

Plan9: Unix aber diesmal richtig

Matthias Bauer
bauerm@weggla.franken.de

Nov 26 2006

Übersicht

- 1 Warum ein Unix Nachfolger?
- 2 Plan9

Unix – Geschichte

- Start ca 1969 bei Bell Labs, als Mini-Projekt der Multics Entwickler
- Sollte im Gegensatz zu Multics einfach und übersichtlich sein
- Frühe Entwickler waren Thompson, Ritchie, Kernighan u.a.
- Kleine Programme, die jeweils eine Sache gut können, und die man hintereinander schalten kann (pipes).

```
ps uaxw | awk '{print $4," ", $11}' | sort  
-n | sed -n '$p' | awk '{print $2}'
```

Findet den Prozess, der am meisten Speicher frisst.

Unix – Geschichte (cont.)

- Alles ist ein File (auch Geräte `/dev/tty00`, `/dev/hda0`, `/dev/audio`)
- Kein Unterschied zwischen ausführbaren Programmen und Daten, z.B. `strings` funktioniert überall
- Menschenlesbare Konfigfiles (ASCII ohne XML)

Unix – Geschichte und Auswirkung

- Immenser Einfluß auf die Entwicklung von Betriebssystemen
- „Who doesn't know UNIX has to re-invent it – poorly”
- Kaum ein System danach ohne deutlichen Einfluß (Dateisystem, Sockets, . . .)
- Standardisiert, leider zu oft (X/Open, POSIX, SysV, . . .)
- Von der Mainframe zur vernetzten Workstation: Berkeley erfindet Sockets und das Netz

Aber — Alles ist ein File? Aecht?

Die Unix Philosophie „Alles ist ein File“ ist leider nicht ganz durchgängig

- Netzwerk Interfaces (`ioctl`)
- Sockets (`connect`, `accept`, `listen`, `setsockopt`)
- Routen (`socket(PF_SOCKET, ...)` mit vielen magischen `#defines`)
- Prozesse: `chown -R nobody.nobody /proc/1234 ?`
- File Deskriptoren
- `ls -l /dev/hda1` liefert 0 Bytes??? Die Partition ist 50Gb gross!
- Wenn alles ein File ist, warum kann man dann `/dev/audio` nicht per NFS exportieren??

Anachronismen

Das Standard IO Konzept hängt immer noch an Hardware der frühen 70er:

`ttys/ptys` **sind Fernschreiber**

- Warum hat mein `xterm` eine Baudrate?
- Warum brauch ich einen Editor oder wirre History-Substitution um an Zeugs zu editieren, was eh schon auf dem Bildschirm steht??
- Wirre Effekte mit 7bit-unclean Anwendungen

Und so weiter

- Graphische Oberfläche X11 ist alles andere als light-weight
- Sicherheit
 - root darf alles
 - man muss root sein, um diverse wichtige Sachen machen zu können, z.B. raw sockets (ping), Listen der descriptoren pro Prozess (lsof), ports unter 1024.
 - man muss root sein, um Sachen in den Dateibaum zu hängen, z.B. Crypto-Filesysteme, CDs, NFS, etc, auch wenn die pro user sein sollten
 - man muss root sein, um chuid machen zu können, also kann ein Normaluser seine sub-Prozesse nicht einschränken.
 - X11 Server läuft als root
 - vieles läuft als root, oder mit suid Bit

Plan9 from Outer Space



Plan9 from Bell Labs – Geschichte

(„Plan9 from Outer Space“ ist ein B-Movie von Ed Wood)

- Bei Bell Labs in den 80ern begonnen (Thompson, Pike, Forsythe, u.a.)
- Release 1 1993, Release 2 1995
- Seit 2000 Open Source
- Release 4 2003
- Seit ein paar Jahren wieder aktive Entwicklung

Plan9 – Features

- Ziel: Arbeiten im Netz, mit zentralen Diensten
- 100% neuer Code, keine Problem mit AT&T, SCO, ...
- eigener Compiler, eigene Shell (`rc`), eigene libc (`libu`)
- Kompiliert und läuft auf 8 Architekturen
- Graphische Oberfläche ist mitgeliefert
- keine POSIX API
- Namensräume pro Prozessgruppe
- Kernel stellt Adressraumverwaltung, Zugriff auf File Server und bietet selbst ein paar an

Alles ist eine Datei

File Server gibts z.B. für:

- Files auf der Festplatte
- Files im Backup
- FTP Server
- Crypto FS
- Devices `ls /dev/disks/` zeigt sinnvoll benannte Partitionen mit ihrer Grösse
- Files in tar/zip/etc Archiven
- Prozesse (mit Permissions!)
Signale per `echo 'text' > /proc/5/note`, d.h. nur ein signal Handler, der Text liest

Alles ist eine Datei – cont.

