Unix on My Mind

William Cheswick

ches@cheswick.com

Lumeta Corp.

A note on these slides

These slides: Powerpoint *vs* prosper(1)

- I’ve used Powerpoint for some 10 years
- I still don’t consider myself an expert: slide-dragging mystery
- Feature-rich, but still doesn’t have the slide count on the bottom
- Powerpoint: Proprietary format. Will it be readable in 100 years?
- Must run Windows to use it. It’s the chief reason I have a Windows partition.
- Importing cut-and-paste text always has font and size issues

The trouble with prosper

- Moving old stuff in from Powerpoint: mostly retyped
- Images and graphs are going to be more work
- Powerpoint’s slide sorter really is useful
- pdf is proprietary, too. But at least the input is ASCII.
- WYSYWIG *vs* WYSIAYG
- Not sure how my clients (*i.e.* you) will like it
- Not sure how X and xpdf will work on the world’s projectors

Unix on my Mind

Introduction

- Title, and most of the concepts, are stolen directly from Doug McIlroy, and a talk he gave with the same name in the early 1990s.[6]
- Also from “Unix isn’t Unix Any More”, by Norman Wilson.
- Many other ideas from many others, including Ken Thompson, Dennis Ritchie, Jon Bentley, Brian Kernighan.
- Any misrepresentations of their ideas in this presentation is my fault, not theirs

Why am I giving this talk

The conceptual integrity of a system determines its ease of use. — Brooks[3]

- Add some perspective for younger listeners
- Hope to motivate developers to “do it right.”
- Suggest projects to those with free time or seeking research ideas

Outline

- My summary of some of the Unix philosophy
- Rants about various minor transgressions and related issues in commands, editors, languages, GUIs
- A little about security
- Some tool samples from Lumeta
- Conclusion

Unix’s Big Ideas: My summary

Unix’s Big Ideas

- Design and build software, even operating systems, to be tried early, ideally within weeks. Don’t hesitate to throw away the clumsy parts and rebuild them.
- Expect the output of every program to become the input to another, as yet unknown, program. Don’t clutter output with extraneous information. Avoid stringently columnar or binary input formats. Don’t insist on interactive input.
  - “I couldn’t sleep this weekend, so I rewrote the TCP/IP stack.” — Phil Winterbottom
  - I have found that version 1 is usually wrong, and it takes two tries to get some non-trivial program about right.

General

Many of these ideas predate Unix. Unix brought them together, and added many excellent new ones.

- small, clean system interface. (We’ve lost a lot of that, alas.)
- if you have nothing to say, don’t say anything. ls in an empty directory should say nothing. ditto grep, and everything else. In the old days, one would see messages like "EOF ENCOUNTERED" at the end of every copy.
- Source code was available!
- make(1) is a wonderful tool

commands

- a command should do one thing, and do it well
- command line interpreters (shells) are not privileged, and not part of the operating system. Any one can write or modify one without further permission. (COMMAND.COM and fast copying)
- principle of least surprise. Things should work the same. No corner cases.

pipes and filters

- pipes let you connect the output of one command to the input of another. Doug McIlroy insisted on this.
- commands are usually built as filters, for use in pipes.
- This promotes clean interfaces, and reuse
- filters fit nicely into regression tests
- I will show some newish samples later

file systems

- directory structured file system, including . and .. for the current and parent directory; and relative and absolute path names
- most files are small, so the operating system must deal with them efficiently (A lesson I think Ken forgot, or should not have overlooked in Plan 9)
- file name extensions do not encode any special meaning for the operating system. Sure, there are conventions, but you can name a C program Pascal.borg and still compile and run it.
- file metadata is kept with the file contents (the inode), not the name of the file.

short names

- ls, pwd, etc. creat(2) should have been create(2)
- you are only a novice for a short time, then you use what you’ve learned for a long itme

VMS and other early users make shell aliases for “dir.”

I should make an alias named “ls-l”. How about “,”

documentation

- manpage notation is another great idea in Unix. See man(1)
- commands should be described as succinctly and completely as possible in man pages. tutorials belong in separate papers.
- single-character options
- options should be relevant to the command.
- commands are often readily found in the permuted index.
- short usage summary is appropriate when a command is mis-typed

files

- lines of text are reasonable units to work with. configuration, logs, etc.
- line editor: ed, or qed

unprivileged command processor

- A shell is just a program. You can write your own, or modify the existing one.
- source code is available for everything. This means you can learn how things work, and make local modified copies if you want to.
- line editor: ed, or qed

quick prototypes, little languages

- a fast prototype is better than specifications.
- A todo list is fine list of requirements. We have a lot of wonderful prototyping tools these days.
- easily written and tested
- usually easy to tell if they are right

Unix: First Impressions

Bad!

- output of one command feeds into the next. Preposterous! What do you do with the printer heading? ls(1) is ugly.
- cryptic command names
- C and its pointers: ugly! It wasn't a bad assembler, better than PL360, Wirth's structured assembly language for the IBM.[8]
- way too many asterisks, and too much pointer arithmetic, not typesafe! printf was nice (Pascal got printing wrong, methinks. writeln())
- Why would I bother learning and using this language? (An opinion I hold today about Perl.)

Good

- single character input available (DEC also had it on their myriad of operating systems), so screen editors were feasible. TECO and line noise.
- tree structured file systems
- typeahead
- easy parallel execution of processes
- ASCII character set (aim high!)

