

98-023A : Concurrent and Distributed Programming w/ Inferno and Limbo

Phillip Stanley-Marbell
pstanley@ece.cmu.edu

Lecture Outline

- **Course overview**
 - Objectives
 - Syllabus
 - Materials / software / textbook
 - Grading
- **Inferno Overview**
- **Relevant History**
- Reading, Homework
- Next lecture: Abstraction, names and resources as “names” (files) in Inferno

Course Outline : Objectives

- To give you a good understanding of the *ideas* behind the **Inferno operating system** and the **Limbo programming language**
- To understand how to take advantage of properties of the systems discussed to **build interesting applications that run across multiple computers**
- To delve into some **issues pertaining to concurrent programming**
 - Debugging concurrent programs
 - Verifying behavior of concurrent programs using formal tools

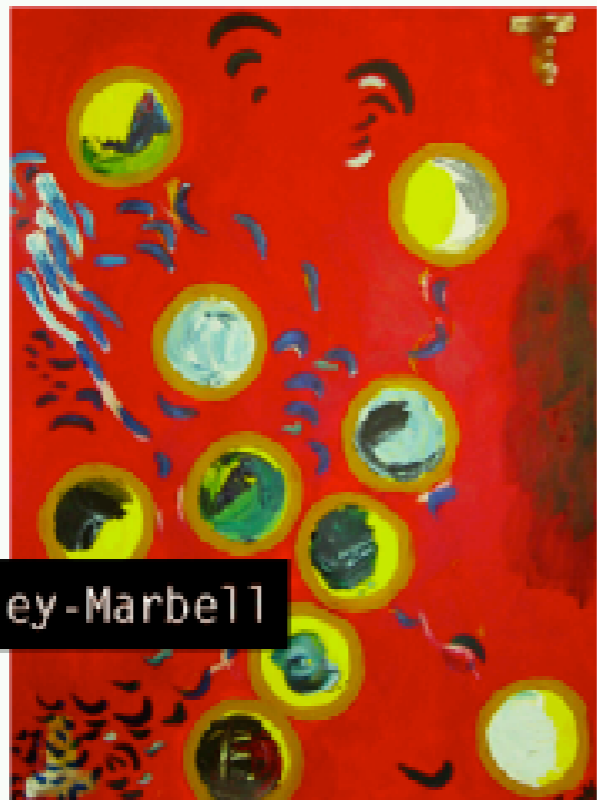
Course Outline : Syllabus

- **Week 1:** Introduction to Inferno
 - **Week 2:** Overview of the Limbo programming language
 - **Week 3:** Types in Limbo; The Dis Virtual Machine
 - **Week 4:** Abstraction as a design tool, Names, Resources as files
 - **Week 5:** 9P and Styx, Resources as files and Limbo threads
 - **Week 6:** C applications as resource servers: Built-in modules and device drivers
 - **Week 7:** Case study I --- building a distributed multi-processor simulator
 - **Week 8:** Platform independent Interfaces: Limbo GUIs; Project Update **Spring Break**
-
- **Week 9:** Programming with threads, CSP
 - **Week 10:** Debugging concurrent programs; Promela and SPIN
 - **Week 11:** Factotum, Secstore and Inferno's security architecture
 - **Week 12:** Case study II --- Edisong, a distributed audio synthesis and sequencing engine

Course Outline : Text / Software

- There is no required textbook
- Having a copy of “Inferno Programming with Limbo” helpful
(Copies are available in the CMU gift store (upstairs, not the bookstore), Amazon, etc.)
 - I will make 2 copies available to anyone who wants to borrow them
- Handouts will be made available the day *after* each lecture
- You should download the Inferno distribution and install it
(<http://www.vitanuova.com>, PC/MacOSX/Linux/*BSD)
- All readings / homework will be distributed via
<http://www.cmu.edu/blackboard>

Inferno Programming with Limbo



Phillip Stanley-Marbell

WILEY

Book's web page

<http://www.ece.cmu.edu/~pstanley/ipwl/>

Complete source for all examples
from book, and more

Free Review Copy

<http://dsonline.computer.org/books/list.htm>

Great opportunity to see your review in print, and get
a free copy to boot!

Course Outline : Grading

- First 4 homeworks are mandatory, the remainder are optional (5% each)
- 1 mini project (20%)
- 1 final project (60%)
- You should not be worried about your grade

