Plan 9 Is (Not) For You

```
From: kalona.ayeliski@fastmail.us
To: 9fans <9fans@9fans.net>
Subject: Re: [9fans] Where can I find active Plan 9 communities for support
 and collaboration?
Date: Sun, 4 Aug 2024 14:27:58 -0400
Reply-To: 9fans <9fans@9fans.net>

From a newcomer’s perspective, it feels like dealing with a cult run
by scam artists.  It seems someone wants to profit from me by selling
books on Amazon, like a multi-level marketing group.  People say
others here are on a spectrum, but it feels more like psychosis, with
a loss of contact with reality.  I really feel like I’m being
gaslighted.  I might seem like a troll, but you don’t understand how
you appear to others.

I am looking for a Plan 9 group that doesn’t behave this way.  If
anyone is interested, let’s form a group that isn’t cult-like, that
just wants to help newcomers and not prey on them.
```

About Me

- “Current” system administrator focusing on Linux/UNIX administration
- “Former” CNSA student
- Full time Linux user since 2012 (currently NixOS)
- On and off Plan 9 user since 2014

About You

- This page intentionally left blank

What is Plan 9?

- This page intentionally left blank

What is Unix?

- Closed-source operating system developed in the late 60’s/early 70’s by Bell Labs
- Design philosophy includes concepts such as:
  - Everything is a file
    - Physical devices (such as disks or modems) are all represented as files in /dev and can be interacted with as if they were normal files
    - Other virtual file systems exist in newer Unix-likes:
      - /sys: inspect and modify kernel behavior
      - /proc: inspect and interact with running processes
    - Allows unified method for accessing system resources, whether via compiled code or via an interactive shell
  - Tooling is modular; each thing does one thing, and does it well
    - Shell utilities like grep, sed, tr, cat, etc. perform one type of operation and behave intuitively, and can be chained together to support complex operations

What are Unix-likes?

- Operating systems either inspired by Unix or forked directly from Unix code
- Can be closed source or open source
- Examples:
  - Linux: Open source, inspired by Unix
  - FreeBSD, Open source, predecessor BSD was forked from Unix
  - AIX: Closed source, forked from Unix

Limitations of Unix and Unix-likes

- Not everything is a file
  - Networking
    - Compiled programs must invoke the socket(2) and connect(2) syscalls to create a socket on which data can be sent/received over the network
    - An interactive shell cannot send/receive data over the network without utilizing a utility like netcat or /dev/tcp (DO NOT RESEARCH)
    - IP configuration is done via syscalls or ip/ifconfig
  - Graphics
    - This line intentionally left blank
  - In many of these cases, the first Unix releases did not support these features, so support was later tacked on rather than designed from within the operating system

Limitations of Unix and Unix-likes (cont.)

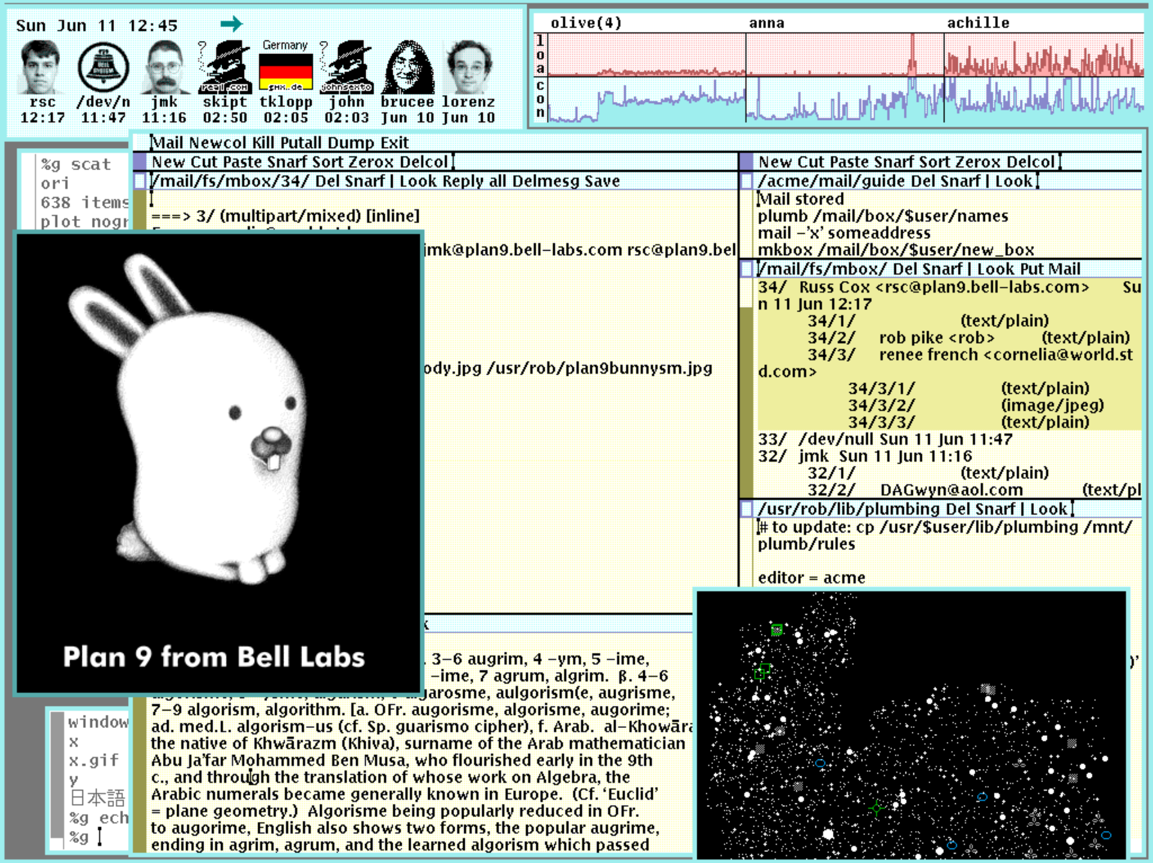
- Processes are privileged by default
  - Full access to the file system tree, when not restricted by file permissions
    - This grants access to devices in /dev, process lists in /proc, etc.
  - Access to syscalls that grant privileges like networking access
  - Breaks the principle of least privilege
- Some Unix-like tools and features “work backwards” to subtract privileges from specific processes:
  - chroot (set root directory of a process to a specific directory)
  - FreeBSD jails (chroot on steroids)
  - OpenBSD pledge(2) and unveil(2)
  - Linux cgroups and namespaces (restrict a processes’ access to various resources)