Unix: in the swimming pool

My editors stripped off trailing blanks on text lines. They didn't mean anything, and just made files longer, which was an issue: cards were 80 columns long, and data files from cards often preserved that. Trailing whitespace was not significant.

- no surprises
- inodes, not files, with linking and aliasing easy.
- Linking commands together is not stupid and inconvenient, if the commands are designed right, and not for the line printer.

Rants and comments

Commands

ls is broken

```
$ ls
amw2005.pdf                     httpd-docs-2.0.49.en.pdf
chew-syscall.ps                 images
html                            music
html40.tar                      photos

$ ls | cat -
amw2005.pdf
chew-syscall.ps
html
html40.tar
httpd-docs-2.0.49.en.pdf
images
music
photos
```

ls -l is broken

```
np:~/proj/talks/uomm$ ls -l | sed 10q
total 2146
drwxr-xr-x  2 ches  ches       512 May 17 08:02 CVS
-rw-r--r--  1 ches  ches       982 May 17 07:11 Makefile
drwxr-xr-x  2 ches  ches       512 May 17 06:47 Unused
-rw-r--r--  1 ches  ches      1158 Jan  4 19:46 abstract
-rw-r--r--  1 ches  ches      3710 May 17 07:12 before.tex
-rw-r--r--  1 ches  ches      6682 May 17 07:04 beforedetail.tex
-rw-r--r--  1 ches  ches       812 May 17 08:01 bib.aux
```

- The first line is *never* useful
- The first line breaks the filter mechanism

Real cats don’t have options.

```
CAT(1)                   FreeBSD General Commands Manual                 CAT(1)

NAME
     cat -- concatenate and print files

SYNOPSIS
     cat [-benstuv] [file ...]

...

-v      Display non-printing characters so they are visible.  Control
        characters print as ‘^X’ for control-X; the delete character
        (octal 0177) prints as ‘^?’.  Non-ASCII characters (with the high
        bit set) are printed as ‘M-’ (for meta) followed by the character
        for the low 7 bits.
```

grep(1) gripe

```
$ grep stupid tutorial.tex
\item   Linking commands together is not stupid and ...

$ grep stupid *
big.tex:   9. Rule of Representation: Fold knowledge...
first.tex:\item Linking commands together is not stu...
languages:Naming things (as in ports) p5-<something>...
tutorial.tex:\item      Linking commands together is...
tutorial.tex:   9. Rule of Representation: Fold know...
```

grep(1) gripe

grep -h fixes the surprise. Note -H is unnecesary:

```
$ grep stupid tutorial.tex /dev/null
tutorial.tex:\item      Linking commands together is...
tutorial.tex:   9. Rule of Representation: Fold know...
```

sed(1) *vs* head(1)

- head(1) is redundant, use sed 10q
- neither sed 0q nor head -0 works

Now broken: fortune(6)

- Used to pick a random text line out of a file
- Now it is full of databases and other goo, and the original, powerful use, is broken:

```
np:~$ seq 10 >x
np:~$ fortune x
Warning: file "x.dat" unreadable
fortune:x not a fortune file or directory
```

Editors

Editors

- mode (vi(1)) and modeless (emacs(1)) editors
- (I used to design modeful editors. I have no tolerance for them now.)
- Sometimes text is the way to go. I use jove(1)
- My favorite editor is sam(1)

sam

- compile it (/usr/ports/editors/sam) and see how small it is
- Smartest use of the mouse I have ever seen.
- Window commands can be learned in a couple of minutes
- command-line stuff is very powerful and useful, but can be skipped
- can apply commands over many files at once.

Languages

Scripting lanugages

- These have really worked out well
- In 2003, 50% of Lumeta’s code was scripted, the remaining was C.
- Easy prototypes, easy debugging, powerful when the right tools are available.
- Text-line databases can go very far and quite fast with scripting tools.

The file system is not a good database when there are tens of thousands of files. It is good when the files are fewer, and longer.

Perl

- Perl: I started learning it twice, and gave up twice from revulsion.
- Perl: Many functions, no discipline, no beauty.
- Perl: over 90% of the Perl programs I encounter are hard to read.
- The libraries are a terrific idea
- Names (in ports) of p5-xxx are dumb

Other Languages

- Python: Looks promising, and I indent anyway
- Ruby: Haven't checked it out yet.
- PHP. Insecure?
- C++: Bulky, slow to compile, but nicely robust. I haven't worked on a project that needed it, but it would be nice for some.
- Java: see C++, mostly

bash

- new tricks for an old dog
- tab file name completion is now a part of my life, even in the wrong places
- control-V is annoying
- I like command history better than Rob Pike's scroll solution.

C

- My chief non-scripting language
- Still ugly and dangerous
- I steal idiom samples for tricky stuff, like pointers to functions, casting sockets
- ifdef considered harmful[7]

GUI *vs* Command Line

Typing *vs* clicking

- The TTY is what we had. Are we a slave to this past?
- No: for most operations, typing is quicker and more accurate.
- It can also be re-edited (bash(1) is good), and stored as a script. Clicking: not so much.
- Text, and filters, are easy to run regression tests on

Clicking certainly has its place

- Interactions with graphs, maps, images
- Interactions with structured data like file system trees. (My MP3 library).
- Macintosh (Windows) interaction is passible to good, depending on the application
- Uniform interface cuts training time
- But I think you can’t reach the speeds and efficiency of typing for most tasks for an experienced user.
- Most people are only novices to a system once.

conf files are better than GUIs

- GUIs make configuration harder, not easier
- Config files can contain comments and examples
- Config files can be packaged up for common cases
- Easy to back up, and share with your friends
- Fairly consistent formats
- They are little languages

Config files!

