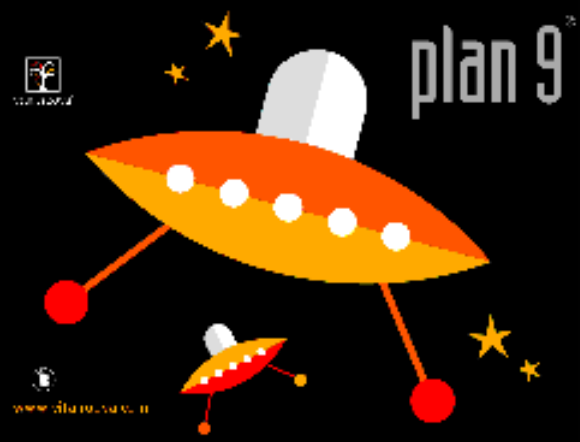
# Plan 9 from Bell-Labs

Computer Science 400 project by:
Andrey Mirtchovski
Supervised by:
Tony Kusalik

# Introduction




- An operating system designed by the people who created UNIX
- Designed with distributed computing in mind
- Main goals concern performance and usability in said environment

# Topics of Discussion

- What is plan9?
- Why plan9?
- Project goals
- Project tentative setup
- Early implementation stages (or “Having old hardware == trouble”)
- Late implementation stages (or “Having little time == trouble”)

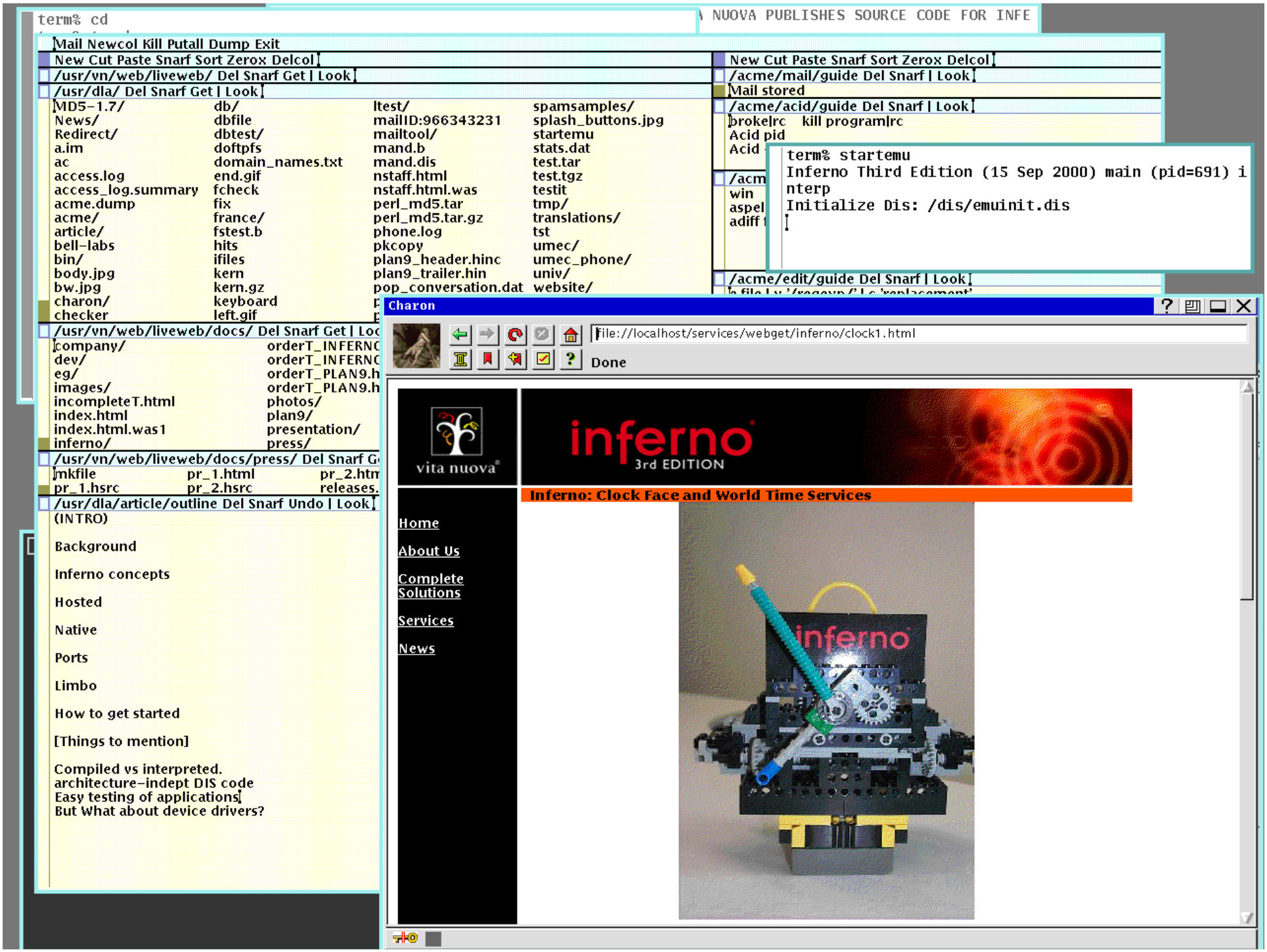
# Topics...

