Distributed, Persistent, Fire is Inferno

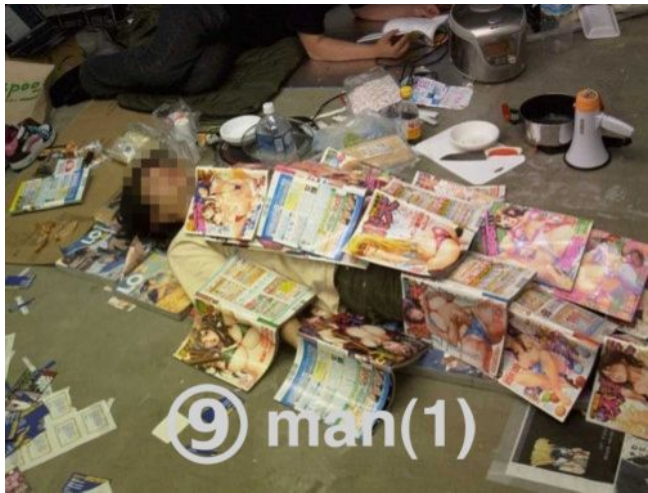
from the people’s front

Updates

- Thread for first program you ever wrote on Pizziaza
- ISU Cyber Defense Competition 2
  - March 29th sign up opens
- The Public Grid is up!
  - http://wiki.9gridchan.org/public_grid/index.html
- Public Plan 9 vps available
  - Email seh

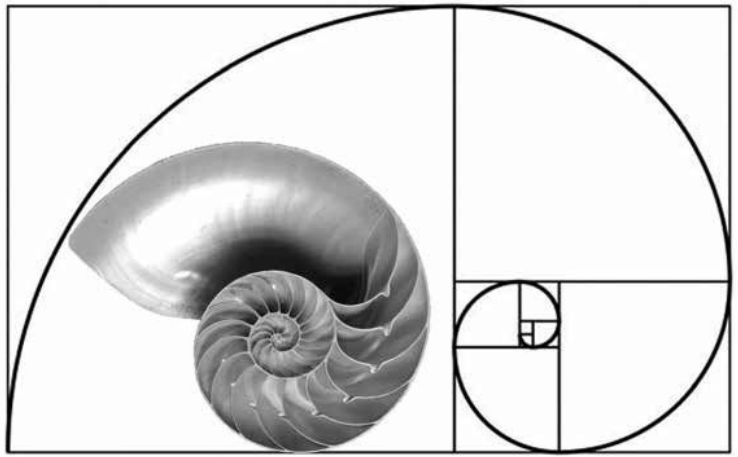
Some terms and keywords

- cat(1) — refers to the section 1 manual for cat, read via `man 1 cat`
- 9p — filesystem protocol, used in docker, kvm, etc.
- Styx — also known as the 9p2000 version of the 9p protocol
- Namespace — similar to scope, but for OS state
- Dialstring — `proto!host!service`, service can be a port
- CSP — a book, *Communicating Sequential Processes* by Hoare
- Inferno — an operating system
- Limbo — a programming language
- Dis — a virtual machine