/etc/defaults/rc.conf

/etc/defaults/pccard.conf

/etc/defaults/periodic.conf

/etc/netconfig

/etc/X11/xorg.conf

/etc/bluetooth/hcsecd.conf

/etc/mail/mailer.conf

/etc/mail/mailer.conf.old

/etc/ppp/ppp.conf

/etc/rc.d/ldconfig

/etc/rc.d/rcconf.sh

/etc/ssh/ssh_config

/etc/ssh/sshd_config

/etc/apmd.conf

/etc/auth.conf

/etc/devd.conf

/etc/devfs.conf

/etc/dhclient.conf

/etc/inetd.conf

/etc/login.conf

/etc/mac.conf

/etc/newsyslog.conf

/etc/portsnap.conf

/etc/pf.conf

/etc/snmpd.config

/etc/sysctl.conf

/etc/syslog.conf

/etc/usbd.conf

/etc/manpath.config

/etc/login.conf.db

/etc/make.conf

/etc/ntp.conf

/etc/manpath.config.bak

/etc/rc.conf

/etc/resolv.conf

/etc/nsswitch.conf

/etc/host.conf

/etc/make.conf.bak

/etc/nsmb.conf

Config files: cupsd.conf

```
cupsd.conf:

#
# DefaultLanguage: the default language if not speci
# If not specified, the current locale is used.
#

DefaultLanguage en
```

Some Thoughts on Unix Security

Host security

- not much has changed over the decades
- we do have better network services (i.e. ssh(1))
- Once you get a user account, I think the game is mostly over
- The only standard sandbox is chroot (see below)

A Measure of host security

```
find / -perm -4000 -user root -print 2>/dev/null |
        tee /tmp/setuid |
        wc -l

      45

uname -a
FreeBSD ches.corp.lumeta.com 6.1-PRERELEASE FreeBSD
```

Setuid-root counts

System	Files	Comments
AIX 4.2	242	a staggering number
BSD/OS 3.0	78	
FreeBSD 4.3	42	someone’s guard machine
FreeBSD 4.3	47	2 appear to be third-party
FreeBSD 4.5	43	see text for closer analysis
HPUX A.09.07	227	about half may be special for thi
Linux (Mandrake 8.1)	39	3 appear to be third-party
Linux (Red Hat 2.4.2-2)	39	2 third-party programs
Linux (Red Hat 2.4.7-10)	31	2 third-party programs
Linux (Red Hat 5.0)	59	
Linux (Red Hat 6.0)	38	2–4 third-party
Linux 2.0.36	26	approved distribution for one uni

Setuid-root counts (cont.)

System	Files	Comments
Linux 2.2.16-3	47	
Linux 7.2	42	
NCR Intel 4.0v3.0	113	34 may be special to this host
NetBSD 1.6	35	
SGI Irix 5.3	83	
SGI Irix 5.3	102	
Sinux 5.42c1002	60	2 third-party programs
Sun Solaris 5.4	52	6 third-party programs
Sun Solaris 5.6	74	11 third-party programs
Sun Solaris 5.8	70	6 third-party programs
Sun Solaris 5.8	82	6 third-party programs
Tru64 4.0r878	72	

Setuid-root counts (new)

System	Files	Comments
FreeBSD 4.11 stable	53	netmapper
FreeBSD 6.1-PRE	50	ches computer
2.6.16-1.2096_FC4	37	Mythtv backend

setuid-root list: My web server

/usr/bin/at
/usr/bin/atq
/usr/bin/atrm
/usr/bin/batch
/usr/bin/chpass
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/ypchpass
/usr/bin/ypchfn
/usr/bin/ypchsh
/usr/bin/lock
/usr/bin/login
/usr/bin/opieinfo

setuid-root list: My web server (cont)

/usr/bin/opiepasswd

/usr/bin/passwd

/usr/bin/yppasswd

/usr/bin/rlogin

/usr/bin/rsh

/usr/bin/su

/usr/bin/crontab

/usr/bin/lpq

/usr/bin/lpr

/usr/bin/lprm

/usr/libexec/pt_chown

/usr/local/lib/mgetty+sendfax/faxq-helper

/usr/local/bin/sudoedit

setuid-root list: My web server (cont)

/usr/local/bin/sudo

/usr/local/bin/procmail

/usr/local/sbin/suexec

/usr/sbin/authpf

/usr/sbin/mrinfo

/usr/sbin/mtrace

/usr/sbin/ppp

/usr/sbin/pppd

/usr/sbin/sliplogin

/usr/sbin/timedc

/usr/sbin/traceroute

/usr/sbin/traceroute6

/usr/X11R6/bin/xterm

setuid-root list: My web server (cont)

/usr/X11R6/bin/Xorg

/usr/compat/linux/usr/libexec/pt_chown

/bin/rcp

/sbin/mksnap_ffs

/sbin/ping

/sbin/ping6

/sbin/shutdown

setuid-root list: My web server (cont)

