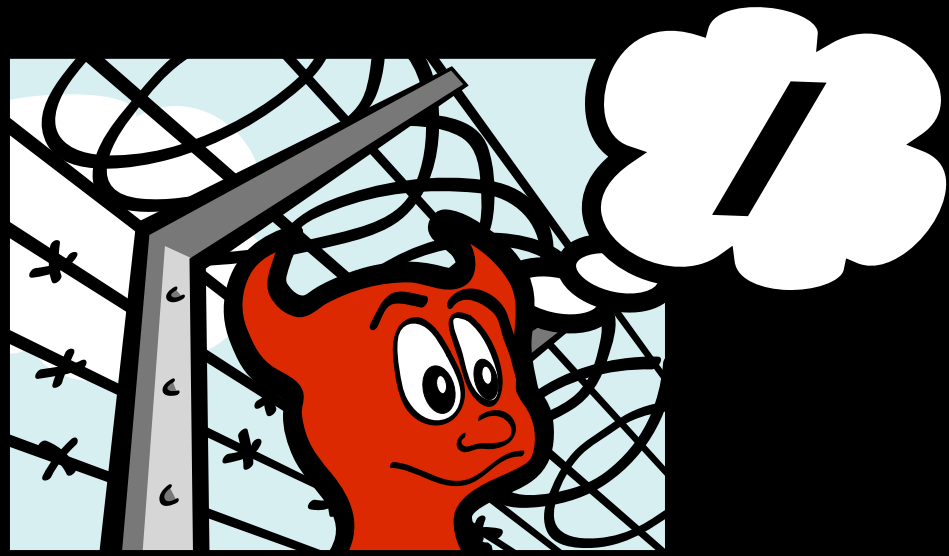


jail(8)

Hacking UNIX with FreeBSD jail(8), Secure Virtual Servers

Presentation for DefCon 14, by Isaac Levy, (.ike)

.ike Context

- I have used jails extensively for web application servers and software development purposes
- the methodology I’m presenting here is attempting to be ‘stock’ UNIX (no ‘ike-specific’ magic formulas)
- I am not a jail author, no commit bit...

Warranty / Announcement

- I’ll be out and about later if anyone has more complex questions or strategies they want to discuss
- I’m *trying* to stick to classic UNIX process and ideas, and ‘stock’ methodology (no ike-specific magic)
- I’m assuming you all know your way around various *NIX Operating Systems

scale, patterns, complexity
(a big picture exercise)

Film: Powers of Ten, 1977, Charles and Ray Eames

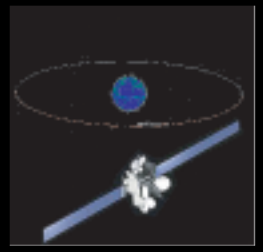
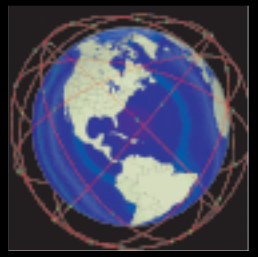
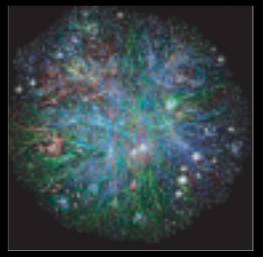
http://www.powersof10.com/
http://www.youtube.com/watch?v=4i6B7HzijSo

Film: Powers of Ten, 1977, Charles and Ray Eames

http://www.powersof10.com/
http://www.youtube.com/watch?v=4i6B7HzijSo

Film: Powers of Ten, 1977, Charles and Ray Eames

http://www.powersof10.com/
http://www.youtube.com/watch?v=4i6B7HzijSo



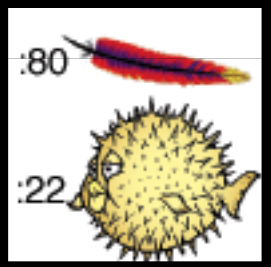













BSD UNIX

Internet universe, (according to ike, today.)