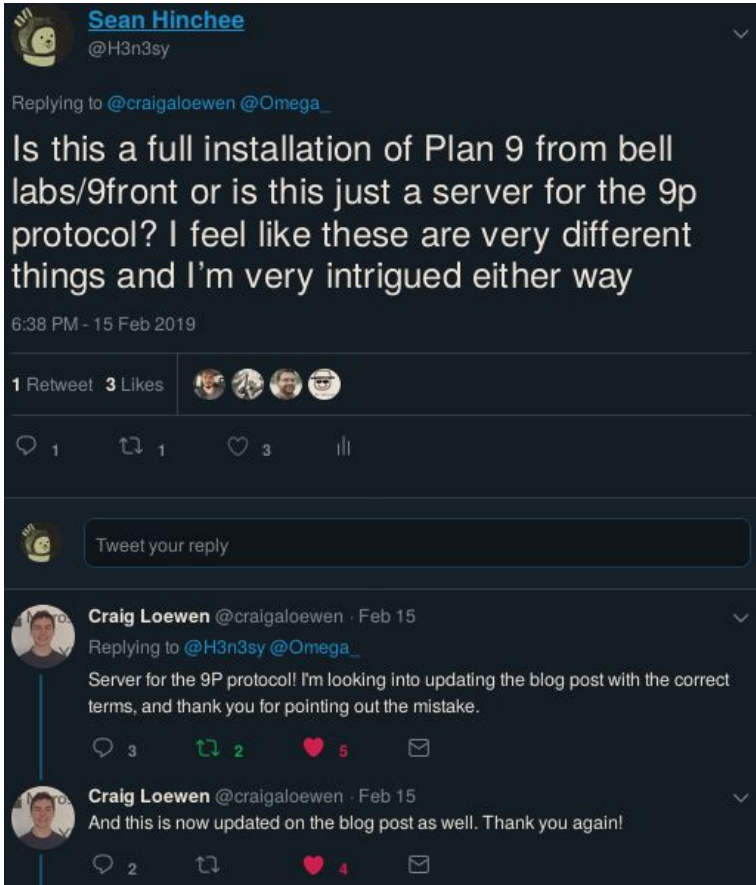
Shell prompts

- Linux
  - $
- BSD
  - #
- Plan 9
  - %
- Inferno
  - ;



⑨times

- Did you know Microsoft adopted 9p2000.l recently?
- WSL update post originally said “a Plan 9 server”
- Lol
- https://blogs.msdn.microsoft.com/commandline/2019/02/15/whats-new-for-wsl-in-windows-10-version-1903/



What is grid computing?

- Multiple, interoperating systems
- Can be heterogeneous
- Can be physically distant

From Microsoft:

Grid computing is a group of networked computers that work together as a virtual supercomputer to perform large tasks, such as analyzing huge sets of data or weather modeling.

What is distributed computing?

- Nuanced difference from grid computing
- All distributed computing is grid computing
- Not all grid computing is distributed

From *Distributed Systems-Principles and Paradigm*:

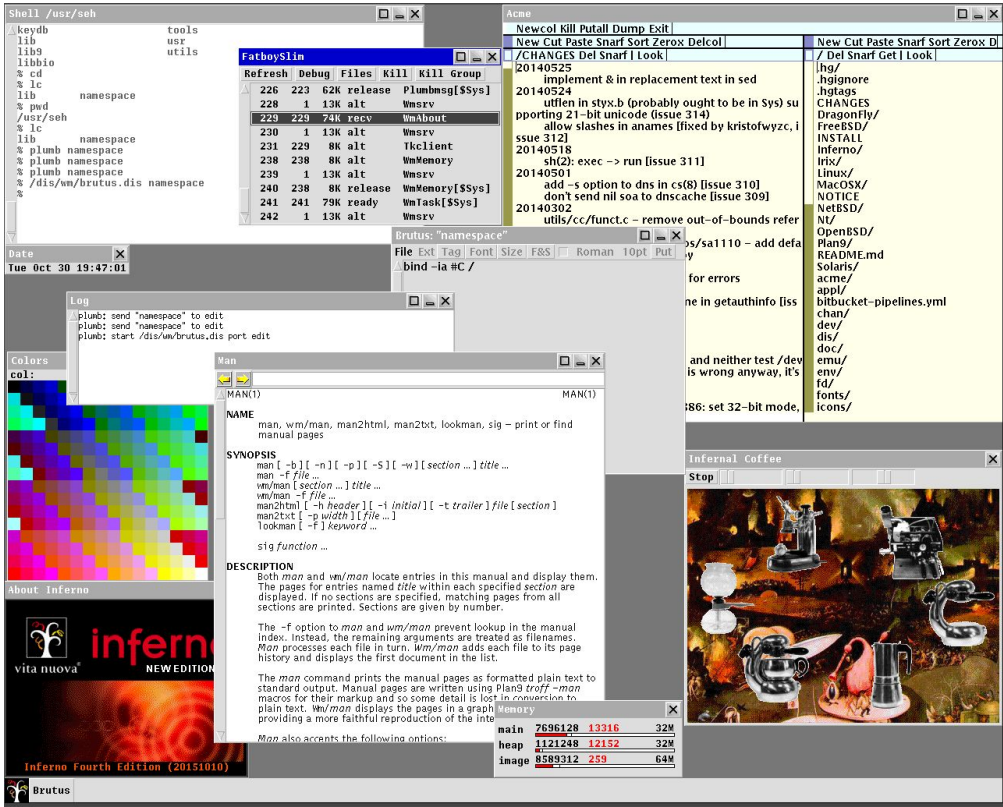
The phrase Distributed Computing can be defined as a Collection of independent computers that appear to its users as a Single Coherent system.