/usr/bin/at	
/usr/bin/atq	
/usr/bin/atrm	
/usr/bin/batch	I don’t use it
/usr/bin/chpass	I don’t use it
/usr/bin/chfn	I don’t use it
/usr/bin/chsh	
/usr/bin/ypchpass	I don’t use it
/usr/bin/ypchfn	I don’t use it
/usr/bin/ypchsh	I don’t use it
/usr/bin/lock	I don’t use it
/usr/bin/login	
/usr/bin/opieinfo	I don’t use it

setuid-root list: My web server (cont)

/usr/bin/opiepasswd	I don’t use it
/usr/bin/passwd	
/usr/bin/yppasswd	I don’t use it
/usr/bin/rlogin	I don’t use it
/usr/bin/rsh	I don’t use it
/usr/bin/su	
/usr/bin/crontab	
/usr/bin/lpq	
/usr/bin/lpr	
/usr/bin/lprm	
/usr/libexec/pt_chown	I don’t use it
/usr/local/lib/mgetty+sendfax/faxq-helper	I don’t use it
/usr/local/bin/sudoedit	I don’t use it

setuid-root list: My web server (cont)

/usr/local/bin/sudo	
/usr/local/bin/procmail	
/usr/local/sbin/suexec	
/usr/sbin/authpf	I don’t use it
/usr/sbin/mrinfo	I don’t use it
/usr/sbin/mtrace	I don’t use it
/usr/sbin/ppp	I don’t use it
/usr/sbin/pppd	I don’t use it
/usr/sbin/sliplogin	I don’t use it
/usr/sbin/timedc	I don’t use it
/usr/sbin/traceroute	
/usr/sbin/traceroute6	
/usr/X11R6/bin/xterm	

setuid-root list: My web server (cont)

/usr/X11R6/bin/Xorg	I don’t use it
/usr/compat/linux/usr/libexec/pt_chown	I don’t use it
/bin/rcp	I don’t use it
/sbin/mksnap_ffs	
/sbin/ping	
/sbin/ping6	
/sbin/shutdown	I don’t use it

setuid-root list: only programs I use

/usr/bin/at
/usr/bin/atq
/usr/bin/atrm
/usr/bin/chsh
/usr/bin/login
/usr/bin/passwd
/usr/bin/su
/usr/bin/crontab
/usr/bin/lpq
/usr/bin/lpr
/usr/bin/lprm
/usr/local/bin/sudo
/usr/local/bin/procmail
/usr/local/sbin/suexec
/usr/sbin/traceroute
/usr/sbin/traceroute6
/usr/X11R6/bin/xterm
/sbin/mksnap_ffs
/sbin/ping
/sbin/ping6

setuid-root list: least privilege

/usr/bin/at	should be run by cron?
/usr/bin/atq	at commands a config file in user’s directo
/usr/bin/atrm	at commands a config file in user’s directo
/usr/bin/chsh	user-supplied in home directory?
/usr/bin/login	why setuid? Should be run only as root
/usr/bin/passwd	ok
/usr/bin/su	ok
/usr/bin/crontab	ok, needs to be user. Get file from HOME
/usr/bin/lpq	user lp, not root
/usr/bin/lpr	user lp, not root
/usr/bin/lprm	user lp, not root
/usr/local/bin/sudo	ok
/usr/local/bin/procmail	maybe, but big and ugly

setuid-root list: only programs I use

/usr/local/sbin/suexec	ok
/usr/sbin/traceroute	no. how about raw network access through
/usr/sbin/traceroute6	no. how about raw network access through
/usr/X11R6/bin/xterm	why?
/sbin/mksnap_ffs	why root? Why not sys call? Not sure.
/sbin/ping	no. how about raw network access through
/sbin/ping6	no. how about raw network access through

Final setuid-root list

/usr/bin/passwd

/usr/bin/su

/usr/bin/crontab

/usr/local/bin/sudo

/usr/local/bin/procmail

/usr/local/sbin/suexec

/usr/X11R6/bin/xterm

/sbin/mksnap_ffs

A Measure of network security

```
netmapper:~$  netstat -a
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign
tcp4       0      0  *.ssh                  *.*
tcp46      0      0  *.ssh                  *.*
```

Win ME

```
Proto  Local Address                Foreign Address
  TCP    127.0.0.1:1032               0.0.0.0:0
  TCP    223.223.223.10:139           0.0.0.0:0
  UDP    0.0.0.0:1025                 *:*
  UDP    0.0.0.0:1026                 *:*
  UDP    223.223.223.10:137           *:*
  UDP    223.223.223.10:138           *:*
```

Win 2K

```
Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1029           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1036           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1078           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1080           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1086           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:6515           0.0.0.0:0              LISTENING
  TCP    127.0.0.1:139          0.0.0.0:0              LISTENING
  UDP    0.0.0.0:445            *:*
  UDP    0.0.0.0:1038           *:*
  UDP    0.0.0.0:6514           *:*
  UDP    0.0.0.0:6515           *:*
  UDP    127.0.0.1:1108         *:*
  UDP    223.223.223.96:500     *:*
  UDP    223.223.223.96:4500    *:*
```