Limitations of Unix and Unix-likes (cont.)

- Clustering is hard
  - Unix systems are not designed to work in tandem, so building highly-available clustered systems typically involves writing an entire application-level stack anew (Kubernetes, etc) rather than utilizing low-level operating system capabilities
- Many tools no longer do one thing
  - Scope creep in certain utilities have made duties ambiguous
    - cat does much more than concatenate files
    - emacs is a fully fledged operating system
    - New features means new exploits: https://gtfobins.github.io/

What is Plan 9?

- Plan 9 is a research operating system created in the 90’s by Bell Labs, including people who worked on the original Unix
- Builds upon Unix concepts and takes them to the “next level” in a world where networking and graphics are ubiquitous



```
Mail Newcol Kill Putall Dump Exit
New Cut Paste Snarf Sort Zerox Delcol
/mail/fs/mbox/34/ Del Snarf | Look Reply all Delmesg Save

===> 3/ (multipart/mixed) [inline]
jmk@plan9.bell-labs.com rsc@plan9.bel
ody.jpg /usr/rob/plan9bunnysm.jpg
```

```
New Cut Paste Snarf Sort Zerox Delcol
/acme/mail/guide Del Snarf | Look
Mail stored
plumb /mail/box/$user/names
mail -'x' someaddress
mkbox /mail/box/$user/new_box
/mail/fs/mbox/ Del Snarf | Look Put Mail
34/  Russ Cox <rsc@plan9.bell-labs.com>       Su
n 11 Jun 12:17
        34/1/                 (text/plain)
        34/2/    rob pike <rob>       (text/plain)
        34/3/    renee french <cornelia@world.st
d.com>
              34/3/1/              (text/plain)
              34/3/2/              (image/jpeg)
              34/3/3/              (text/plain)
33/  /dev/null Sun 11 Jun 11:47
32/  jmk  Sun 11 Jun 11:16
        32/1/                 (text/plain)
        32/2/    DAGwyn@aol.com            (text/pl
/usr/rob/lib/plumbing Del Snarf | Look
# to update: cp /usr/$user/lib/plumbing /mnt/
plumb/rules

editor = acme
```

```
%g scat
ori
638 items
plot nogr
```

```
. 3–6 augrim, 4 -ym, 5 -ime,
-ime, 7 agrum, algrim.  β. 4–6
arosme, aulgorism(e, augrisme,
7–9 algorism, algorithm. [a. OFr. augorisme, algorisme, augorime;
ad. med.L. algorism-us (cf. Sp. guarismo cipher), f. Arab.  al-Khowāra
the native of Khwārazm (Khiva), surname of the Arab mathematician
Abu Ja'far Mohammed Ben Musa, who flourished early in the 9th
c., and through the translation of whose work on Algebra, the
Arabic numerals became generally known in Europe.  (Cf. ‘Euclid’
= plane geometry.)  Algorisme being popularly reduced in OFr.
to augorime, English also shows two forms, the popular augrime,
ending in agrim, agrum, and the learned algorism which passed
```

```
window
x
x.gif
y
日本語
%g ech
%g
```