What are dynamic namespaces?

- Unix systems lack these
- Implemented in: Plan 9, Inferno, Akaros, Redox, and a few others
- Linux Ex:
  - `$ mount /dev/sdc2 /mnt/pr0n`
  - Every process can see your `pr0n/` with sufficient permissions
  - Permissions and ACL's determine who sees what
- Plan9 Ex
  - `% mount /srv/pr0n /mnt/pr0n`
  - Only this process and this process's children can see `pr0n/`
  - Permissions can also apply!

What is Inferno?

- Operating system from Bell Labs
  - Later Alcatel-Lucent
  - Even Later, Vitanuova Holdings LTD.
- Designed for embedded systems
- Alcatel-Lucent phones and servers
- Strong corporate backing
- Many closed-source resources
- Forked to Purgatorio by 9front



How to get?

- Try in the browser (thanks Pete): http://tryinferno.reverso.be/
- Using mercurial, the 9front fork
  - `% hg clone` http://code.9front.org/hg/purgatorio
- Using mercurial, the official tree
  - `% hg clone` https://bitbucket.org/inferno-os/inferno-os/

To install hg(1) on Debian derivatives:

```
$ sudo apt install mercurial
```

To install hg(1) on OpenBSD:

```
# doas pkg_add mercurial
```

How to use?

- Follow the instructions in INSTALL
- Run emu, read emu(1)
- Read the README
- Have fun!!

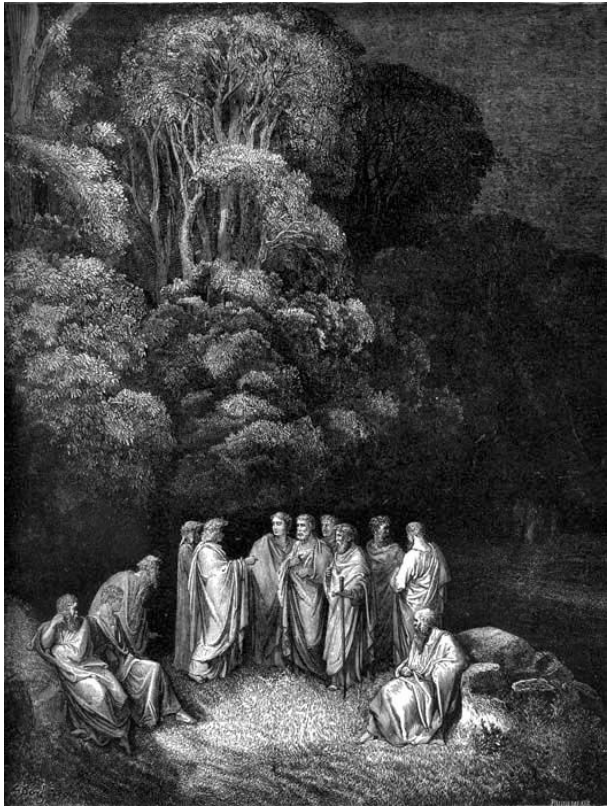
```
; pwd
/usr/seh
; lc
charon/   keyring/  module/   tmp/
dis/      lib/      namespace
; lc ..
inferno/ seh/     test/
; lc /dev
cons            hoststdin      msec           snarf
consctl         hoststdout     notquiterandom sysctl
cursor          jit            null           sysname
drivers         keyboard       pointer        time
hostowner       kprint         random         user
hoststderr      memory         scancode
;
```

What is Dis?