Win XP SP1

```
Proto  Local Address          Foreign Address        State
  TCP    ches-pc:epmap          ches-pc:0              LISTENING
  TCP    ches-pc:microsoft-ds   ches-pc:0              LISTENING
  TCP    ches-pc:1025           ches-pc:0              LISTENING
  TCP    ches-pc:1036           ches-pc:0              LISTENING
  TCP    ches-pc:3115           ches-pc:0              LISTENING
  TCP    ches-pc:3118           ches-pc:0              LISTENING
  TCP    ches-pc:3470           ches-pc:0              LISTENING
  TCP    ches-pc:3477           ches-pc:0              LISTENING
  TCP    ches-pc:5000           ches-pc:0              LISTENING
  TCP    ches-pc:6515           ches-pc:0              LISTENING
  TCP    ches-pc:netbios-ssn    ches-pc:0              LISTENING
  TCP    ches-pc:3001           ches-pc:0              LISTENING
  TCP    ches-pc:3002           ches-pc:0              LISTENING
  TCP    ches-pc:3003           ches-pc:0              LISTENING
  TCP    ches-pc:5180           ches-pc:0              LISTENING
  UDP    ches-pc:microsoft-ds   *:*
  UDP    ches-pc:isakmp         *:*
  UDP    ches-pc:1027           *:*
  UDP    ches-pc:3008           *:*
  UDP    ches-pc:3473           *:*
```

Win XP SP2 (May 2006)

```
Proto  Local Address           Foreign Address        State
TCP    0.0.0.0:135             0.0.0.0:0              LISTENING
TCP    0.0.0.0:445             0.0.0.0:0              LISTENING
TCP    0.0.0.0:6515            0.0.0.0:0              LISTENING
TCP    0.0.0.0:1026            0.0.0.0:0              LISTENING
TCP    0.0.0.0:1036            toolbar.google.com:5222  ESTABLISHED
UDP    0.0.0.0:microsoft-ds    *:*
UDP    0.0.0.0:isakmp          *:*
UDP    0.0.0.0:1025            *:*
UDP    0.0.0.0:1058            *:*
UDP    0.0.0.0:4500            *:*
UDP    0.0.0.0:6514            *:*
UDP    0.0.0.0:6515            *:*
UDP    0.0.0.0:6516            *:*
UDP    0.0.0.0:ntp             *:*
UDP    0.0.0.0:1900            *:*
UDP    0.0.0.0:ntp             *:*
UDP    0.0.0.0:netbios-ns      *:*
UDP    0.0.0.0:netbios-dgm     *:*
UDP    0.0.0.0:1900            *:*
UDP    0.0.0.0:ntp             *:*
```

RedHat FC4 MythTV backend, May 2006

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address               Foreign Address             St
tcp        0      0 0.0.0.0:3306                0.0.0.0:*                   LI
tcp        0      0 0.0.0.0:6543                0.0.0.0:*                   LI
tcp        0      0 0.0.0.0:111                 0.0.0.0:*                   LI
tcp        0      0 0.0.0.0:6544                0.0.0.0:*                   LI
tcp        0      0 127.0.0.1:631               0.0.0.0:*                   LI
tcp        0      0 127.0.0.1:5335              0.0.0.0:*                   LI
tcp        0      0 0.0.0.0:41752               0.0.0.0:*                   LI
tcp        0      0 127.0.0.1:6010              0.0.0.0:*                   LI
tcp        0      0 :::80                       :::*                        LI
tcp        0      0 :::22                       :::*                        LI
tcp        0      0 ::1:6010                    :::*                        LI
tcp        0      0 ::ffff:223.223.223.77:22    ::ffff:223.223.223.30:56718 ES
udp        0      0 0.0.0.0:32768               0.0.0.0:*
udp        0      0 0.0.0.0:942                 0.0.0.0:*
udp        0      0 0.0.0.0:68                  0.0.0.0:*
udp        0      0 0.0.0.0:5353                0.0.0.0:*
udp        0      0 0.0.0.0:111                 0.0.0.0:*
udp        0      0 0.0.0.0:631                 0.0.0.0:*
udp        0      0 223.223.223.77:123          0.0.0.0:*
```

Jails and Sandboxes

chroot a good start, a long time ago

- never try to jail user root
- be sure to cd(1) into the chroot jail
- only controls file system access, not network, CPU, process, other access
- It’s a good belt-and-suspenders for dangerous network services, like apache, named, samba. But difficult to set these up.

FreeBSD jail

- a good attempt
- man page is daunting: “make world into the jail.” I don’t want to.
- limits other things, like network access. This is good.
- Not available on other *nixes, I believe.

Sandboxes: we need better ones

- Lots of research on these over the years
- The goal is difficult. System call control? Virtual machines? Modified libraries? ptrace? /proc?

Suggested sandbox checklist

- Must be available on all of *nix
- Must not require root
- Understandable config files, with several policy examples available for common, important programs
- The goal: support browsers, mail readers, network servers, safely, with clearly-configured containment exceptions.
- reasonably efficient. Ten percent overhead is fine.
- At the moment, systrace looks the closest. Why doesn’t FreeBSD have it?

Kernel complaints

The size is astounding

- This was a classic 1127 complaint
- Kernel bloat can be fought
- Ken and Norman removing old code.
- growing with drivers, which can't be helped
- growing with features, which are not always that useful, and are dangerous
- My problem with cp(1)!
- I don't like kernel modules much
- Pike's Law: the software guys will slow down the software faster than the hardware guys will speed up the hardware.
- BPF is getting pushed too far: but ring buffer in user space?

system calls

- Too many of them
- Plan 9 got this right
- Setuid Demystified[2]
- What else is wrong?
- Need a trip test for kernel calls. And libc, etc. etc.

My Complaints with NFS

- Implemented in networking code
- Implemented on unsafe (portmapper) networking
- Does not implement Unix file system semantics
- Stifled user-level file system research for a while

My Complaints with X

- It’s huge
- There is policy hidden in the implementation
- Ackward to call and program: Tk/Tcl, openGL, and others make this easier
- Hundreds of calls to X. (Plan 9 had fewer than 20.)

Some Linux beefs

- The MythTV saga
- Cutting edge (HDTV3000) or non-standard hardware (damn VIA chips) changes something easy into a battle.
- nvidia updates stopped working on linux fc4....kernel call changed
- IR remote control: some change to I2C support
- I really miss control-T

Documentation

- Documentation these days often doesn’t hack it
- Man pages are still a good idea. Not in html, not in info, etc.
- A good man page is hard to write
- Where is the cd(1) man page?

Useless man pages