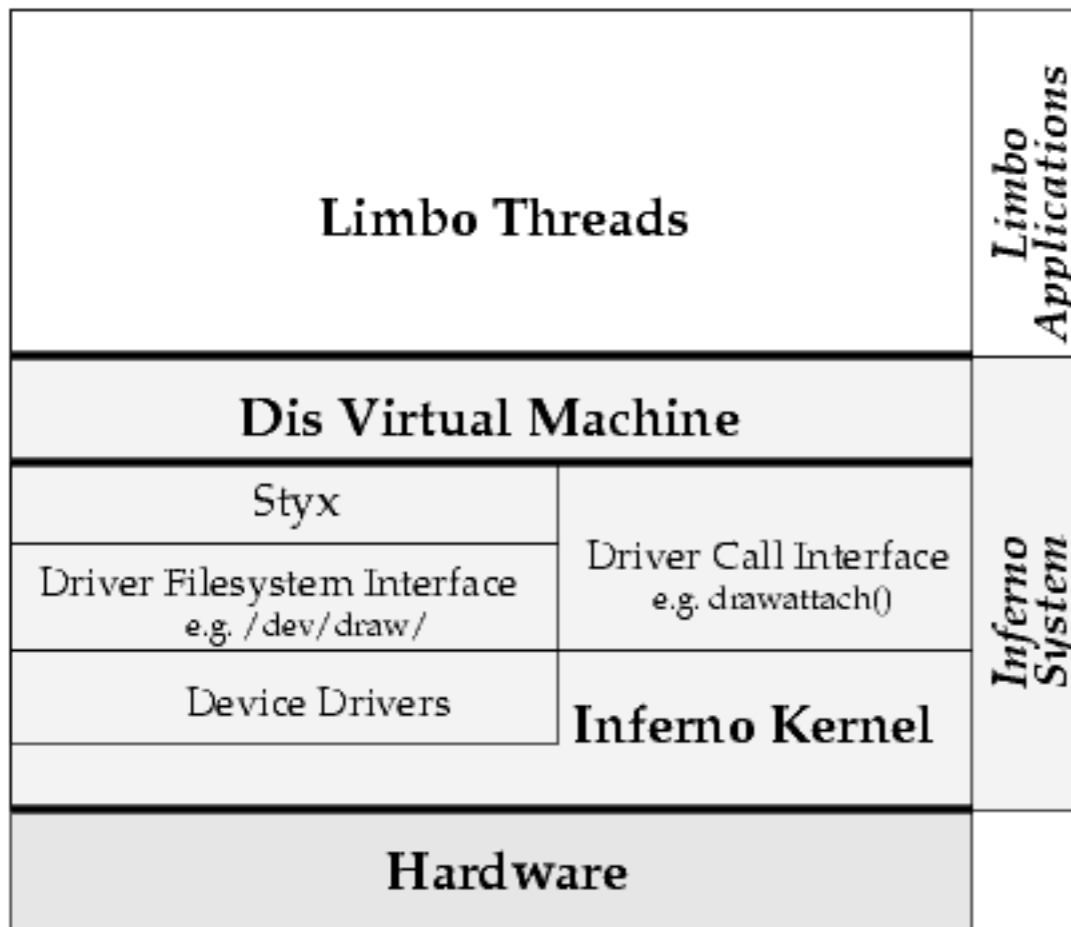
Introduction: Terminology

- **Inferno**
 - An operating system
- **Limbo**
 - A programming language for developing applications for Inferno
- **Dis**
 - Inferno abstracts away the hardware with a virtual machine, the Dis VM
 - Limbo programs are compiled to bytecode for execution on the Dis VM
- **Plan 9**
 - A research operating system, being actively developed at Bell Labs and elsewhere
 - A direct ancestor of Inferno