- Virtual machine
- Closer to the JVM than to KVM
- Inferno runs on top of a Dis VM
- Anything can target Dis, in theory

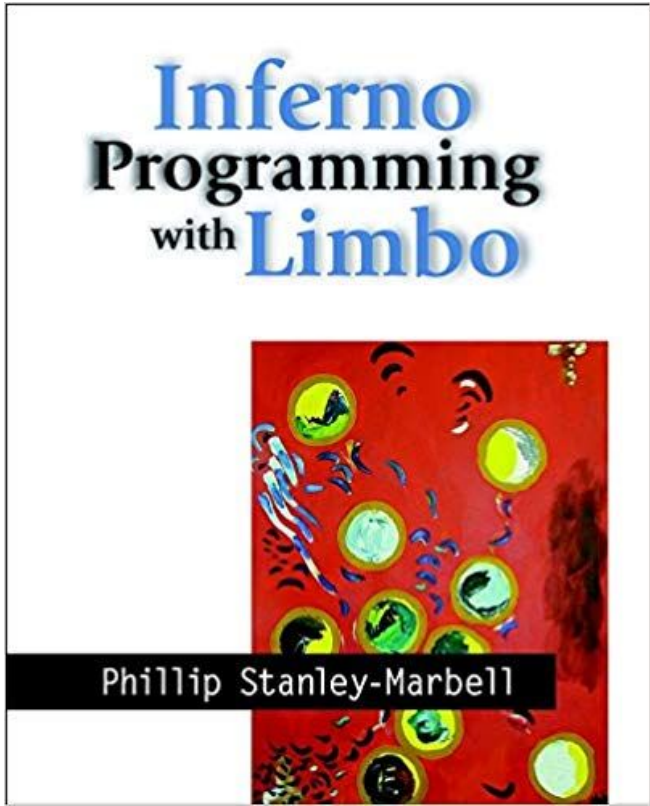
What is Limbo?

- Programming language
- The Inferno compilers output Dis bytecode
- A .dis file would be comparable to a Java .class file
- Dis bytecode is executed at runtime by the Dis VM
- Supports modules and dynamic loading of modules
- Uses CSP concepts heavily



History of Limbo

- C
  - Syntax
- Alef
  - Channels and Abstract Data Types (ADT)
- CSP
  - Channels alternating on Channels
- Newsqueak, ML
  - Module system
  - Compile-time types
  - Garbage collection



A bit of limbo

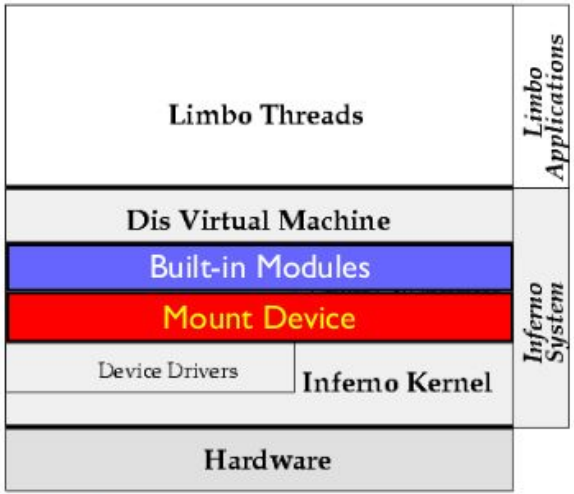
```
myvar := <-syncchan;
```

- Waits for a value from a channel, then writes it to a variable
- Sets type implicitly

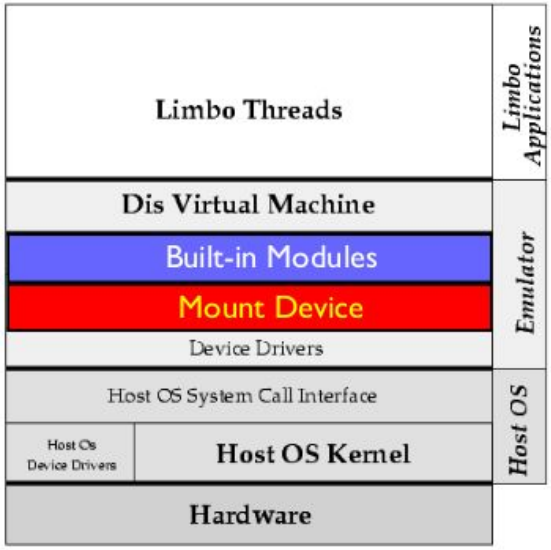
```
B: adt {
    year: int;
    month: string;
    day: int;
    age : fn(me : B) : int;
};
```

- Types are on the right side, see: Golang

How do these compose?