```
NAME
       mktextfm - create a TFM file for a font
NAME
       mktextfm - create a TFM file for a font
DESCRIPTION
       This  manual page is not meant to be exhaustive.  The complete documen-
       tation for this version of TeX can be found in the info file or  manual
       Kpathsea: A library for path searching.
```

I wanted environment doc

```
$ strings /usr/local/bin/tex | grep TEX
TEXMFOUTPUT
TEXEDIT
Usage: tex [OPTION]... [TEXNAME[.tex]] [COMMANDS]
  Run TeX on TEXNAME, usually creating TEXNAME.dvi.
  Any remaining COMMANDS are processed as TeX input, after TEXNAME is read.
  If the first line of TEXNAME is %&FMT, and FMT is an existing .fmt file,
TEXSIZES
TEXFONTS
TEXMFINI
TEXBIB
TEXPKS
TEXMFCNF
TEXMFDBS
TEXFORMATS
```

This is a horror

```
Version: ImageMagick 6.2.5 03/23/06 Q16 http://www.imagemagick.org
Copyright: Copyright (C) 1999-2005 ImageMagick Studio LLC

Usage: mogrify [options ...] file [ [options ...] file ...]

Where options include:
  -affine matrix       affine transform matrix
  -annotate geometry text
                       annotate the image with text
  -antialias           remove pixel-aliasing
  -authenticate value  decrypt image with this password
  -background color    background color
  -bias value          add bias when convolving an image
  -black-threshold value
                       forces all pixels below the threshold into black
  -blue-primary point  chromaticity blue primary point
  -blur geometry       blur the image
  -border geometry     surround image with a border of color
  -bordercolor color   border color
...
```

Sample tools from Lumeta

Filter sample: scramble

- How do you unsort a text file?
- Why would you want to?

```
perl -e ’
        srand;
        $bnd = 10000000;

        while (<>)

          $r = int(rand($bnd))/$bnd;
          print "$r     $_";

’ $*            |
sort -n         |
sed ’s/^[^      ]*[     ]*//’
```

Choose 6 of 12

- Giving away six PCs, with 12 interested parties, and no one gets more than one.
- Don’t ask how it was actually done.

```
seq 12 | scramble | sed 6q | sort -n
```

ilookup

```
$ ilookup 209.123.16.98
204.178.16.6
204.178.16.6    dirty.research.bell-labs.com    209.123.16.98
65.198.68.193
65.198.68.193   zathras-people.corp.lumeta.com
4.68.97.129
4.68.97.129     ae-1-55.bbr1.NewYork1.Level3.net
216.239.48.190
216.239.48.190  (ns1.google.com)
```

fhin

```
NAME
       fhin - find host in network.

SYNOPSIS
       fhin [ -n ] [ -s ] [ -u ] [ -v ] [ -w ] cidrlist1 cidrlist2 ...
...


            $ cat >1918.cidr <<!EOF
            > 10.0.0.0/8   private
            > 172.16.0.0/12     private
            > 192.168.0.0/16    private
            > !EOF
```

fhin sample

```
EXAMPLE
       Label hosts that are in RFC 1918 private address space:
            $ cat >ip <<!EOF
            > 135.104.52.1 router
            > 192.168.1.1  router
            > 10.4.0.1
            > 192.169.0.1
            > !EOF
            $ fhin 1918.cidr <ip
            135.104.52.1   router
            192.168.1.1    router    private
            10.4.0.1  private
            192.169.0.1
```

fhin options, etc

- **-n** suppresses unmatched lines
- **-v** write only unmatched lines
- We have fnin(1) now.

Rethinking traceroute(8)