File Server gibts auch für:

- GUI

```
cp /mnt/wsys/window $home/tmp/ (Screenshot von diesem Fenster)
```

```
grep `top.c` /mnt/wsys/wsys/*/text (suche top.c im Inhalt/Scrollback aller Fenster)
```

- GPS Empfänger

- Files/Songs auf CDs, schreibt auch:

```
cp /tmp/songs/* /mnt/cd/wa  
rm /mnt/cd/wa
```

Alles ist eine Datei – cont.

Auch Netzdaten sind Dateien

- `cat /net/arp` → bekannte Adressen
- `cat /net/ndb` → Adresse, DNS server, Router, ...
- Netcat in a Shellskript:

```
num='{cat < /net/tcp/clone}  
echo connect tcp!$1!$2 > /net/tcp/$num/ctl  
cat /net/tcp/$num/data
```
- Statt mühevolem Umschreiben/Linken auf SOCKS:

```
socksfs tcp!socksgateway!9050  
bind -ac /mnt/socks /net/
```

Alles ist eine Datei – cont.

File Server Protokoll

- Zugriff über ein einziges Protokoll, 9P2000
- Auch lokale Zugriffe!
- Alle FS ausser dem Crypto FS können ex/importiert werden
- Z.B. Remote Login durch Importieren von `/dev/cmd` und Exportieren von `/dev/screen`, `/dev/mouse`, `/dev/kbd`
- Remote Debugging durch Importieren von `/proc`

Plan9 – Namensräume

Jeder Prozess kann sich seinen Dateibaum selber basteln

- `mount` lädt Bäume von File Servern rein
- `bind` legt Dirs/Files unter/über ein Verzeichnis (Union Mount)
- `ns` zeigt die Kommandos, die zum momentanen Namensraum geführt haben
- Prozesse können beim `rforke`n angeben, ob Kindprozess das gleiche sieht, oder noch was ändern kann.

Plan9 – Fossil & Venti

Das FS für dauerhaftes Speichern ist Fossil, Blockspeicher dahinter Venti

- Blöcke erstmal auf lokalem, schnellem Speicher
- Jeder Block kann auf einen Blockspeicher (lokal oder im Netz) gelegt werden
- Völlig transparent, einfach con `fscons`
`>>> snap -a ESC-\q`
- Damit hat man eine Zeitmaschine!
`/archive/yyyy/mm/dd/path/file`

Plan9 – Sicherheit

- Kein root. Lokaler Konsolen-Account ist lokal allmächtig.
- Jeder kann nach none UID wechseln
- Prozesse können Kindprozesse stark einschränken
- Eine Art ssh-agent names factotum für alle Authentifikationen
- Auch die Serverseite benutzt einen Agent → keine Geheimnisse in Konfigfiles
- amount: authenticated mount
- auth/authsrv eine Art Kerberos

Plan9 – Auswirkungen

Plan9 Konzepte, die woanders aufgenommen wurden:

- /proc : Linux, FreeBSD, Solaris
- rfork: OpenBSD Threads
- UTF8 : ISO, ...
- 9P2000 : Linux ab 2.6.16

Plan9 – Erfahrungen

- Extrem steile Lernkurve
- Sehr viel über gute alte UNIX Tools gelernt (sed, pic, troff, ...)
- Coole Ideen drin (venti, Netz als Files, ...)
- Immer noch am Lernen
- Wenig Doku, viel Hilfe von den 9fans
- 9.intro.pdf von Francisco J. Ballesteros (Uni Madrid) ist unentbehrlich
- Plan9 Wiki absolut nötig, um irgendetwas machen zu können
- Manches ist erstaunlich einfach, wenn man sich mal an die Denke gewöhnt hat
- Manches aber nicht. Horrorbeispiel: Einrichten einer neuen fossil-partition.

Zum Ausprobieren

- Live CD
- Läuft in VMware und Qemu
- Installation ist *relativ* einfach
- p9p, `plan9.us` Plan9 from userspace liefert einen Teil der Leistung für UNIX Umgebungen

The End

Links:

- Bell Labs Plan9
<http://www.cs.bell-labs.com/plan9/index.html>
- Nemos Vorlesungs Skript
<http://plan9.escet.urj.es/who/nemo/9.intro.pdf>
- Kurzanleitung
<http://www.magma.com.ni/moin/Plan9Tutorial>