Native (i.e., running directly over hardware)



Hosted (i.e., emulator)

Ok cool, what does it run on?

- Whatever, really
- Not 64-bit
  - Requires significant modifications to Dis
- List:
  - DragonFlyBSD
  - FreeBSD
  - Irix
  - Linux
  - MacOSX
  - NetBSD
  - OpenBSD
  - Plan9
  - Solaris
  - Windows

But also the Raspberry Pi

- https://github.com/yshurik/inferno-rpi

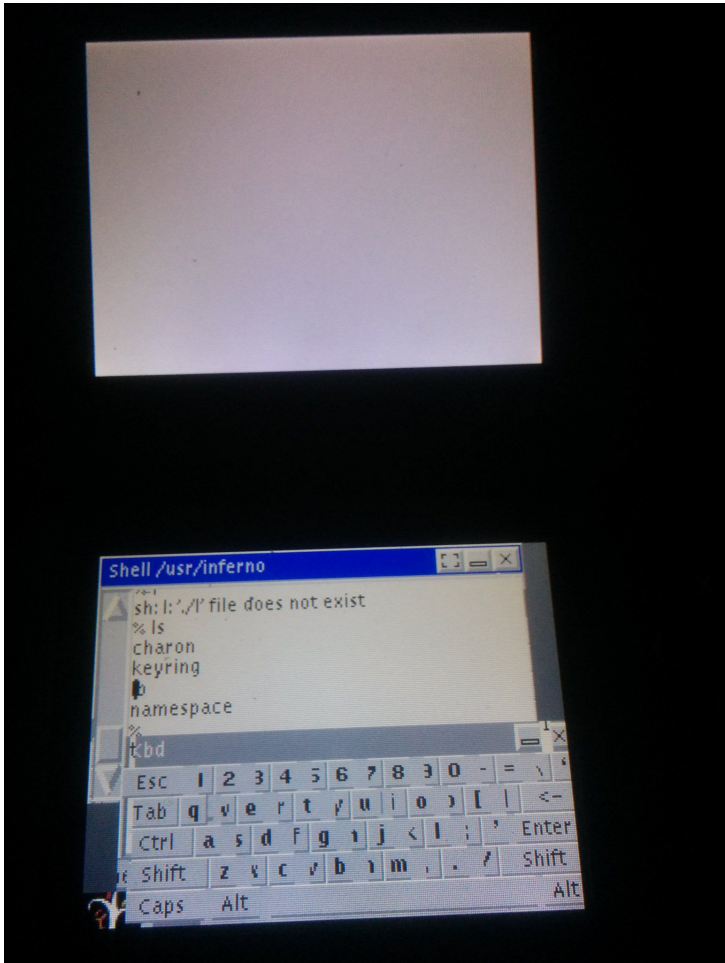
But also Mindstorms RCX

- http://doc.cat-v.org/inferno/4th_edition/styx-on-a-brick/

`llp.h`	Link level protocol constants
`styx.c`	The firmware implementation
`styx.srec`	The firmware image (S-record format)
`styx_abp.srec`	The firmware with the alternating bit part of the link protocol enabled
`send.b`	Test app – sends RCX op codes to the brick
`firmdl.b`	Firmware download app
`rcxsend.m`	Util module header
`rcxsend.b`	Util module – supports RCX ROM message format on serial link
`timers.m`	Timer module header
`timers.b`	Util module – general purpose timers
`legolink.b`	Implements the link protocol via a limbo file2chan()
`clockface.b`	The controller app for our Clockface robot