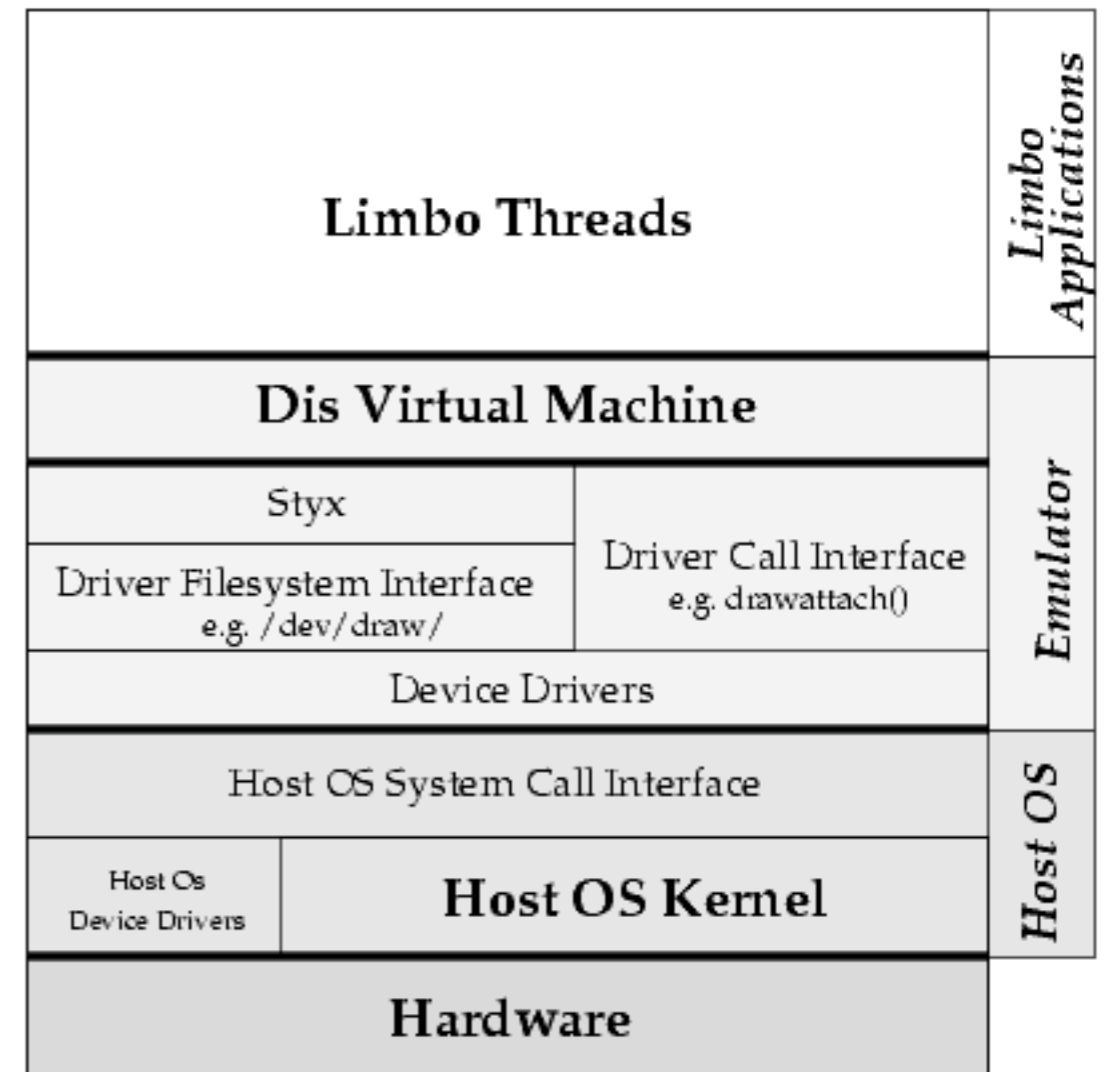
Inferno

- Like any other traditional OS, Inferno **runs directly over bare hardware** (PowerPC, Intel x86, SPARC, MIPS, ARM, more...)
- **Also available as an emulator** which runs over many modern operating systems (Windows, Linux, *BSD, Solaris, IRIX, MacOS X)
- **Emulator provides interface identical to native OS, to both users and applications**