- Setup
  - Mashines in initial project state
  - Machines in adult project state
- Results
  - Usability
  - Computational fitness
- Conclusions

# What is Plan9?

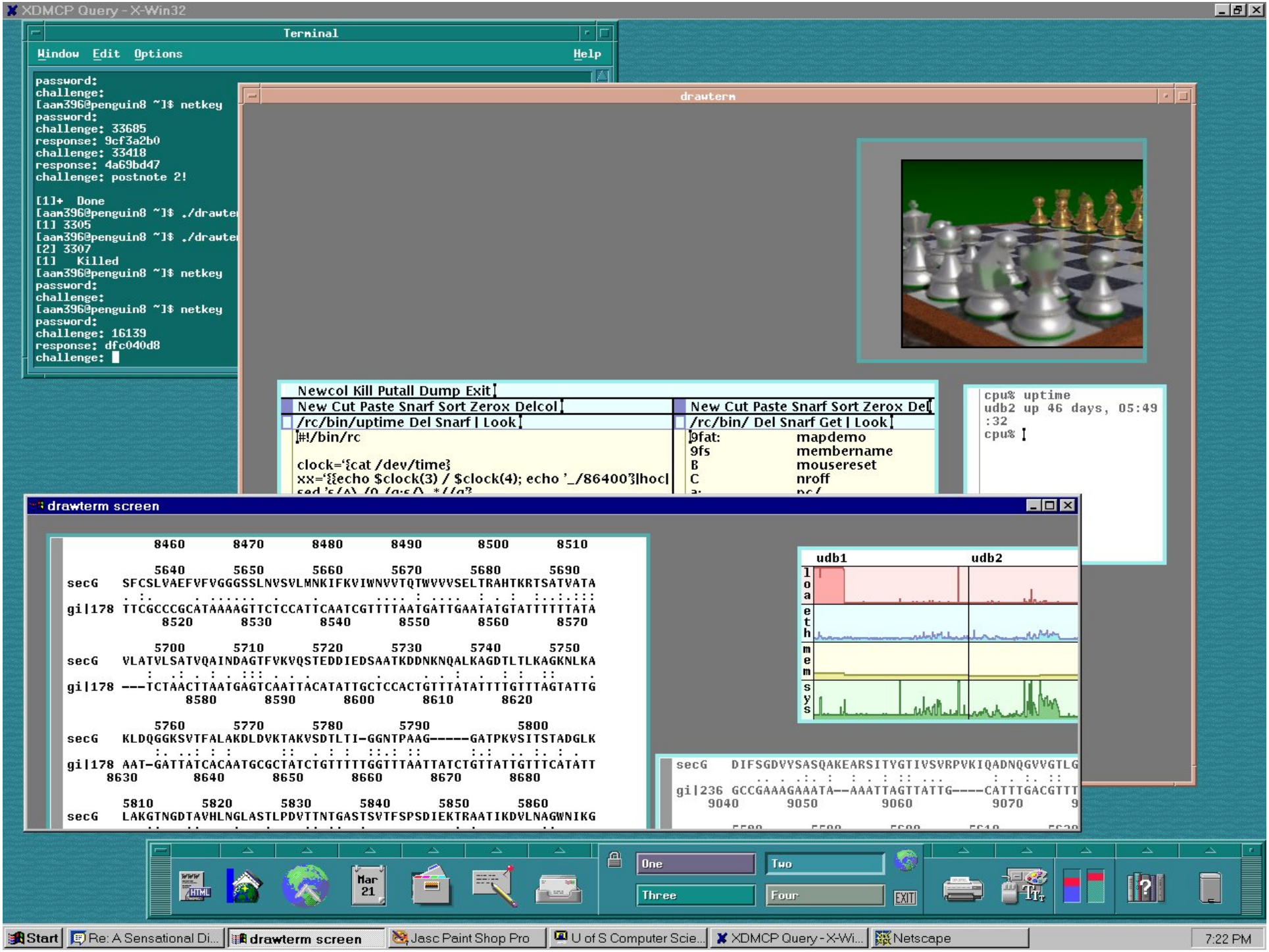
- OS
- Created by programmers for programmers
- Has all the requirements for a full-blown commercial OS minus software and price
- Distributed
- **Research-Oriented**

- Full set of development tools including
  - Compiler for several platforms
  - Text editors (more on that later)
  - Libraries
- GUI
- Full network connectivity
- Open Source