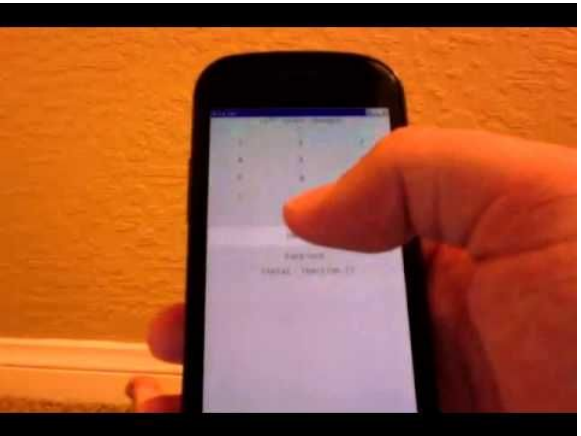
But also the Nintendo DS

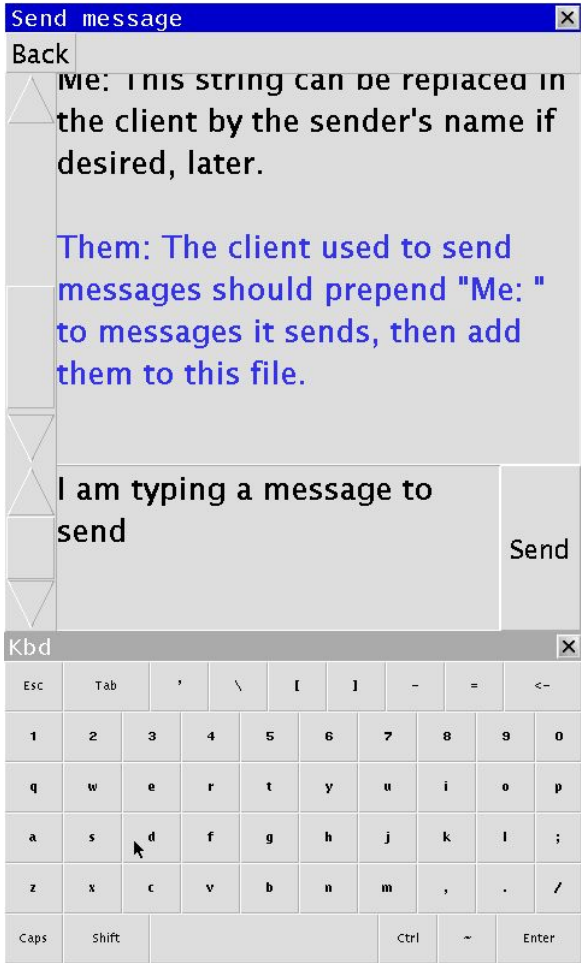
- https://bitbucket.org/mjl/inferno-ds/src



But also Android

- https://www.youtube.com/watch?v=9UNqJN0vuE4
- https://bitbucket.org/floren/inferno/wiki/Home
- https://github.com/bhgv/Inferno-OS_Android





But also...

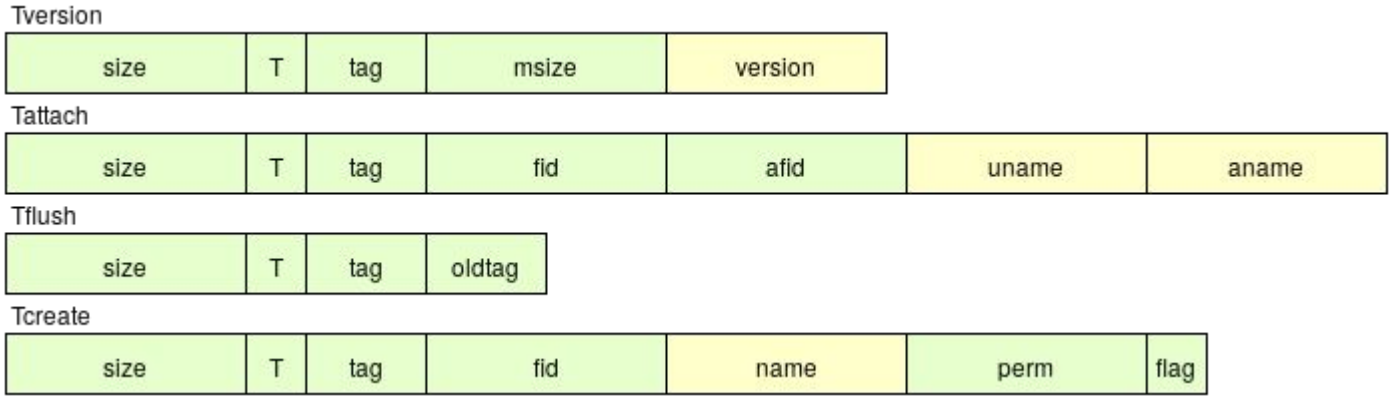
- Marvell Kirkwood (Sheevaplug)
  - https://bitbucket.org/mjl/inferno-kirkwood/src/default/
- OpenMoko
  - https://code.google.com/archive/p/inferno-openmoko/source/default/source
- etc.

