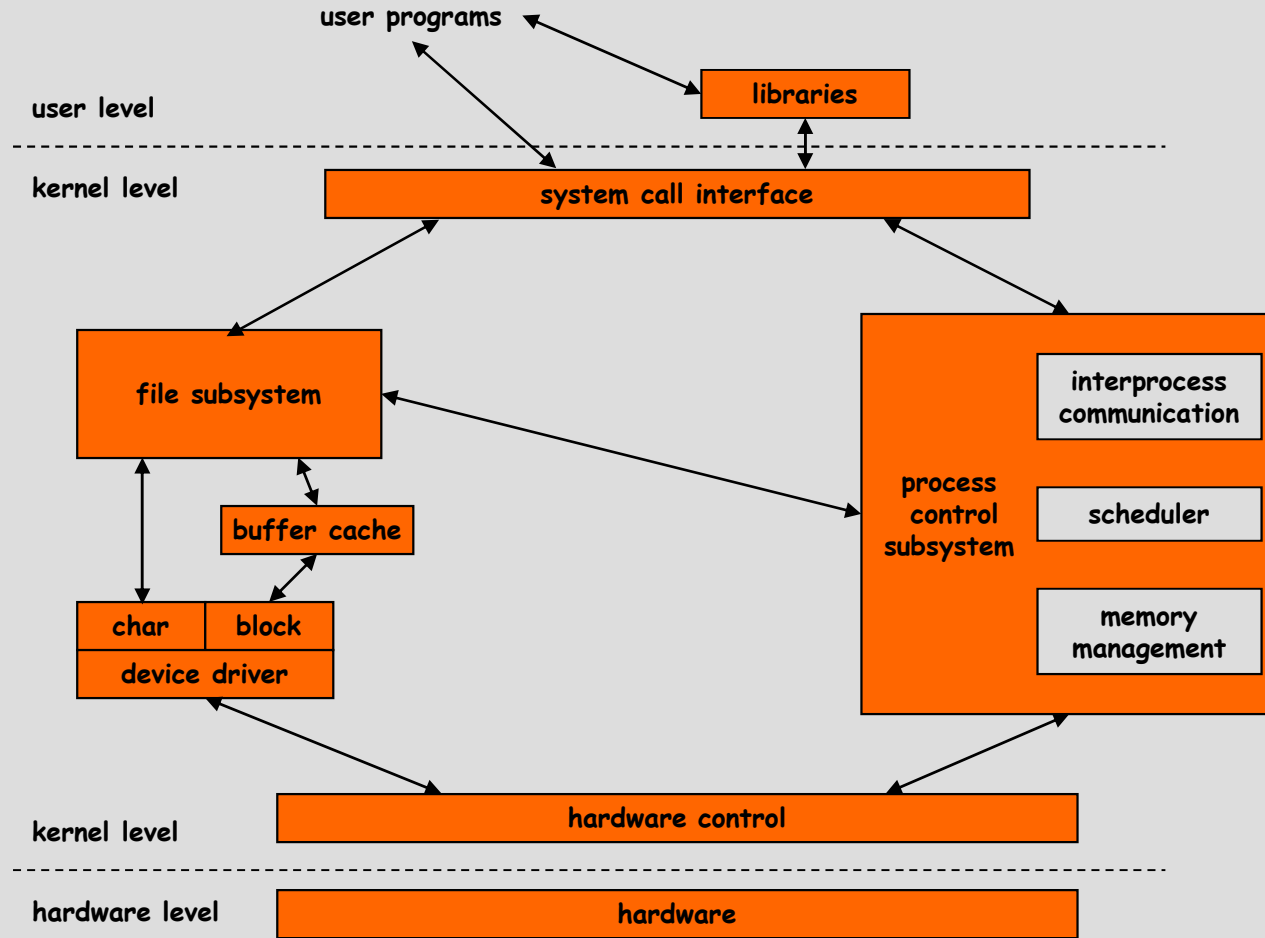
⌘ *programs must go through the standard system call interface*

- relatively simple

⌘ *contributes to portability*

- OS and application

Block Structure



S
V
E
R
E
NA
G
E

■

■

■

■

■

■

■

■

■

■

■

U
NIX

Features & Foibles

- ⌘ multiuser, multitasking
- ⌘ process control is fairly primitive in standard unix
- ⌘ virtual memory handled by paging and swapping
- ⌘ device-independent I/O
- ⌘ various filesystem types supported
 - ⌘ ufs, ext, etc2, msdos, nfs, etc.
- ⌘ text-oriented with a GUI veneer
- ⌘ various network protocols supported
 - ⌘ excels with TCP/IP so good for the internet
 - ⌘ Sun says "the network is the computer"
- ⌘ limited tuning capabilities
 - ⌘ how to improve on 'perfection'?
- ⌘ accounting/quota systems are fairly primitive
 - ⌘ due to UNIX's origins in small 'trusted' communities