 - Filesystem and other system services, applications, etc.
 - The emulator **virtualizes the entire OS, including filesystem, network stack, graphics subsystem --- everything --- not just code execution** (e.g., in Java Virtual Machine)

Native and Hosted Environments



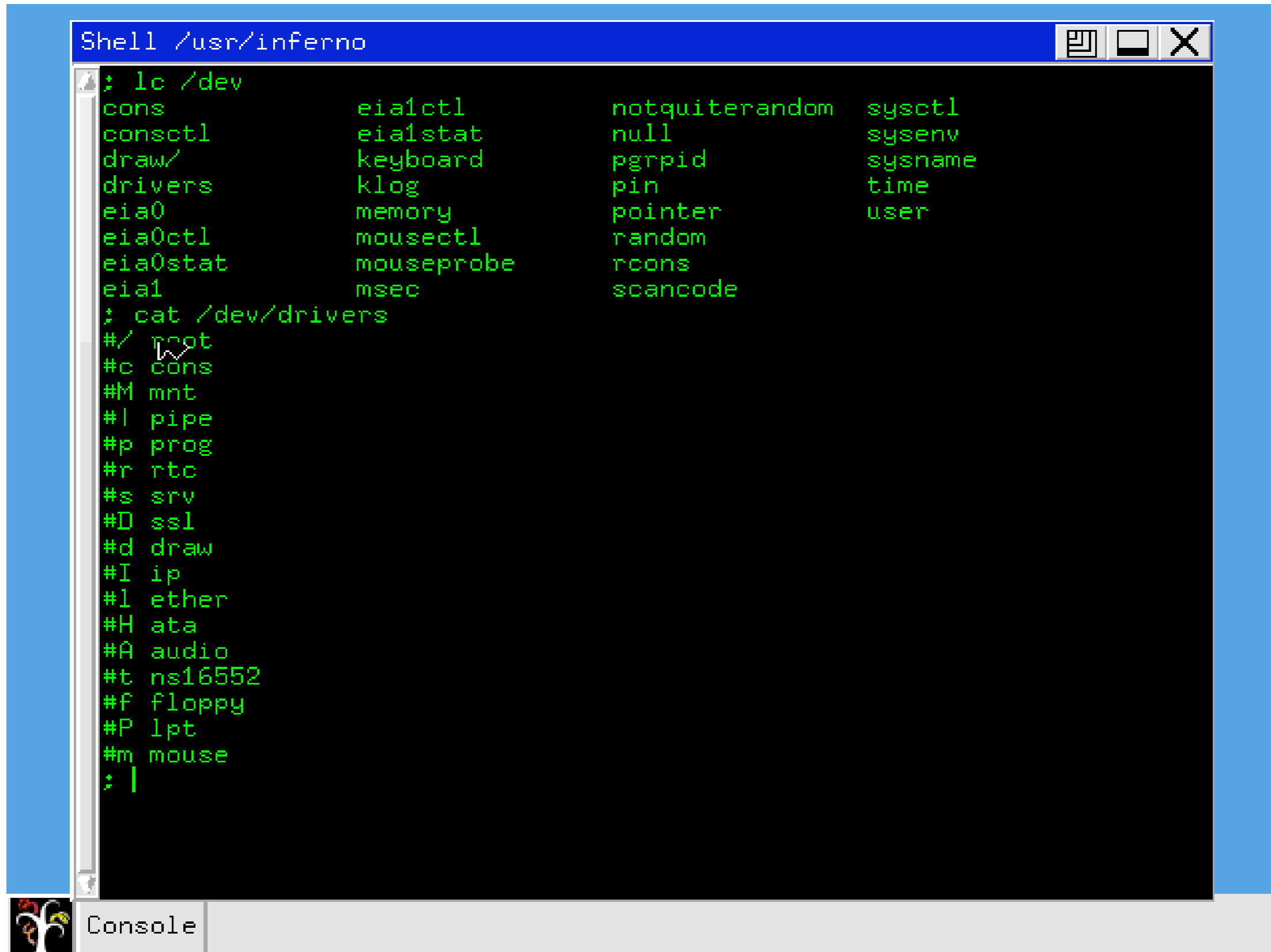
Native (i.e., running directly over hardware)