term% cd
NUOVA PUBLISHES SOURCE CODE FOR INFE
Mail Newcol Kill Putall Dump Exit
New Cut Paste Snarf Sort Zerox Delcol
/usr/vn/web/liveweb/ Del Snarf Get | Look
/usr/dla/ Del Snarf Get | Look
MD5-1.7/ db/ ltest/ spamsamples/
News/ dbfile mailID:966343231 splash_buttons.jpg
Redirect/ dbtest/ mailtool/ startemu
a.im doftpfs mand.b stats.dat
ac domain_names.txt mand.dis test.tar
access.log end.gif nstaff.html test.tgz
access_log.summary fcheck nstaff.html.was testit
acme.dump fix perl_md5.tar tmp/
acme/ france/ perl_md5.tar.gz translations/
article/ fstest.b phone.log tst
bell-labs hits pkcopy umec/
bin/ ifiles plan9_header.hinc umec_phone/
body.jpg kern plan9_trailer.hin univ/
bw.jpg kern.gz pop_conversation.dat website/
charon/ keyboard
checker left.gif
/usr/vn/web/liveweb/docs/ Del Snarf Get | Loc
company/ orderT_INFERNC
dev/ orderT_INFERNC
eg/ orderT_PLAN9.h
images/ orderT_PLAN9.h
incompleteT.html photos/
index.html plan9/
index.html.was1 presentation/
inferno/ press/
/usr/vn/web/liveweb/docs/press/ Del Snarf G
mkfile pr_1.html pr_2.htm
pr_1.hsrc pr_2.hsrc releases.
/usr/dla/article/outline Del Snarf Undo | Look
(INTRO)
Background
Inferno concepts
Hosted
Native
Ports
Limbo
How to get started
[Things to mention]
Compiled vs interpreted.
architecture-indept DIS code
Easy testing of applications
But What about device drivers?
New Cut Paste Snarf Sort Zerox Delcol
/acme/mail/guide Del Snarf | Look
Mail stored
/acme/acid/guide Del Snarf | Look
broke|rc kill program|rc
Acid pid
Acid
/acm
win
aspel
adiff
/acme/edit/guide Del Snarf | Look
term% startemu
Inferno Third Edition (15 Sep 2000) main (pid=691) i
nterp
Initialize Dis: /dis/emuinit.dis
Charon
file://localhost/services/webget/inferno/clock1.html
Done
vita nuova
inferno
3rd EDITION
Inferno: Clock Face and World Time Services
Home
About Us
Complete Solutions
Services
News
inferno

# Typical Plan9 Setup

- CPU
  - One or many
- Terminals
  - Typically one per user
  - Minimal hardware requirements
- File Servers
  - Could be incorporated with cpu servers
- Auth Servers
- Security


XDMCP Query - X-Win32
Terminal
Window Edit Options Help
password:
challenge:
[aam396@penguin8 ~]$ netkey
password:
challenge: 33685
response: 9cf3a2b0
challenge: 33418
response: 4a69bd47
challenge: postnote 2!
[1]+ Done
[aam396@penguin8 ~]$ ./drawte
[1] 3305
[aam396@penguin8 ~]$ ./drawte
[2] 3307
[1] Killed
[aam396@penguin8 ~]$ netkey
password:
challenge:
[aam396@penguin8 ~]$ netkey
password:
challenge: 16139
response: dfc040d8
challenge:
drawterm
Newcol Kill Putall Dump Exit
New Cut Paste Snarf Sort Zerox Delcol
/rc/bin/uptime Del Snarf | Look
#!/bin/rc
clock='{cat /dev/time}
xx='{{echo $clock(3) / $clock(4); echo '_/86400'}|hoc|
New Cut Paste Snarf Sort Zerox Del
/rc/bin/ Del Snarf Get | Look
9fat: mapdemo
9fs membername
B mousereset
C nroff
cpu% uptime
udb2 up 46 days, 05:49
:32
cpu%
drawterm screen
8460 8470 8480 8490 8500 8510
5640 5650 5660 5670 5680 5690
secG SFCSLVAEFVFVGGGSSLNVSVLMNKIFKVIWNVVTQTWVVVSELTRAHTKRTSATVATA
gi|178 TTCGCCCGCATAAAAGTTCTCCATTCAATCGTTTTAATGATTGAATATGTATTTTTTATA
8520 8530 8540 8550 8560 8570
5700 5710 5720 5730 5740 5750
secG VLATVLSATVQAINDAGTFVKVQSTEDDIEDSAATKDDNKNQALKAGDTLTLKAGKNLKA
gi|178 ---TCTAACTTAATGAGTCAATTACATATTGCTCCACTGTTTATATTTTGTTTAGTATTG
8580 8590 8600 8610 8620
5760 5770 5780 5790 5800
secG KLDQGGKSVTFALAKDLDVKTAKVSDTLTI-GGNTPAAG-----GATPKVSITSTADGLK
gi|178 AAT-GATTATCACAATGCGCTATCTGTTTTGGTTTAATTATCTGTTATTGTTTCATATT
8630 8640 8650 8660 8670 8680
5810 5820 5830 5840 5850 5860
secG LAKGTNGDTAVHLNGLASTLPDVTTNTGASTSVTFSPSDIEKTRAATIKDVLNAGWNIKG
udb1 udb2
loa
eth
mem
sys
secG DIFSGDVYSASQAKEARSITYGTIVSVRPVKIQADNQGVVGTLG
gi|236 GCCGAAAGAAATA--AAATTAGTTATTG----CATTTGACGTTT
9040 9050 9060 9070
One
Two
Three
Four
EXIT
Start
Re: A Sensational Di...
drawterm screen
Jasc Paint Shop Pro
U of S Computer Scie...
XDMCP Query - X-Wi...
Netscape
7:22 PM

# Why Plan9?

- Interest in Operating Systems Design and Implementation
- Creators' reputation (hence:)
- Quality of code
- Minimalist design
- **Code understandability**

# Project goals

- Setup and use a working plan9 research environment
- Implement and measure fitness of several Bioinformatics algorithms when implemented in a distributed manner
- Measure performance and compare to other (more robust) operating systems (BSD or Linux)
- Compare ease of use and application development with other OSs

# Bioinformatics

- Purpose of bioinformatics algorithms
- Algorithm used
  - Initially decided on BLAST (huge success in PBI's research)
  - At a later stage moved to FASTA
  - Optional other algorithms never implemented (lack of time)