CryptoKids™ America's Future Codemakers & Codebreakers

http://www.nsa.gov/kids/home.cfm

Google

America's CryptoKids™

Future Codemakers & Codebreakers

CHARACTER BIOGRAPHIES

STUDENT RESOURCES

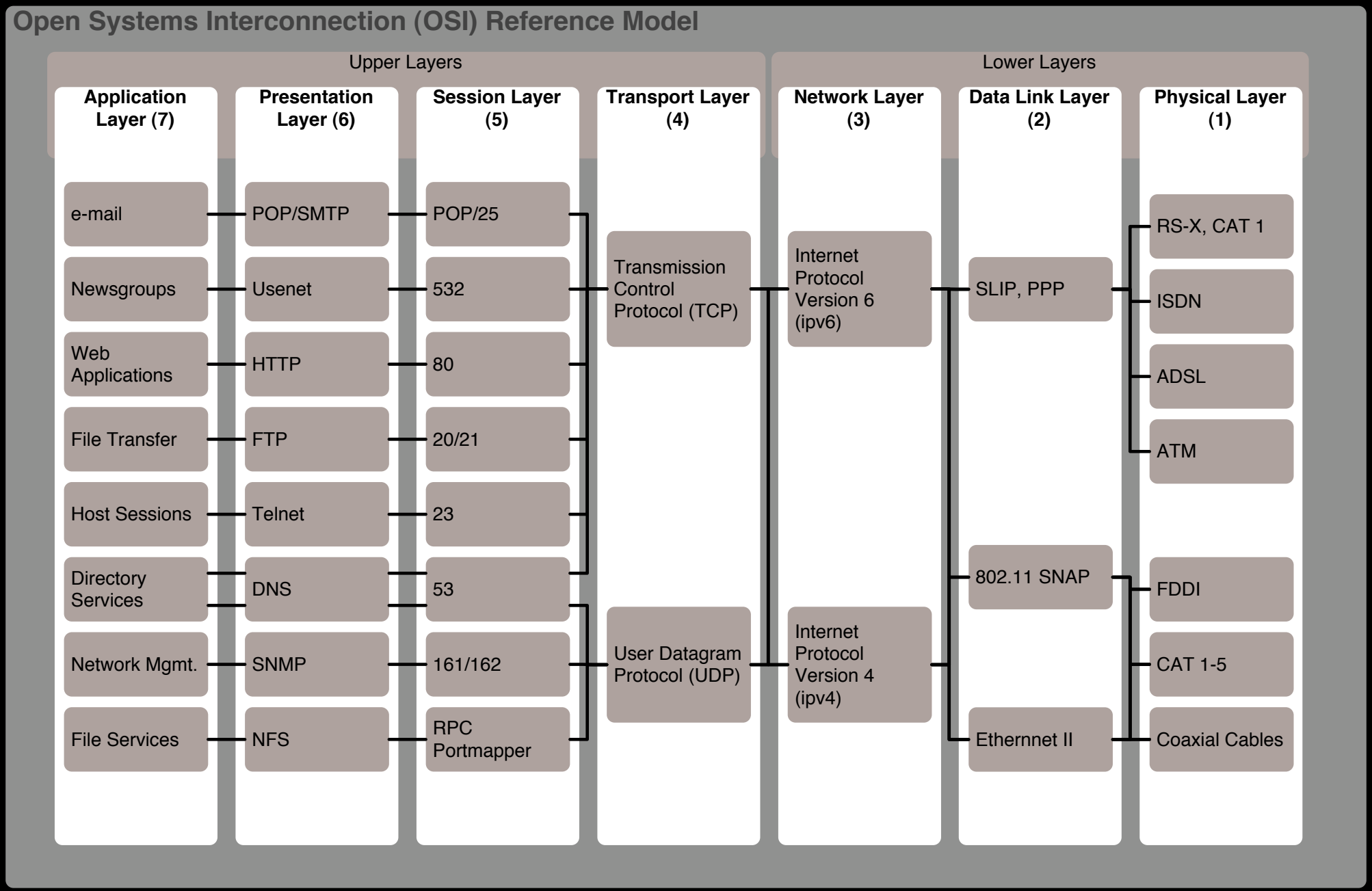
GAMES & ACTIVITIES

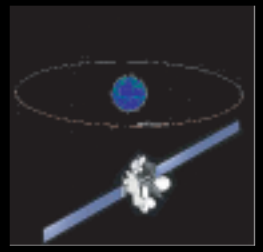
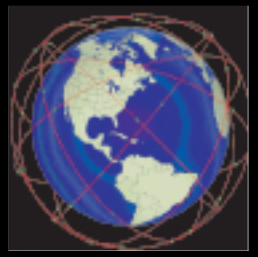
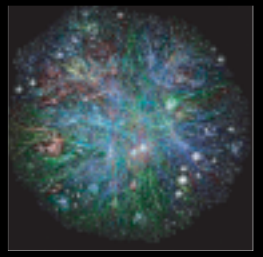
CODES & CIPHERS

GAMES & ACTIVITIES

Codes & Ciphers | Games & Activities | Student Resources | Character Biographies

NSA.gov | Links | Parents & Teachers | Trademark | Accessibility | Privacy & Security


Open Systems Interconnection (OSI) Reference Model
Upper Layers
Lower Layers
Application Layer (7)
Presentation Layer (6)
Session Layer (5)
Transport Layer (4)
Network Layer (3)
Data Link Layer (2)
Physical Layer (1)
e-mail
POP/SMTP
POP/25
Newsgroups
Usenet
532
Web Applications
HTTP
80
File Transfer
FTP
20/21
Host Sessions
Telnet
23
Directory Services
DNS
53
Network Mgmt.
SNMP
161/162
File Services
NFS
RPC Portmapper
Transmission Control Protocol (TCP)
User Datagram Protocol (UDP)
Internet Protocol Version 6 (ipv6)
Internet Protocol Version 4 (ipv4)
SLIP, PPP
802.11 SNAP
Ethernnet II
RS-X, CAT 1
ISDN
ADSL
ATM
FDDI
CAT 1-5
Coaxial Cables