Hosted (i.e., emulator)

Demo

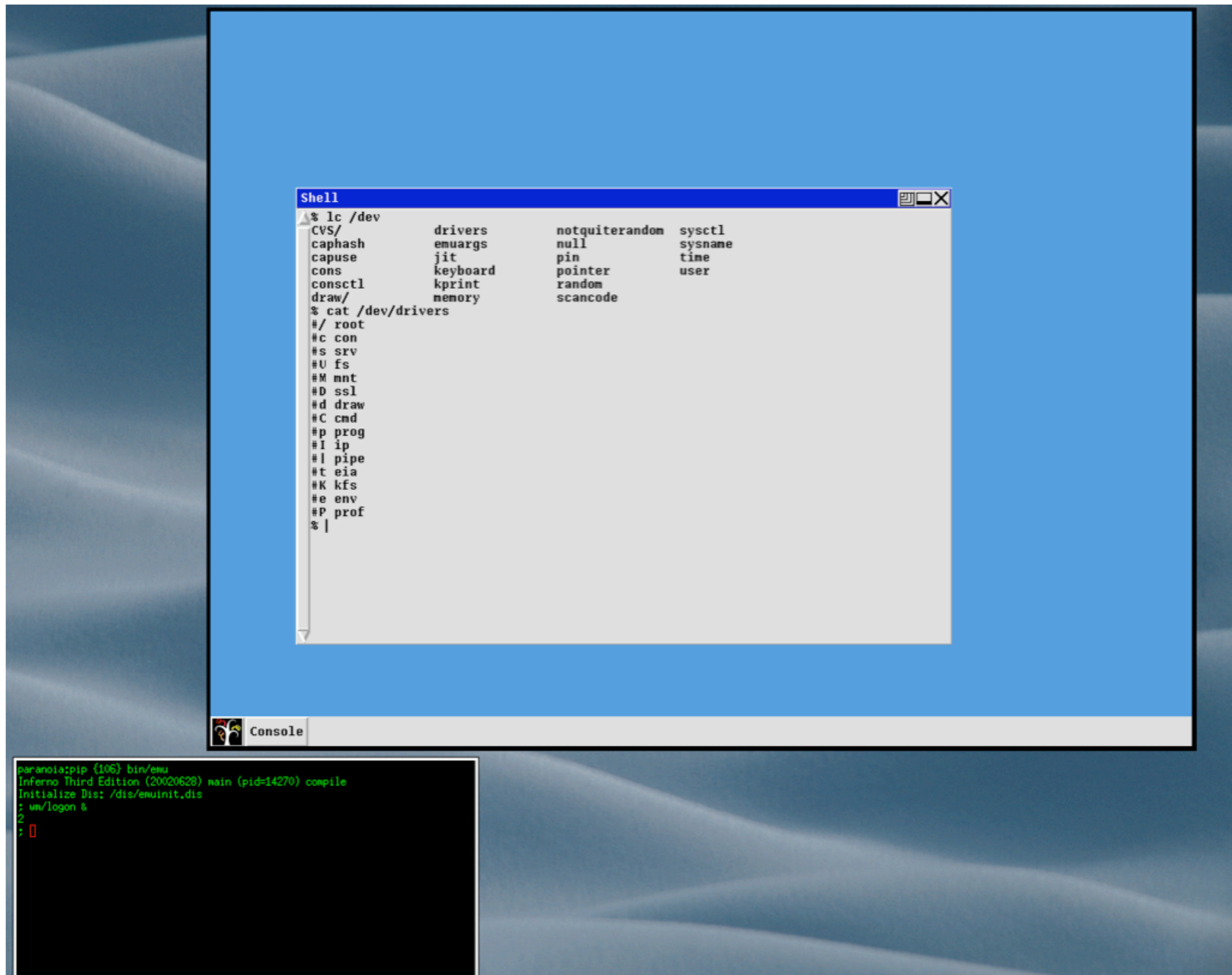
Native Inferno Screenshot



```
Shell /usr/inferno
: ls /dev
cons          eia1ctl      notquiterandom  sysctl
consctl      eia1stat    null            sysenv
draw/        keyboard    pgrp            sysname
drivers      klog        pin             time
eia0         memory      pointer         user
eia0ctl     mousectl    random
eia0stat    mouseprobe  rcons
eia1        msec        scancode
: cat /dev/drivers
#/ root
#c cons
#M mnt
#l pipe
#p prog
#r rtc
#s srv
#D ssl
#d draw
#I ip
#l ether
#H ata
#A audio
#t ns16552
#f floppy
#P lpt
#m mouse
: |
```