# Early stage setup

- The horrors of using old, unsupported hardware
- Lost tremendous amount of time within kernel space trying to create some sort of useful environment so the project could start
- From of a month (at most) allotted for initial setup, scope crept to a whole semester

# Early stage setup...

- Learning from experience (benefits from using old hardware):
  - Hands got very dirty very early in the research phase
  - Appreciation for minimalist design and lack of code bloat
  - Appreciation for overall OS and hardware design (serial consoles and cross compilers)
  - Increased understanding of operating system internals and behaviour

# Early network organization

- 3 CPU servers (udb[123]). Alpha UDB machines
- Running NetBSD
- Udb2 attempts to boot Plan9 kernel off udb1
- Hard to use…
- Unsuccessful…

# Late stage setup

- New machines brought thanks to Supervisor and Dave Bocking
- Luck with hardware (all supported)
- Initial installation and setup of Plan9 took less than 5 hours, and installed over the network
- Lack of proper hardware substituted for lack of time

# Current network organization

- 3 CPU servers (udb[123])
- CPU servers act as file servers (locally)
- udb1 acts as an authentication server
- All machines can access vital Plan9 services
- All machines allow terminal access from on-campus machines (using challenge-response based authentication)
- No root!

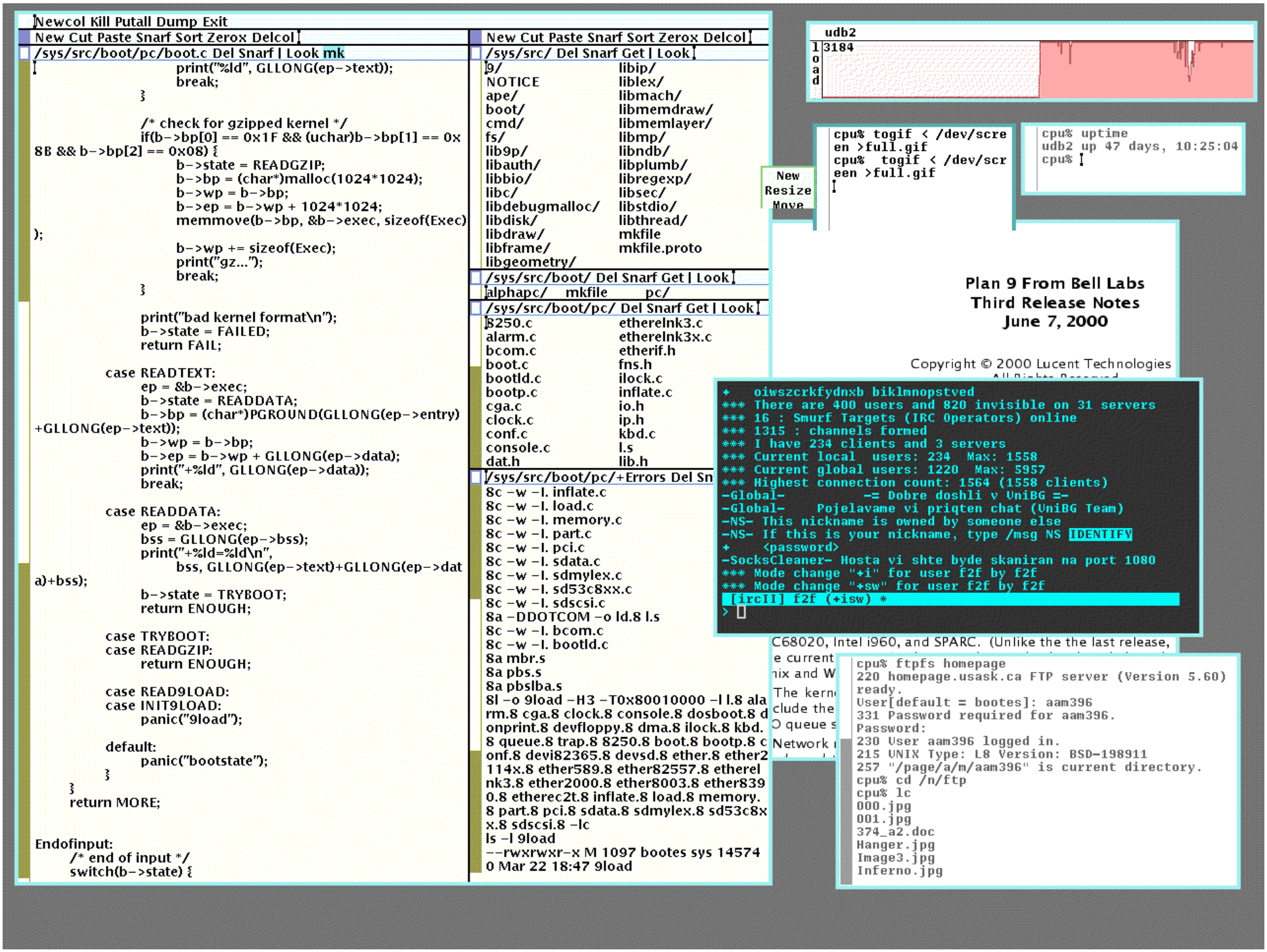
# Results

- Usability
  - Extremely programmer-friendly environment
  - Editor created ‘from programmers for programmers’
  - Debugging is a breeze
  - Libraries and API exclude code bloat (unless one is careless)
    - IRC client – 200 lines of code
    - Graphics – 70 lines of code (bouncing balls)