```
$ traceroute www.google.com
traceroute: Warning: www.google.com has multiple addresses; using 216.239.37.1
traceroute to www.l.google.com (216.239.37.104), 64 hops max, 40 byte packets
 1  zathras-people (65.198.68.193)  0.824 ms  0.476 ms  0.449 ms
 2  cinderblock-inside (65.198.68.5)  0.305 ms  0.290 ms  0.376 ms
 3  lumeta-gw.lumeta.com (65.198.68.1)  0.530 ms  0.442 ms  0.421 ms
 4  651.ATM1-0.GW2.NYC8.ALTER.NET (157.130.95.173)  15.293 ms  81.633 ms  15.8
 5  171.at-5-0-0.XR1.NYC8.ALTER.NET (152.63.18.154)  5.114 ms  5.202 ms  5.330
 6  0.so-2-2-0.XL1.NYC8.ALTER.NET (152.63.19.29)  88.653 ms  45.135 ms  67.626
 7  0.so-3-1-0.XL1.NYC4.ALTER.NET (152.63.1.50)  5.894 ms  6.093 ms  6.017 ms
 8  0.so-6-0-0.BR1.NYC4.ALTER.NET (152.63.21.77)  6.070 ms  5.912 ms  6.052 ms
 9  if-4-3.core1.NTO-NewYork.teleglobe.net (216.6.82.25)  6.287 ms  6.758 ms
10  if-6-0.mcore3.NJY-Newark.teleglobe.net (216.6.57.49)  7.025 ms  7.174 ms
11  if-13-0.core1.AEQ-Ashburn.teleglobe.net (216.6.57.42)  11.782 ms  11.863 m
12  ix-2-0.core1.AEQ-Ashburn.teleglobe.net (209.58.27.130)  34.924 ms  12.119
13  216.239.49.38 (216.239.49.38)  12.970 ms  12.890 ms  12.976 ms
14  66.249.95.126 (66.249.95.126)  12.105 ms  50.593 ms  12.102 ms
15  72.14.232.106 (72.14.232.106)  12.796 ms  12.599 ms  12.747 ms
16  216.239.37.104 (216.239.37.104)  12.213 ms  12.837 ms  12.361 ms
```

Traceroute

- valuable tool since the mid-1980s
- sends three packets per hop, once a second if no response
- contains IP addresses, inverse DNS lookups, round-trip information
- Has old batch-style header

Let’s make a filter: netio(8)

- Unix filter: one line of text to stdin describes a packet to send
- Received packets each generate one line to stdout
- For now, a simple language for a few simple packets
- Goal: a tool with more flexibility than traceroute(8), but certainly not as pretty.
- Runs as root, so you don’t have to.

Input

Three fields, white-space separated:

- A packet ID, an integer less than 16 bits long. Corresponding return packets will have matching ID field
- A numeric IP destination
- Packet type: “P” for ICMP echo request, or n, the TTL setting of a traceroute UDP port range probe.

Sample hand-typed session

```
1 216.239.37.104 P
1 216.239.37.104 pinged
2 216.239.37.104 40
2 216.239.37.104 exceeded
3 216.239.37.104 16
3 216.239.37.104 reached
4 216.239.37.104 1
4 65.198.68.193 died
```

tracert, Version 2

```
#!/bin/sh
#
prog=ntr1
usage="$prog <ip-addr>"
#
# demo traceroute to a numeric IP target

for i in ‘seq 40‘
do
        echo "$i $1 $i"
done            |
netio -l 2 2>/dev/null |
sort -n
```

tracert, Version 2

```
1 65.198.68.193 died
2 65.198.68.5 died
3 65.198.68.1 died
4 157.130.95.173 died
5 152.63.18.154 died
6 152.63.19.29 died
7 152.63.21.17 died
8 152.63.21.77 died
9 209.244.160.181 died
10 4.68.97.129 died
11 64.159.3.254 died
12 4.68.121.114 died
13 4.79.228.38 died
14 72.14.232.108 died
```

tracert, Version n

```
#!/bin/sh
#
#       tracert - a netio-based simple traceroute

for i in ‘seq 32‘
do
        echo "$i $1 $i"
done            |
netio -l 2 2>/dev/null  |
sort -n                 |
sed ’/exceeded/,$d’     |
sed ’   s/ died//
        s/ reached//’   |
awk ’print $2, $1, $3’        |
ilookup ‘grep nameserver /etc/resolv.conf | awk ’print $2’‘   |
sort -k2,2n     |
awk ’print $1 "        " $3’
```

tracert, Version n

```
$ tracert 64.233.161.104
65.198.68.193   zathras-people.corp.lumeta.com
65.198.68.5     cinderblock-inside.corp.lumeta.com
65.198.68.1     lumeta-gw.lumeta.com
157.130.95.173  651.ATM1-0.GW2.NYC8.ALTER.NET
152.63.18.154   171.at-5-0-0.XR1.NYC8.ALTER.NET
152.63.19.29    0.so-2-2-0.XL1.NYC8.ALTER.NET
152.63.21.17    0.so-0-3-0.XL1.NYC4.ALTER.NET
152.63.21.77    0.so-6-0-0.BR1.NYC4.ALTER.NET
209.244.160.181 so-8-1.car1.NewYork1.Level3.net
4.68.97.129     ae-1-55.bbr1.NewYork1.Level3.net
4.68.128.206    as-3-0.bbr2.Washington1.Level3.net
4.68.121.146    ae-11-55.car1.Washington1.Level3.net
4.79.228.38     GOOGLE-INC.car1.Level3.net
72.14.232.108   65.246.245.2
72.14.232.99    65.246.245.2
72.14.232.103   65.246.245.2
216.239.48.190  (ns1.google.com)
64.233.161.104  (ns1.google.com)
```

tracert improvements and comments

- Look up alphabetic targets
- Send several packets instead of one
- Multiprotocol probes: different netio field three gives different packet types
- add hop number to output

cscan