Console

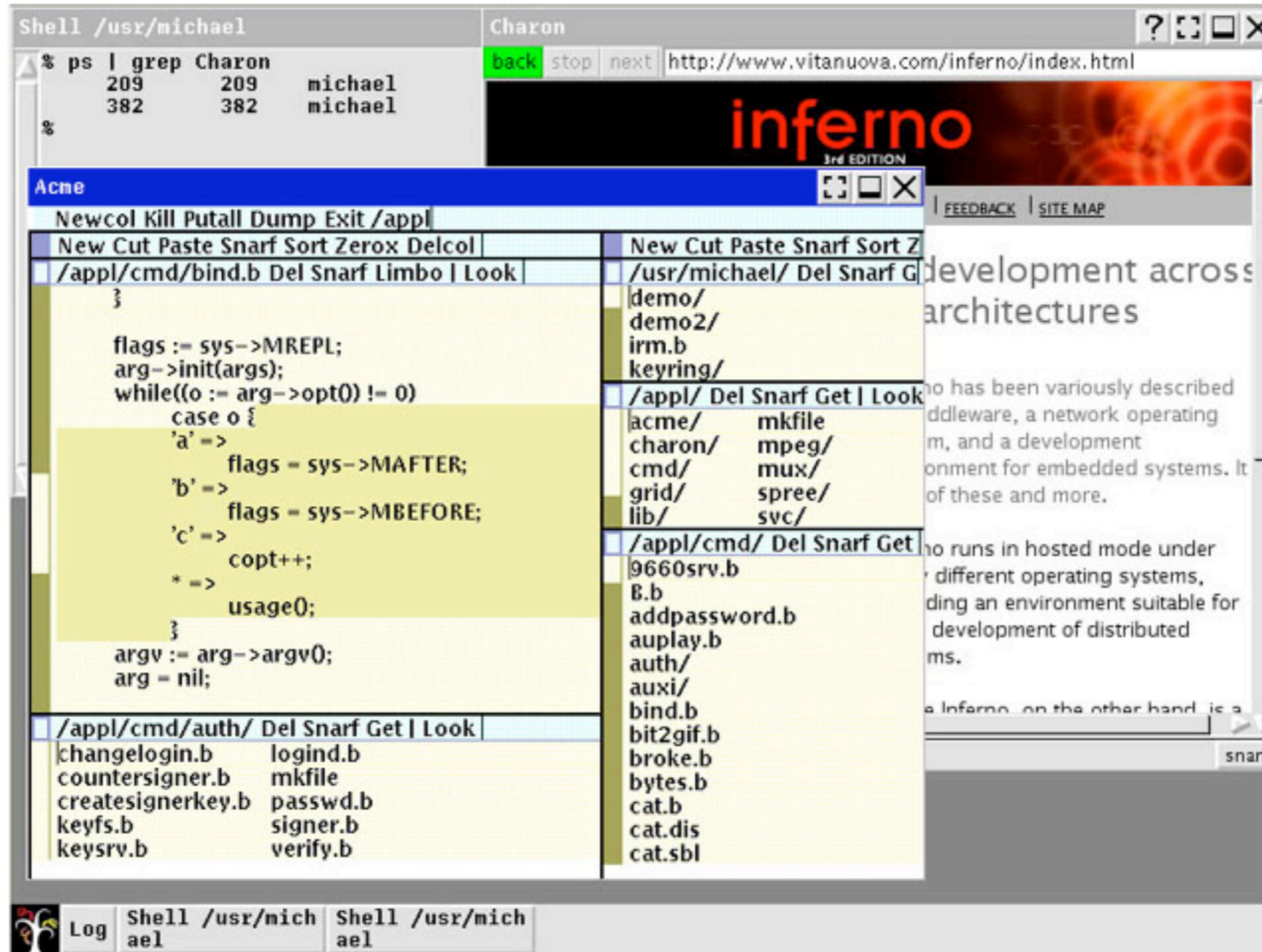
Inferno Emulator on OpenBSD



Available Software

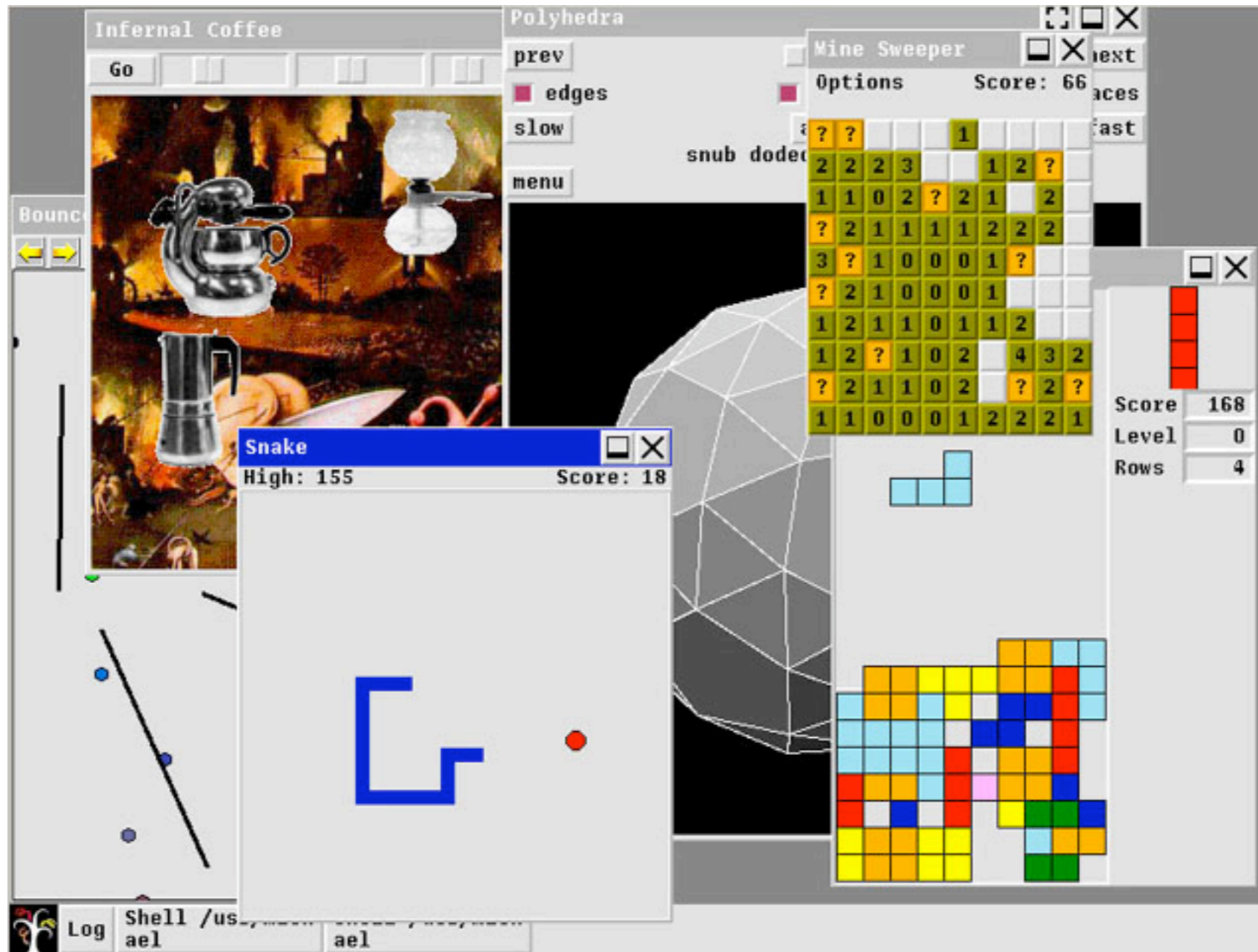
- Text/SGML editors
- Web browser, WML browser, Mail Client
- Graphical debugger
- Games
- “Grid computing” tools
- Clones of Unix tools ([sed](#), [banner](#), etc.)
- Other (not part of the distribution)
 - Audio editor / sequencer / synthesis
 - Image manipulation tools