# Results...

- The one true power of shell scripting:

```
cpu% cat /bin/uptime
#!/bin/rcclock=`{cat /dev/time}
xx=`{{echo $clock(3) / $clock(4);
echo '_/86400'}|hoc|sed 's/^\./0./g;s/\..*//g'}
sec=$xx(1)
days=$xx(2)
cat /dev/sysname
echo -n ' up '$days' days, date $sec |awk
    '{print $4}'
```


Newcol Kill Putall Dump Exit
New Cut Paste Snarf Sort Zerox Delcol
/sys/src/boot/pc/boot.c Del Snarf | Look mk
print("%ld", GLLONG(ep->text));
break;
}
/* check for gzipped kernel */
if(b->bp[0] == 0x1F && (uchar)b->bp[1] == 0x8B && b->bp[2] == 0x08) {
b->state = READGZIP;
b->bp = (char*)malloc(1024*1024);
b->wp = b->bp;
b->ep = b->wp + 1024*1024;
memmove(b->bp, &b->exec, sizeof(Exec));
b->wp += sizeof(Exec);
print("gz...");
break;
}
print("bad kernel format\n");
b->state = FAILED;
return FAIL;
case READTEXT:
ep = &b->exec;
b->state = READDATA;
b->bp = (char*)PGROUND(GLLONG(ep->entry)+GLLONG(ep->text));
b->wp = b->bp;
b->ep = b->wp + GLLONG(ep->data);
print("+%ld", GLLONG(ep->data));
break;
case READDATA:
ep = &b->exec;
bss = GLLONG(ep->bss);
print("+%ld=%ld\n",
bss, GLLONG(ep->text)+GLLONG(ep->data)+bss);
b->state = TRYBOOT;
return ENOUGH;
case TRYBOOT:
case READGZIP:
return ENOUGH;
case READ9LOAD:
case INIT9LOAD:
panic("9load");
default:
panic("bootstate");
}
}
return MORE;
Endofinput:
/* end of input */
switch(b->state) {
New Cut Paste Snarf Sort Zerox Delcol
/sys/src/ Del Snarf Get | Look
9/ libip/
NOTICE liblex/
ape/ libmach/
boot/ libmemdraw/
cmd/ libmemlayer/
fs/ libmp/
lib9p/ libndb/
libauth/ libplumb/
libbio/ libregexp/
libc/ libsec/
libdebugmalloc/ libstdio/
libdisk/ libthread/
libdraw/ mkfile
libframe/ mkfile.proto
libgeometry/
/sys/src/boot/ Del Snarf Get | Look
alphapc/ mkfile pc/
/sys/src/boot/pc/ Del Snarf Get | Look
8250.c ethere1nk3.c
alarm.c ethere1nk3x.c
bcom.c etherif.h
boot.c fns.h
bootld.c ilock.c
bootp.c inflate.c
cga.c io.h
clock.c ip.h
conf.c kbd.c
console.c l.s
dat.h lib.h
/sys/src/boot/pc/+Errors Del Sn
8c -w -I. inflate.c
8c -w -I. load.c
8c -w -I. memory.c
8c -w -I. part.c
8c -w -I. pci.c
8c -w -I. sdata.c
8c -w -I. sdmylex.c
8c -w -I. sd53c8xx.c
8c -w -I. sdscsi.c
8a -DDOTCOM -o ld.8 l.s
8c -w -I. bcom.c
8c -w -I. bootld.c
8a mbr.s
8a pbs.s
8a pbslba.s
8l -o 9load -H3 -T0x80010000 -l l.8 alarm.8 cga.8 clock.8 console.8 dosboot.8 onprint.8 devfloppy.8 dma.8 ilock.8 kbd.8 queue.8 trap.8 8250.8 boot.8 bootp.8 conf.8 devi82365.8 devsd.8 ether.8 ether2114x.8 ether589.8 ether82557.8 etherelnk3.8 ether2000.8 ether8003.8 ether8390.8 etherec2t.8 inflate.8 load.8 memory.8 part.8 pci.8 sdata.8 sdmylex.8 sd53c8xx.8 sdscsi.8 -lc
ls -l 9load
--rwxrwxr-x M 1097 bootes sys 14574 0 Mar 22 18:47 9load
udb2
load
3184
cpu% togif < /dev/screen >full.gif
cpu% togif < /dev/screen >full.gif
New
Resize
Move
cpu% uptime
udb2 up 47 days, 10:25:04
cpu%
Plan 9 From Bell Labs
Third Release Notes
June 7, 2000
Copyright © 2000 Lucent Technologies
+ oiwszcrkfydnxb biklmnopstved
*** There are 400 users and 820 invisible on 31 servers
*** 16 : Smurf Targets (IRC Operators) online
*** 1315 : channels formed
*** I have 234 clients and 3 servers
*** Current local users: 234 Max: 1558
*** Current global users: 1220 Max: 5957
*** Highest connection count: 1564 (1558 clients)
-Global- -= Dobre doshli v UniBG =-
-Global- Pojelavame vi priqten chat (UniBG Team)
-NS- This nickname is owned by someone else
-NS- If this is your nickname, type /msg NS IDENTIFY
+ <password>
-SocksCleaner- Hosta vi shte byde skaniran na port 1080
*** Mode change "+i" for user f2f by f2f
*** Mode change "+sw" for user f2f by f2f
[ircII] f2f (+isw) *
>
C68020, Intel i960, and SPARC. (Unlike the the last release,
e current
nix and W
The kern
clude the
O queue s
Network
cpu% ftpfs homepage
220 homepage.usask.ca FTP server (Version 5.60) ready.
User[default = bootes]: aam396
331 Password required for aam396.
Password:
230 User aam396 logged in.
215 UNIX Type: L8 Version: BSD-198911
257 "/page/a/m/aam396" is current directory.
cpu% cd /n/ftp
cpu% lc
000.jpg
001.jpg
374_a2.doc
Hanger.jpg
Image3.jpg
Inferno.jpg