Differences from Plan 9?

- The core concept
- Solve different problems, etc.
- Rio, somehow, looks better

The cool shit

- Everything is a file (again)
- Everything speaks 9p (or styx, in this case)
- Inferno can manage registries of permissions
- Network connections are files



More cool shit

- Files can be exported, shared, authenticated for use
- Inferno instances can multiplex each other
- Inferno instances can run on anything™
- Inferno doesn’t care what it’s running on

- Result? Meta-programming with meta-abstractions!

Cool namespace usage

- `; mount net!192.168.56.101 /n/remote`
- `; bind -a /n/remote/prog /n/prog`

What can we do?

- Control any process on the remote machine
- Debug any crashed process on the remote machine
- If bound into place, ps(1) on the remote machine

Cool shell features

- The shell reflects the traits of limbo very readily
- Shell features are limbo modules
  - Can be loaded and unloaded at runtime
  - Stored in `/dis/sh/` by default

Ex.

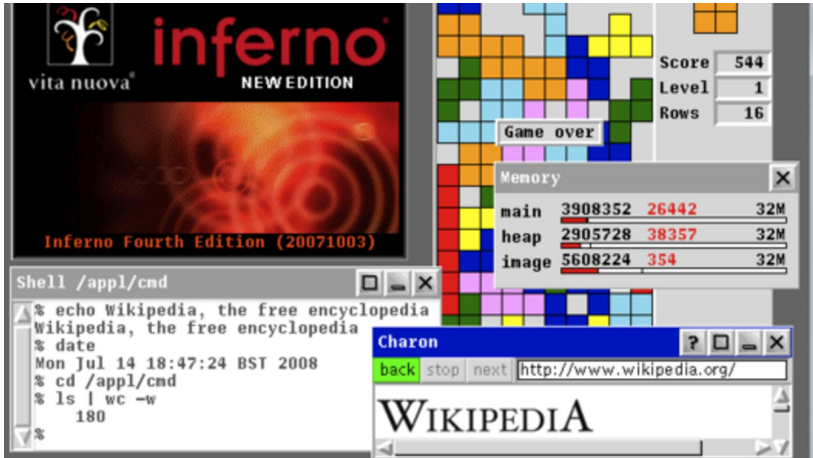
```
; loaded
# list of modules printed
; unload regex
; load regex
```

Some connotations

- Write programs once, use many
  - Like the jvm
- All the benefits of 9p abstractions
- Benefits from CSP-oriented language (limbo)
- Result? Fast, portable, and simple services