Applications



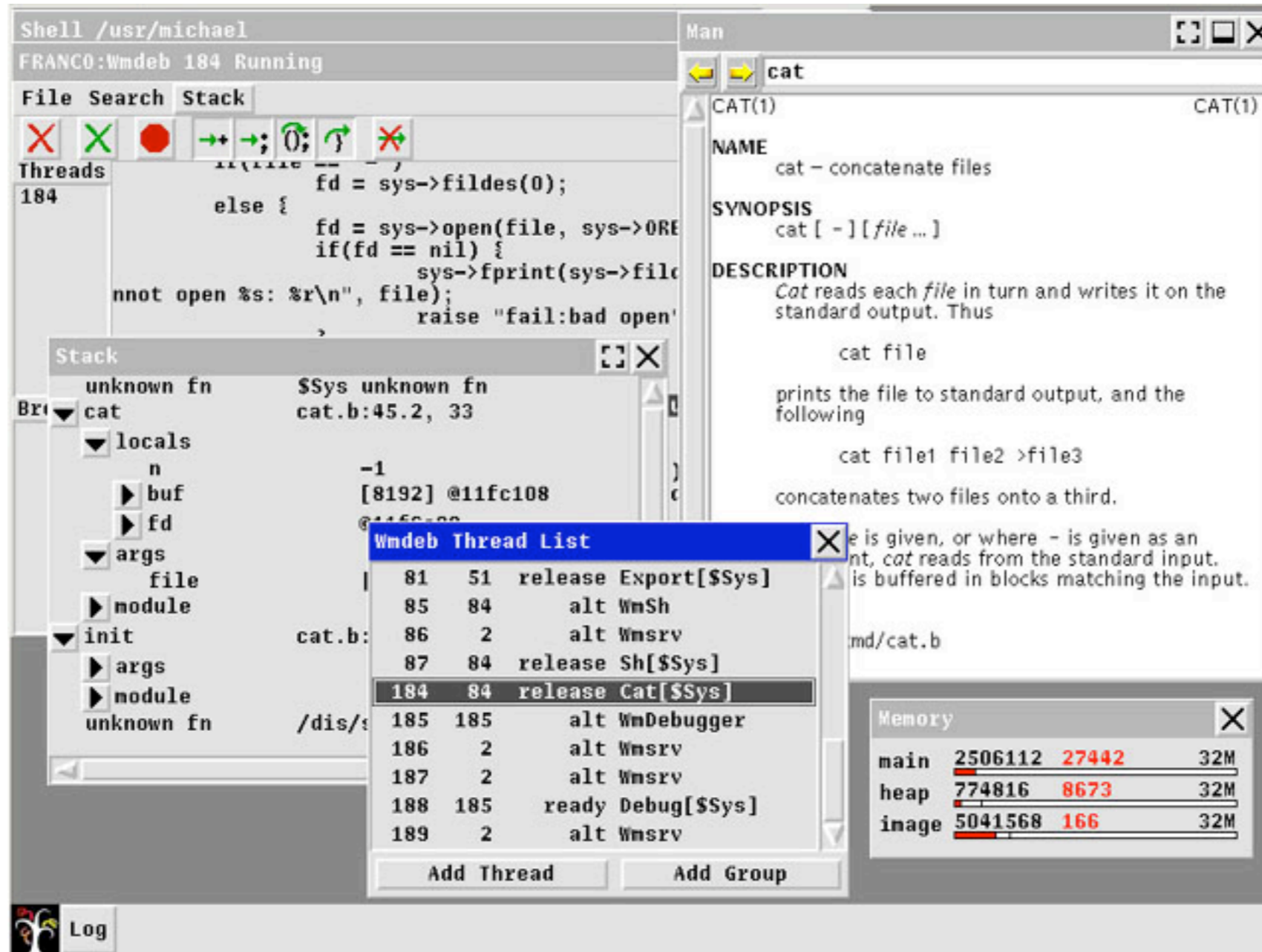
- Shell, Acme editor, Charon web browser

Applications: Games



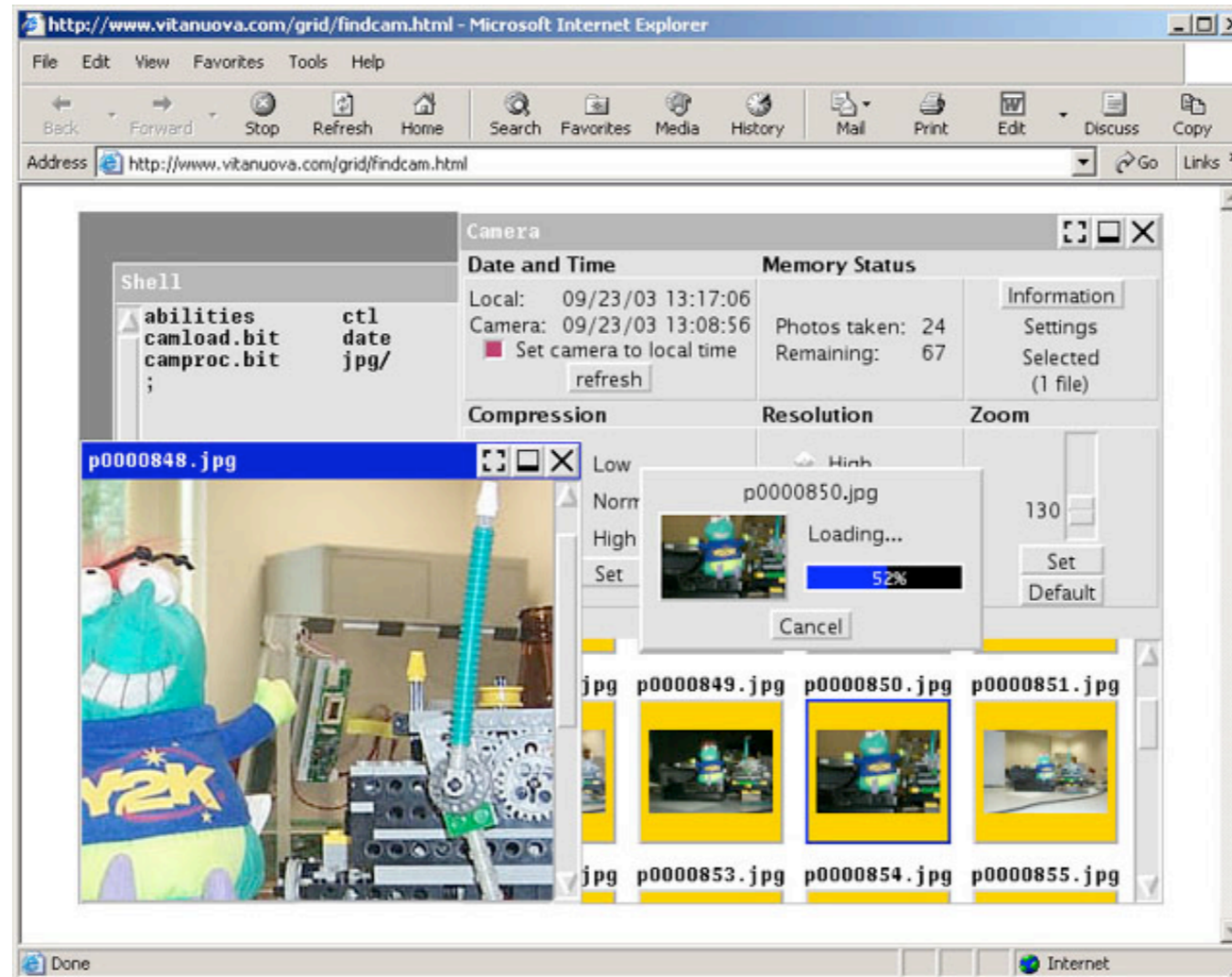
- “Coffee”, Snake, Polyhedra, Mine Sweeper, Tetris