Processes

⌘ programs in execution

⌘ children of init (process 1)

☐ created using the fork() system call

☐ *fork creates an almost identical copy of a process*

- open files remain open, variable values are the same, etc.
- only PID and PPID differ

⌘ scheduled preemptively

☐ multi level queue

☐ *round-robin within each queue; 100ms quantum*

☐ favors small, interactive processes

⌘ kernel maintains two basic data structures

☐ process table

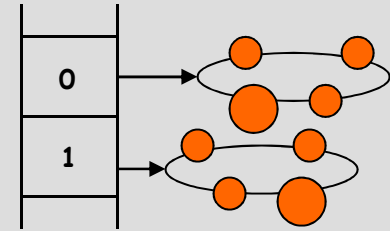
☐ *one for all processes in the system*

☐ *scheduling parameters, memory images, signals status, etc.*

☐ user structure

☐ *per process structure*

☐ *registers, system call state, open file descriptors, accounting info., kernel stack area*



Process Memory

⌘ composed of three sections

⌘ text

⌘ *actual code of the process*

⌘ *read only, shared*

⌘ *attempts to write will cause a signal*

- by default the process will die

⌘ data

⌘ *heap memory: dynamic data structures*

⌘ *read/write, individual to each process*

⌘ bss

⌘ *application's working stack*

⌘ *read/write, individual to each process*

⌘ (don't forget the registers!)

Virtual Memory Management

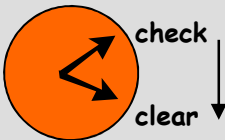
⌘ the use of secondary storage to augment RAM

☒ swapping

- ☒ *SysV originally used this exclusively*
- ☒ *whole process' image moved out to disk*
- ☒ *allows more processes to execute, no process can be bigger than real memory*

☒ paging

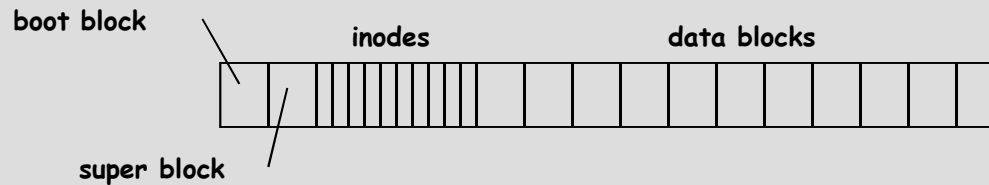
- ☒ *introduced to UNIX by BSD*
 - now used by most systems, often in addition to swapping
- ☒ *process' image is divided into small (typically 1K) pages*
- ☒ *only a process' working set is held in RAM*
 - pages not used for a while are discarded
 - *two-handed clock algorithm scans the core map every 250ms*
 - kernel tries to keep *lotsfree* pages always available
 - if it can't, it will swap complete processes out of memory
- ☒ *allows for an arbitrary number of arbitrarily large processes (subject to amount of available secondary storage)*
- ☒ *paging is implemented partly in the kernel, partly by the page daemon (process 2)*
- ☒ secondary storage always referred to as "swap space"
 - ☒ *regardless of actual mechanism used*
 - ☒ *uses (a) (special) swap partition(s) on disk via the swapon (8) command*



Filesystem Structure

⌘ SysV

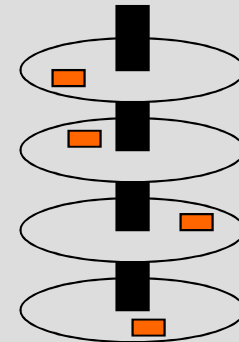
- ⌘ rather simple, c.f. DOS FAT
- ⌘ names ≤ 14 characters



- ⌘ *"Destruction of the superblock will render the filesystem unreadable."*

⌘ BSD's "Fast File System"

- ⌘ more complex structure
 - ⊗ *cylinder groups with duplicated superblocks staggered around the disk allow for more robustness*
 - ⊗ *2 block sizes: block and fragment*
 - caters for large and small files efficiently
- ⌘ names ≤ 255 chars