How Plan 9 Does Things Better™: Filesystems

- More things are files (everything is a file *system*)
  - Networking stack is represented as files in /net
    - To initiate a new tcp connection:
      - Read /net/tcp/clone for a new connection ID
      - Write a command to the control file for the connection ID:
        - echo connect 142.11.219.191!80 > /net/tcp/X/ctl
      - Send and receive data by reading and writing to /net/tcp/X/data
    - Also see /net/udp, /net/tls, /net/icmp, each their own filesystem mounted to /net
  - Graphical devices are represented as files in /dev
    - /dev/screen is your current display: to take a screenshot, copy it and convert it to a png
  - Audio devices are represented as files; to play audio, write to /dev/audio
  - All of these devices are provided by specific filesystems

How Plan 9 Does Things Better™: Namespaces

- The concept of global mounts does not exist; instead, each process has its own namespace
- A process’s namespace builds its view of the file system
  - “mount” attaches a filesystem to the current namespace at a specific location
  - “bind” mounts a directory to another, optionally combining the contents of directories
    - In Plan 9, /dev consists of multiple kernel device file systems bound together in your namespace
    - /bin consists of multiple executable directories bound together: thus, your $path only contains two directories: /bin and .
  - “unmount” (sic) removes a mount or bind from the namespace
- Each process can thus build its own file system based on its requirements, fulfilling the principal of least privilege
- Processes without audio filesystems bound to /dev will not be able to play audio, and a process without /net filesystems will not be able to send network packets

How Plan 9 Does Things Better™: 9p Protocol

- The 9p protocol is the backing protocol for all file-based inter-process communication in Plan 9, local and remote
- 9p is network transparent, meaning that all parts of the system can be replaced with components from remote Plan 9 computers; you can, for example:
  - Mount a remote audio filesystem to /dev to play audio to a remote computer
  - Mount your local /dev/draw (monitor) and /dev/kbd (keyboard) to a remote computer to achieve remote desktop capabilities
  - Mount a remote computer’s /net/ether0 (ethernet device) to your local computer, and send traffic over it to implement a VPN
- In each of these examples, because only a single processes’ namespace is modified, only that process utilizes remote resources: the rest of the system continues functioning as normal

How Plan 9 Does Things Better™: Clustering

- Plan 9 is a distributed operating system
  - Through the use of the 9p protocol, distributed computing with Plan 9 is simple, as the entire system is network transparent
  - The idealized Plan 9 lab consists of a number of servers, each performing their own functions:
    - Terminals: thin clients with no local storage, only a mouse, keyboard, and lightweight CPU + RAM, used to access a Plan 9 cluster
    - CPU servers: servers capable of performing intensive CPU operations
    - File servers: servers dedicated to storing and serving files
    - Auth servers: servers dedicated to performing authentication to other servers
  - This may sound similar to modern virtualization setups, but the key is that the clustering happens at the *operating system level*, not the application level

State of Plan 9 today

- Plan 9 failed to gain a significant market share, as Unix-likes were already "good enough"
- Development slowed in the late 1990’s
- Fourth edition was released under a custom open source license in 2002
  - This enabled the start of many different forks of Plan 9, such as 9front, 9legacy, 9atom, Harvey OS, JehanneOS…
- All editions were released under the GPL in 2015
- In 2021, copyright was transferred from Bell Labs to the community-ran Plan 9 Foundation, after which all releases were relicensed to the permissive MIT license

What is 9front?

- Fork of Plan 9 created in the late 2000’s
- Developed by cat-v, who define themselves as a “Random Contrarian Insurgent Organization”
- More or less the de facto fork for most Plan 9 users today
- https://fqa.9front.org/fqa.html

With Un
SPEED LIMIT 35
EVERYTHING HAPPENS FOR A REASON. SOMETIMES THE REASON IS YOU'RE STUPID AND MAKE BAD DECISIONS
9front System

9front fixes and improvements to Plan 9

- All around better hardware support, specifically aimed at consumer laptops
- Improved file systems (cwfs, hjfs)
- git support
- USB drive booting
- Modern wifi support (wpa/wpa2)
- Various Nintendo emulators (gba, nes, etc)
- SSH clients
- Modern(?) web browsers (mothra, netsurf/nsport)
- An expanded /lib, including:
  - Nietzsche quotes
  - The communist manifesto
  - Insults from OpenBSD’s Theo De Raadt

Who is Cirno?

- Official 9front mascot
- From Touhou
- Associated with the number 9
- Known to be the strongest

Using 9front

- Demo time!
  - Installing 9front
  - Inspecting the filesystem and our namespace
  - Using Plan 9 tools
  - Updating
  - Kernel hacking
  - ???

Further Reading/References

- https://9p.io/sys/doc/
- https://fqa.9front.org/
- https://wiki.9front.org/unix2plan9
- https://doc.cat-v.org/plan_9/
- https://9fans.topicbox.com/groups/9fans
- https://github.com/henesy/awesome-plan9
- https://drewdevault.com/2022/11/12/In-praise-of-Plan-9.html
- https://web.mit.edu/~simsong/www/ugh.pdf