Applications: Developer Utilities



- Graphical debugger, thread list, manual page browser, memory monitor

Inferno IE4 plugin



- Emulator is available as an Internet Explorer plugin, under windows
 - Run entire system (shell, compiler, web browser, filesystem utils., etc.) inside a browser window

History

- Developed, circa 1995, by the Computing Science Research Center at Bell Labs

Sean Dorward, Rob Pike, David Presotto, Dennis Ritchie, Howard Trickey, Phil Winterbottom



- Drew upon lessons learned building Plan 9 research OS
- Developed as a decidedly commercial product
 - Developed by Bell Labs research, transferred to the “Inferno Business Unit” a semi-autonomous spinoff, housed within Lucent (Murray Hill, NJ)

History

- Inferno Business Unit attempted to market Inferno as the ideal solution for developing “Screen Phones”



Philips Consumer Communications IS2630 Screen Phone
Developed by Philips facility in Eatontown NJ,
Manufactured in Guadalajara Mexico

- Using Inferno for “screen phones” was not a particularly inspired idea.
- No significant improvement of prototype (as delivered by research group) is done before attempting to sell it
 - Circa 1999, Inferno Business Unit folds.

History

- 2000: Rights to Inferno sold to Vita Nuova Holdings, York England



- VN was co-founded by Charles Forsyth, who had been actively involved in Plan 9 and Inferno work, and wrote the Plan 9 PowerPC compiler
- Source code license reduced from \$300,000 to \$100, more liberal license
- 2003: Inferno, including source code to entire system, made available under a dual licensing scheme, for free

History

- Some commercial Products that used inferno

- Lucent Managed Firewall



- Lucent PathStar Access Server

- PathStar was a carrier-grade voice-over-IP platform
- Served multiple POTS phone lines (bottom of cabinet), converging both voice traffic and data traffic, to be sent over a WAN, possibly to another PathStar
- Inferno used for control / management / user interface
- Custom “microkernel”, LCOS, run on the Line Cards
- Original prototype designed and built primarily by Phil Winterbottom and Ken Thompson

- Philips Consumer Communications IS2630 Screen phone



Axes are not good for shaving



- Some personal opinions

- Inferno is an interesting system, with lots of technically interesting ideas (*this course*)
- It is particularly useful when resources have to be distributed over a heterogeneous system (multiple hardware architectures, etc.), across a network (*we'll learn why*)
- It has a very small memory footprint, easily runs in systems with under 2MB of RAM, making it well suited for use in embedded systems
- Has a very clean design, making it relatively easy to port to new hardware architectures
- **It is not a realtime operating system, no realtime schedulers, so its use in embedded systems may be restricted**
- **Does not have a particularly aesthetically pleasing GUI, so i would not say it is a good choice for PDAs, let alone “screen phones” (though you can roll your own GUI)**

Homework I

1. Login to <http://www.cmu.edu/blackboard>, get familiar
2. Send me an email saying what you hope to learn in the class, say if there are any particular things not in the syllabus that you would like to see covered
3. Do *either of the 2 readings* for lecture I
 - Grading: you get full credit by sending me the above email, i'll assume you did the reading.
 - Due (i.e., the email above) any time before class on wednesday 01/14/2004

Reading for Lecture 1

- Readings available under “course documents/relevant reading/week1” on <http://www.cmu.edu/blackboard>
- “Inferno”
S. Dorward, R. Pike, D. L. Presotto, D. Ritchie, H. Trickey, and P. Winterbottom, In *Proceedings of IEEE Compcon 1997*.
- “Inferno Application Development with Limbo”
P. Stanley-Marbell, In *Dr. Dobbs Journal of Software Tools*, December 2000
- Reading either or both articles will help prepare you for lecture 2
- Additional reading (if you want more information)
 - References 15, 18, 19, 20, 53, 54, 60 and 72 from bibliography in *Inferno Programming with Limbo*

Next lecture

- A real (i.e., more in depth) introduction to Inferno
 - Abstractions, Names and Resources as files in Inferno

Fin.