Fast File System

⌘ several nice features

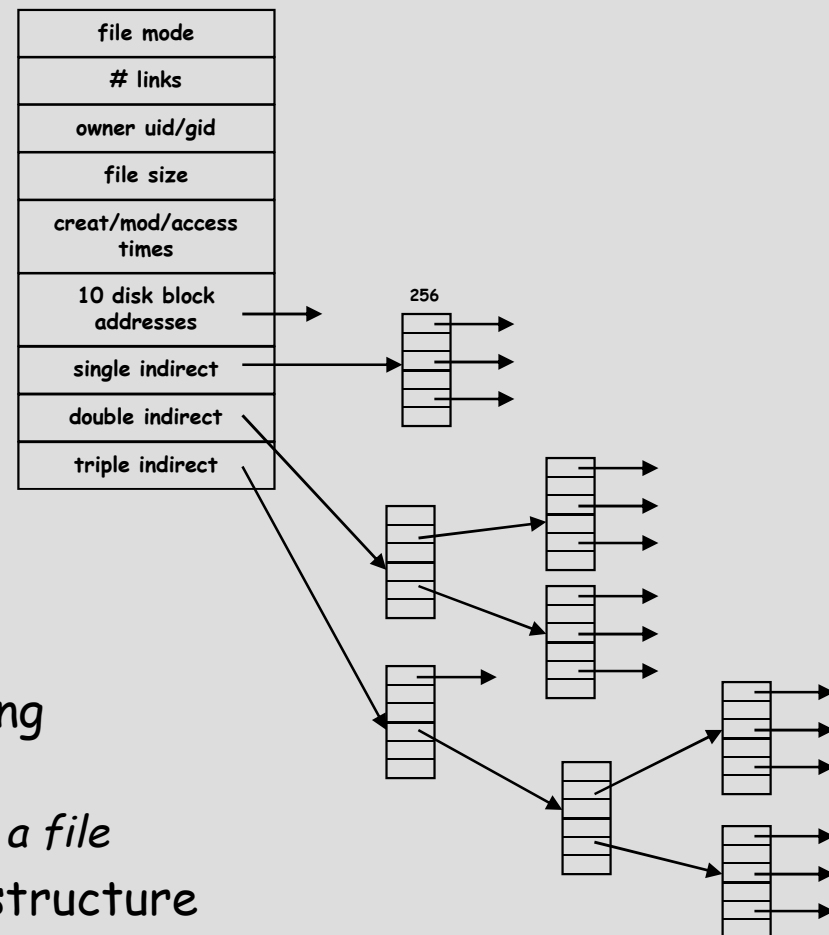
- ☒ **4 KB block size:** serves as a read-ahead. To reduce wastage by small files, has 1 K segments within each block. (4096/1024 file system)
- ☒ **cylinder groups:** file system organized as (i-nodes, data), (i-nodes, data); ... data not belonging together will be sent to different cylinder groups. Better locality among data in a group.
- ☒ **contiguous allocation:** allocates the blocks of a file contiguously
- ☒ **filesystem parameterization**
- ☒ **fault tolerance:** replicate super-block in every cylinder group. Even if one cylinder crashes, super-block can be read from another cylinder group. To further increase reliability, place the super-blocks at different positions in each cylinder group.
- ☒ **functional enhancements:** advisory file locking, long file names (upto 255 chars), symbolic links, renaming of files and directories, quotas for accounting

More Filesystem Structure

⌘ inodes

- ☑ accounting and protection information
- ☑ disk addresses
- ☑ allow for files up to 16Gb:

☒ 10+
 256+
 $256 * 256 = 65K +$
 $256 * 65K = 16Mblocks$
 $= 16Gbytes$



⌘ directory structure

- ☑ originally a file containing name/inode pair
 - ☒ *read/written just like a file*
- ☑ now is a more complex structure
 - ☒ *with a proper API*
 - ☒ *caused a lot of problems when introduced!*

Input/Output

⌘ mostly handled through device drivers

⌘ important to distinguish between:

⌘ *block*

- disks, tapes, etc.

⌘ *char*

- terminal, modem, printer, etc.
- associated with a *line discipline*
 - raw, cbreak, cooked

⌘ standard veneer

⌘ open, close, read, write, ioctl (not POSIX)

⌘ *map onto the actual device driver versions via a lookup table in the kernel*

⌘ *~500 ioctl calls!*

⌘ other calls

⌘ *mount, umount, chmod, umask, creat*

- may not directly map onto driver functionality

SysV vs. BSD

BSD
UCB
academic/internet
SunOS

SysV
AT&T
commercial
Solaris, IRIX, SCO

filesystems
init/system startup
accounting
printing
stream I/O
device configuration
cron facility
process scheduling
memory management

much cross-fertilization
AIX, HP/UX, OSF/1, Linux

"You can determine the ultimate origins of any UNIX variant by examining two items:
whether it uses */etc/inittab* for booting and the format of the raw accounting file."

SysV
to
m

A
d
m
-
i
s
t
r
a
t
e
g
y

U
n
IX

More Features & Foibles

⌘ root (user 0) is 'superman'

⌘ no constraints whatever

⊗ *rm -rf /etc*

⌘ the command-line is king

⌘ especially for administration

⊗ *SMIT, admintool, etc. partially useful*

- slow

- opaque

- can actually increase the learning curve

⊗ *still need a good knowledge of the real toolkit*

- when things really go wrong, GUI tools are generally helpless

⌘ many commands, many options

⌘ interrelationships/procedures aren't fully explained

⌘ SEE ALSO manual section helps

⌘ easy to mess things up

⊗ *only do 1/2 the complete job*

⌘ infinite variation of commands, options, procedures between platforms

⌘ need to be flexible and adaptable

⌘ need an elephant's memory!