```
#!/bin/sh

TMP=/var/tmp/cscan

for i in 1 2
do
        for j in 0 ‘seq 255‘
        do
                echo "$j $1.$j P"
        done
done |
netio -l 2      | tee $TMP |
awk ’print $2’        |
sort -u -t. -n -k4

grep -v pinged $TMP 1>&2
rm -f $TMP
```

cscan

```
cscan 209.123.16 | pr -4 -t
209.123.16.1   209.123.16.21     209.123.16.41     209.123.16.108
209.123.16.2   209.123.16.25     209.123.16.42     209.123.16.109
209.123.16.3   209.123.16.26     209.123.16.43     209.123.16.110
209.123.16.4   209.123.16.32     209.123.16.44     209.123.16.113
209.123.16.6   209.123.16.33     209.123.16.45     209.123.16.114
209.123.16.7   209.123.16.34     209.123.16.50     209.123.16.117
209.123.16.8   209.123.16.36     209.123.16.65     209.123.16.118
209.123.16.10   209.123.16.38     209.123.16.97     209.123.16.129
209.123.16.12   209.123.16.39     209.123.16.98     209.123.16.193
209.123.16.20   209.123.16.40     209.123.16.99
```

a better cscan

```
#!/bin/sh

TMP=/var/tmp/cscan

for i in 1 2
do
        for j in 0 ‘seq 255‘
        do
                echo "$j $1.$j P"
        done
done |
scramble |
netio -l 2      | tee $TMP |
awk ’print $2’        |
sort -u -t. -n -k4

grep -v pinged $TMP 1>&2
rm -f $TMP
```

netio(8) improvements

- Second field can encode loose source routing, various tunnels, spoofed source addresses
- Third field can encode other packets: UDP, SNMP, DNS queries, etc.
- netio -x gives extended return information

```
packet-id [tunnel-path:[I]]dest-path[/spoof-ip] packet-type

ttl-returned ttl-remote target-ip [round-trip-time]
```

netio(8) problems

- input packets .ne. output packets: loss of power?
- rate limiting only works per copy of netio
- what about fancier packets? Does the idea still work?

netio in a product

- Lumeta has now grown to > 60 people, with core software running under a GUI in IPsonar.
- It has proved versatile, amenable to change and testing, and robust
- The same tools were used to generate html reports
- The report code is pretty ugly, but we have made three versions, without too much trouble.

```
echo "IP addresses:  ‘wc -l <tmp/iplist | commas‘"  >>index.html
```

Conclusion

Prosper

- Though I have used LaTeX extensively (two books), I spent more time wrestling the tool than making slides
- slide count was nice
- easy to import Unix examples and similar text
- Decent monospaced type is lacking
- I really missed WYSIWIG slide sorter
- I had to run Windows anyway: couldn't get rid of the bookmarks display on xpdf(1)
- Markup languages are worth the pain when gorgeous output is needed. Papers written in Word have noticably poor typography.
- I wanted to use powerdot, but it didn't Just Work.

I joined the Labs because I liked their clean philospohy. I left because I used it effectively.

Epic Battles

- It once took me two days to fix a COMPASS macro
- Prosper has evolved into an epic battle
- My experience with Linux plus MythTV was certainly an Epic battle
- Epic battles can be caused by poor documentation. Even the source code didn’t help me on prosper.
- An epic battle is not a victory for usability.

There is a lot of work to do

- “The job is not done until you do the paperwork” — Cliff Stoll
- Take the time to write a man page, and do it really well.
- I’d like to see apropos work as well as the old permuted index.
- test suites for system calls. Might be a job for VM.
- Our security is not so hot, and we can do much better.

User interfaces are hard to do

- Be humble, but bold. Try some conceptual integrity.
- Read Raskin, Norman, Tufte and others
- It’s not usable until it passes usability tests

Unix is not for everyone

- There is a lot of whining about Unix[4]
- Over-arching consistency makes something that is easier to use, but that tool certainly doesn't solve all problems
- Unix is not for everyone. I think of it as a professional programmer's workbench. People can use tables and chairs without building their own.

“With a teletype interface and the Fortran language, the computer will be easy to use”

Usability and System Administration

- We don’t know how to do system administration well.
- Software has only progressed a little since the 1970s
- The only computer I use with very few system administration problems is the Treo 650.
- The problem is largely unsolved for general purpose computers
- I’d like to see a lot more experimentation in the area. I think it calls for Brook’s “conceptual integrity”.
- Experimentation needs to be cheap, so we can try lots of things

*

References

[1] AHO, A. V., KERNIGHAN, B. W., AND WEINBERGER, P. J. *The awk programming language*. Addison-Wesley series in Computer Science. Addison Wesley, 1988.

[2] CHEN, H., WAGNER, D. A., AND DEAN, D. Setuid demystified. In *Proceedings of the of the Eleventh USENIX Unix Security Symposium* (San Francisco, CA, 2002).

[3] FREDERICK P. BROOKS, J. *The Mythical Man-Month (anniversary ed.)*. Addison-Wesley Longman Publishing Co., Inc., 1995.

[4] GARFINKEL, S., WEISE, D., AND STRASSMANN, S. *The UNIX-Haters Handbook*. IDG Books World, 1994.

[5] KERNIGHAN, B. W., AND PIKE. *The UNIX Programming Environment*. Software Series. Prentice Hall, 1984.

[6] MCILROY, M. D. Unix on my mind. In *Proc. Virginia Computer Users Conference* (Blacksburg, September 1991), vol. 21, Virginia Tech, pp. 1–6.

[7] SPENCER, H., AND COLLYER, G. #ifdef considered harmful or portability experience with C news. pp. 185–198.

[8] WIRTH, N. PL360, A programming language for the 360 computers. *Journal of the ACM 15*, 1 (Jan. 1968), 37–74. See also [**?**].