# Results...

- Porting
  - APE/PSH
  - Some programs compile flawlessly
    - FASTA
    - POVray
  - Bad programming practices lead to unportable/unmaintainable code

# Results...

- Computational fitness
  - Comparable executables speed with linux running on same hardware
  - Some compiler research has lead to significant improvement of execution speed
  - Have in mind – operating system is not tuned for performance (as Linux/BSD are for example)

# Results.. POVray

- Accidentally ported to Plan9
- Used to measure raw CPU performance and compare with Linux
- Speed results:

  (create a complex 3d ray-tracing image)
  - Linux:
    - Total Time: 368 seconds
  - Plan9:
    - Total Time: 415 seconds

Linux

Plan9

## Image Sizes:

```
-rw-------   1 aam396   dip       192074 Feb 27 13:46 linux.ppm
-rw-------   1 aam396   dip       192074 Feb 27 13:47 p9.ppm
```

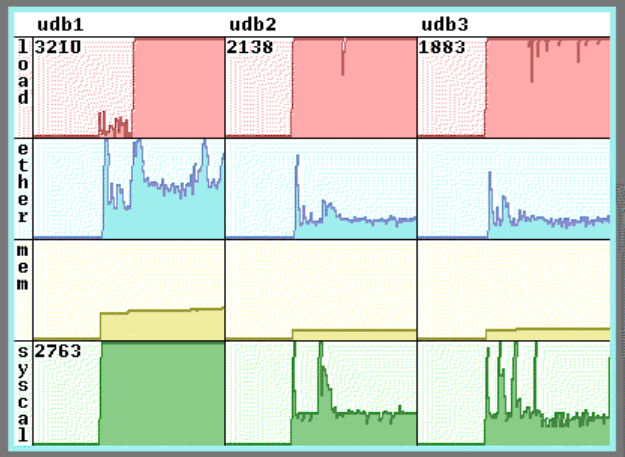
# Possible uses of PovRay:

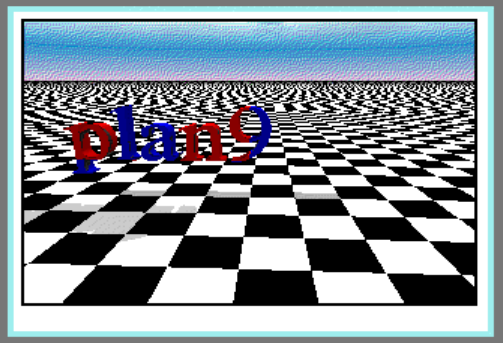
- Rendering farm (tested)
- Rendering single image on several machines without deploying any multiprocessor hardware (even though supported
- Some testing results (180 image animation):
  - povray -- 180 images, single machine: 9311.36u 26.78s 9465.01r
  - povray -- 180 images, 3 machines: 3403.79u 10.42s 3251.90r