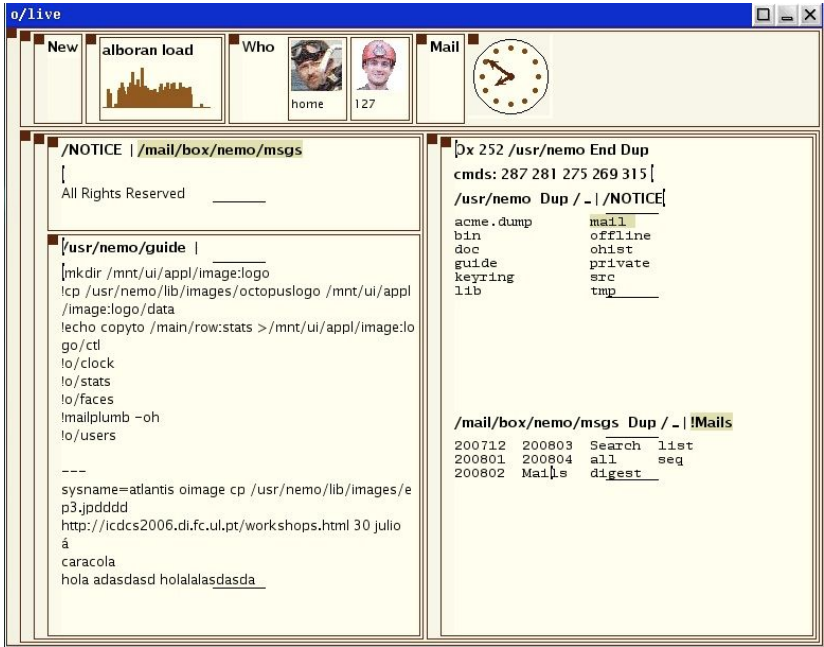
Software highlights

- os(1) — interface to host OS commands (hosted Inferno only)
- collab(8) — chat, poll, poller, whiteboard - collaborative activities
- dmview(1) — dmview, dmwm - view remote displays
- wm(1) — window manager
  - Optional for many graphical programs
- spree(4) — distributed interactive sessions
- registry(4) — dynamic registration of services
- fs(3) — host filesystem interface (#U driver)
- A suite of grid software
  - `; man grid`
- `/net` — network filesystem



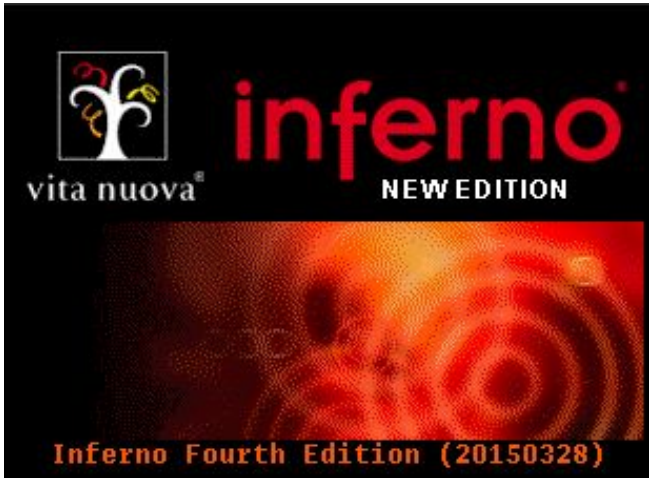
The Octopus

- http://lsub.org/ls/octopus.html
- Provide ubiquitous access to computing resources
- Leverage Inferno portability
- Everything can connect to one thing
- Facilitate distributed computing and resource sharing



What sucks?

- The docs, but only kind of, and only sometimes
  - Many sources of information
  - Some official papers outdated
- The primary source branch isn't updated frequently
  - It's stable, though
- Many resources to make tasks easier are closed source
  - Thanks Alcatel-Lucent



What can you make?

- Programs which are:
  - Networked
  - Concurrent
  - Fast
  - Memory-compact
    - As little as 1MiB of memory!
  - Secure™
  - Portable
- Any tool could be recomposed in Inferno and run on any platform
  - Ex. Acme-SAC — https://github.com/caerwynj/acme-sac
- You can just vendor a pre-compiled Inferno
  - .exe/.app could just contain startup scripts

Some notable resources

- pete — http://debu.gs/tags/inferno
- caerwyn — http://ipn.caerwyn.com/
- powerman — https://powerman.name/
- mjl — https://www.ueber.net/who/mjl/inferno/
- bls — http://umdrive.memphis.edu/blstuart/htdocs/index.html
- nemo — http://lsub.org/who/nemo/

I composed an Inferno resource archive (incomplete):
https://drive.google.com/file/d/1o_XyCZhOD7WkQ4xFQw1vqpdAr3FRkbjr/view

References

- https://www.youtube.com/watch?v=3d1SHOCCDn0
- https://www.youtube.com/watch?v=dF_-jQc53jw
- http://doc.cat-v.org/inferno/
- http://vitanuova.com/
- http://code.9front.org/hg/purgatorio/
- https://bitbucket.org/inferno-os/inferno-os/
- http://www.gemusehaken.org/ipwl/
- http://debu.gs/blog/archives