# Drawbacks

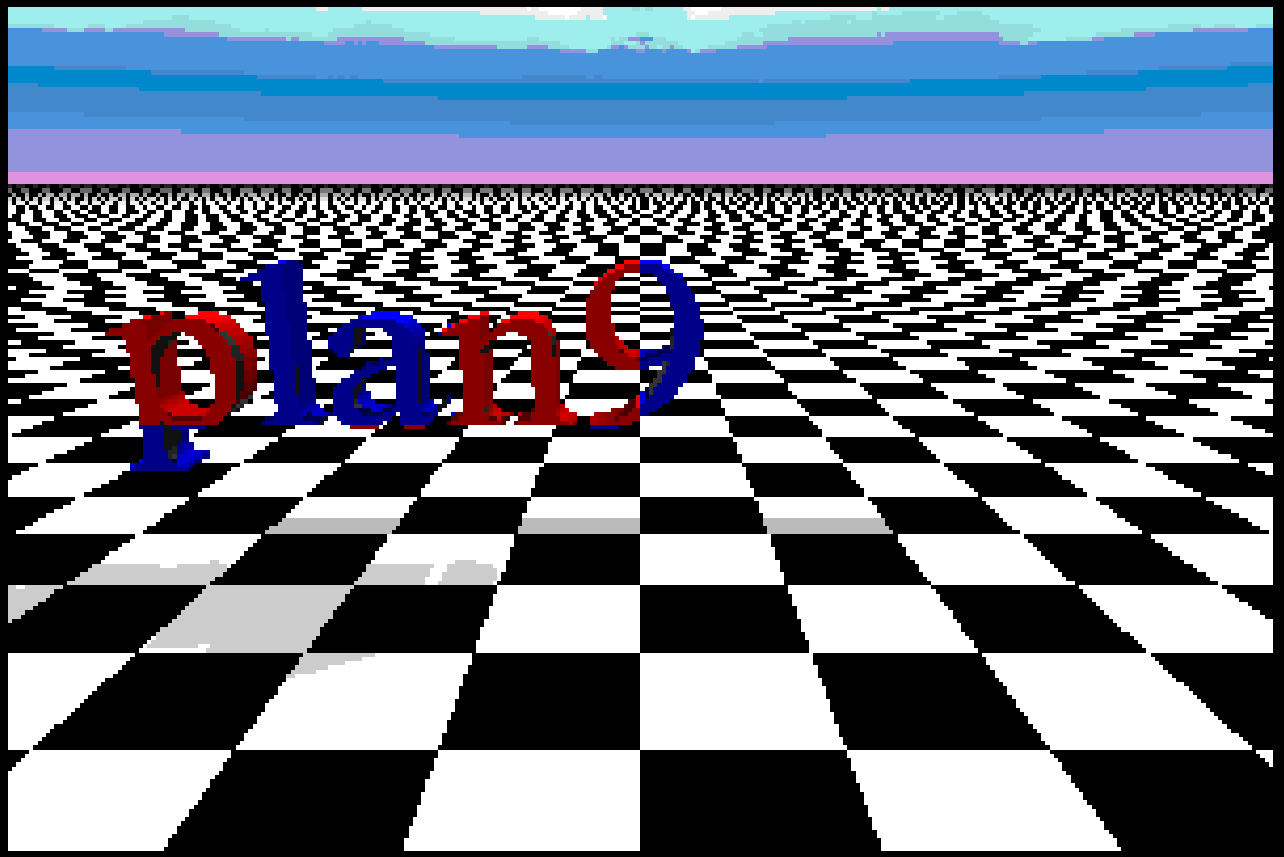
- Significantly higher network load then expected
- Very high load imposed on file server

```
cpu% cd povray_test
cpu% time run
starting udb2
starting udb3
starting local udb1
udb1: generating images from 0 to 60
udb1: creating image 0
udb2: generating images from 60 to 120
udb2: creating image 60
udb3: generating images from 120 to 180
udb3: creating image 120
udb1: creating image 1
```




```
cpu% togif < /dev/screen >scr.gif
```




Plan9 from Bell-Labs

# Bioinformatics

- Algorithm: FASTA, SSEARCH (version 3.3)
- Easily transformable into a distributed application (break database into smaller files and search separately – thanks to support from S. O'Hearn of PBI)
- Several possible distributed scenarios depending on where the database files are located – prompted by real problems encountered at PBI
- Searched the entire ecoli bacteria genome – bonus points added for fun

# Possible scenarios:

- Single database on remote fileserver
- Several small databases on remote server

(both viable for FASTA and SSEARCH)

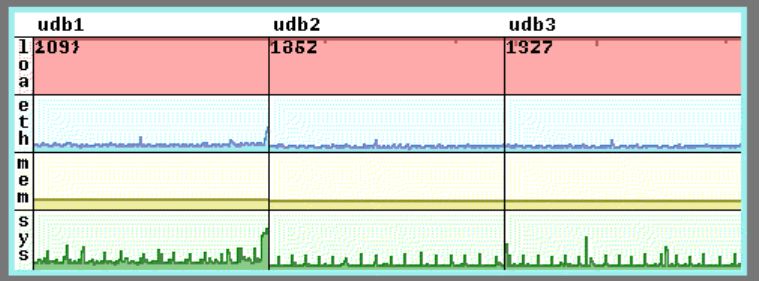
- Locally stored database files

```
 T. F. Smith and M. S. Waterman, (1981) J. Mol. Biol. 147:195-197;
 W.R. Pearson (1991) Genomics 11:635-650


***[fasta3/ssearch33] Query sequence undefined***

cpu%
```

```
cpu% hget http://lists.cse.psu.edu/archives/9f
ans.mbox/9fans.mbox > 9fans
cpu% togif < /dev/screen > scrshot.gif
```




```
cpu%
cpu%
cpu% time "  "
time: ": file does not exist
0.00u 0.00s 0.02r     " cat t1.seq
 SSEARCH searches a sequence databa
 using the Smith-Waterman algorithm
 version 3.3t08 Jan. 17, 2001
Please cite:
 T. F. Smith and M. S. Waterman, (1
 W.R. Pearson (1991) Genomics 11:63


***[fasta3/ssearch33] Query sequenc

cpu%
 SSEARCH searches a sequence database
 using the Smith-Waterman algorithm
 version 3.3t08 Jan. 17, 2001
Please cite:
 T. F. Smith and M. S. Waterman, (1981) J. Mol. Biol. 147:195-197;
 W.R. Pearson (1991) Genomics 11:635-650

 sequence may be truncated 80000 80000
 sequence truncated to 20000
 aa
 sequence truncated to 20000
 aa
@: 20000 aa
 secG 1694550..1694888
 vs  ecoli.nt.1.fas library
searching ecoli.nt.1.fas library
cpu% cat t1.seq | fasta3/ssearch33 -q @ ecoli.nt.1.fas -r udb1.results
 SSEARCH searches a sequence database
 using the Smith-Waterman algorithm
 version 3.3t08 Jan. 17, 2001
Please cite:
 T. F. Smith and M. S. Waterman, (1981) J. Mol. Biol. 147:195-197;
 W.R. Pearson (1991) Genomics 11:635-650

 sequence may be truncated 80000 80000
 sequence truncated to 20000
 aa
 sequence truncated to 20000
 aa
@: 20000 aa
 secG 1694550..1694888
 vs  ecoli.nt.1.fas library
searching ecoli.nt.1.fas library
```

```
li.nt library
g ecoli.nt library

tes/imp1
 t1.seq | ../fasta3/ssearch33 -q @ ecoli.nt.1.fas -r udb2.results
 searches a sequence database
he Smith-Waterman algorithm
 3.3t08 Jan. 17, 2001
ite:
mith and M. S. Waterman, (1981) J. Mol. Biol. 147:195-197;
 W.R. Pearson (1991) Genomics 11:635-650

 sequence may be truncated 80000 80000
 sequence truncated to 20000
 aa
 sequence truncated to 20000
 aa
@: 20000 aa
 secG 1694550..1694888
 vs  ecoli.nt.1.fas library
searching ecoli.nt.1.fas library
```

```
d-rwxrwxr-x M 74 bootes sys       0 Mar 12 12:01 tmp
cpu% cat t1.seq | ../
../: access permission denied
cpu% fasta3/ssearch33 -q @ ecoli.nt.1.fas -r udb3.results
cpu%
cpu%
cpu% cat t1.seq | ../fasta3/ssearch33 -q @ ecoli.nt.1.fas -r udb3.results
 SSEARCH searches a sequence database
 using the Smith-Waterman algorithm
 version 3.3t08 Jan. 17, 2001
Please cite:
 T. F. Smith and M. S. Waterman, (1981) J. Mol. Biol. 147:195-197;
 W.R. Pearson (1991) Genomics 11:635-650

 sequence may be truncated 80000 80000
 sequence truncated to 20000
 aa
 sequence truncated to 20000
 aa
@: 20000 aa
 secG 1694550..1694888
 vs  ecoli.nt.1.fas library
searching ecoli.nt.1.fas library
```

# Some results:

- Algorithm: FASTA (version 3.3)
- Easily transformable into a distributed application (break database into smaller files and search separately – thanks to support from S. O'Hearn of PBI)
- Several possible distributed scenarios depending on where the database files are located – prompted by real problems encountered at PBI
- Searched the entire ecoli bacteria genome – bonus points added for fun!

# Results (cont):

- Small database, networked:
  - 241.68u 0.24s 256.35r run fasta3/fasta33 network
- Small databases, local:
  - 245.01u 0.31s 248.65r run fasta3/fasta33 local

Difference comes from less context switching with local database and corresponds to 3-5 seconds delay in fetching a 4MB file over a 100baseT connection.

# Results (cont):

- Large database, local:
  - Scan time: 53.366 Display time: 473.390

Note: local database run time is not wall-clock execution as above, but a simple execution of the program. This does not increase significantly the result margin.

# Conclusion

- Bell-Labs research group have presented us with a one-of-a-kind research platform, which definitely has the possibility of attracting students and professors' attention
- Maybe not a commercially viable product, but a very good learning tool
- Need to emphasize on the lower learning curve with regards to deep internals. Code very easy to read
- Ability to share ideas and learn from some of the groundbreaking personalities in this field is an enormous ego boost

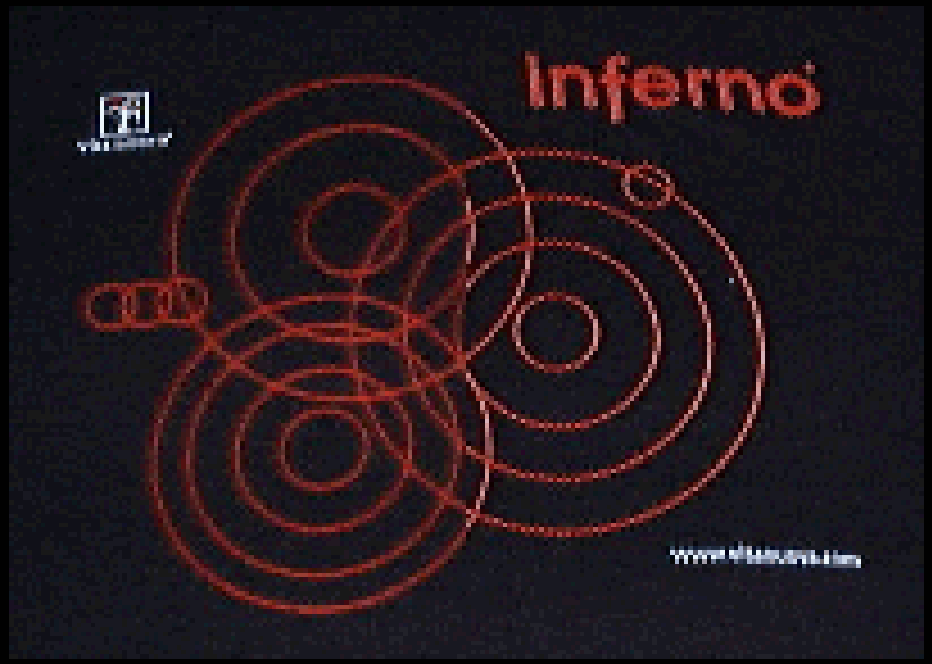
# Would not have been be possible without:

Tony Kusalik (CS)

Stephen O'Hearn (PBI)

Dave Bocking (CS)

Cary Bernath and Greg Oster (CS)




Questions?