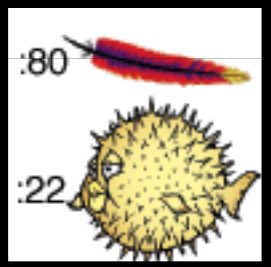








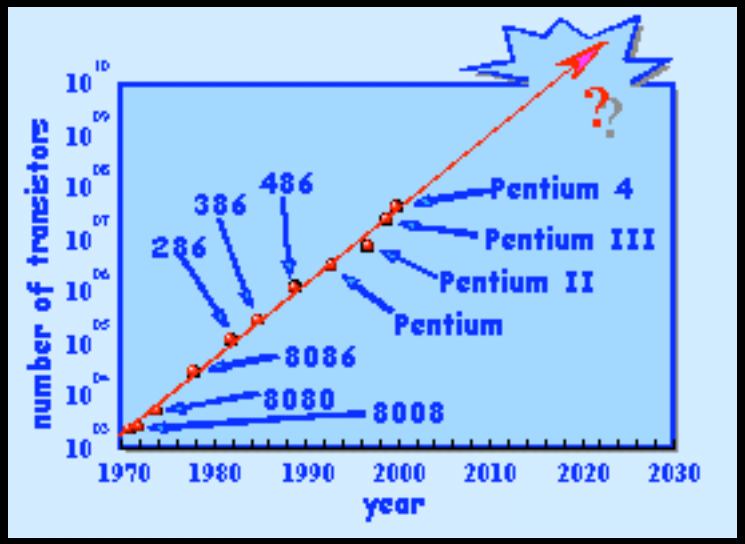


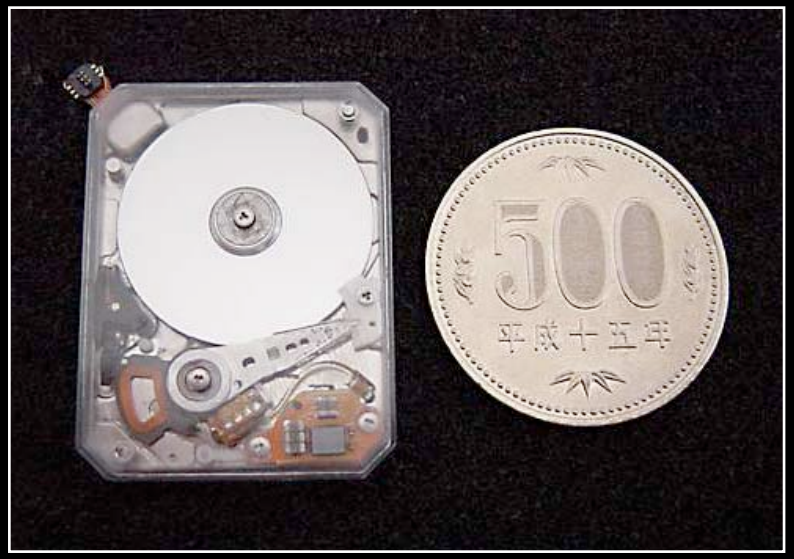


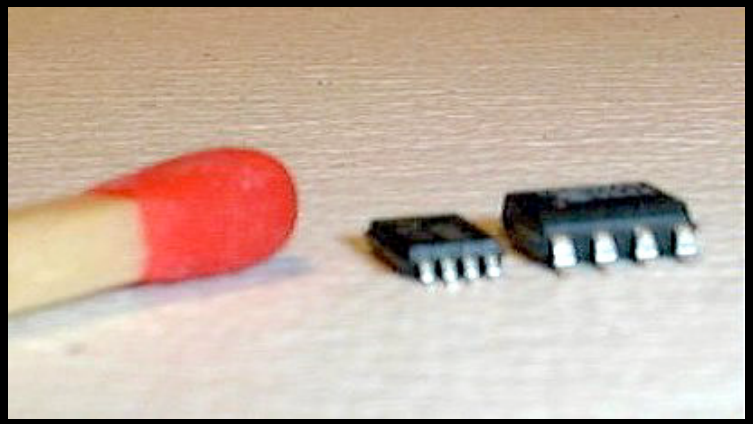

BSD UNIX

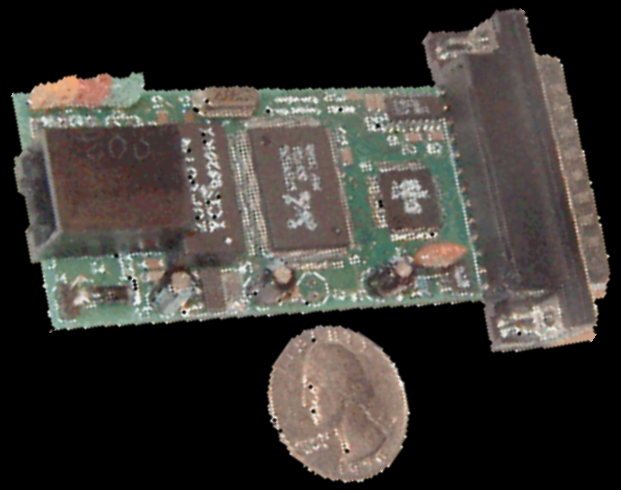
Internet universe, (according to ike, today.)

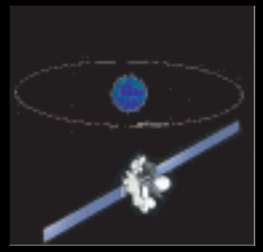
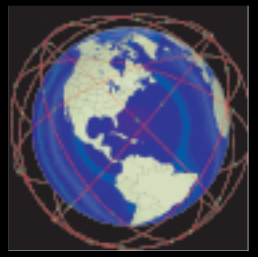
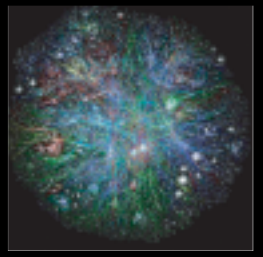
yadda yadda


number of transistors
year
1970
1980
1990
2000
2010
2020
2030
?
Pentium 4
Pentium III
Pentium II
Pentium
486
386
286
8086
8080
8008


500







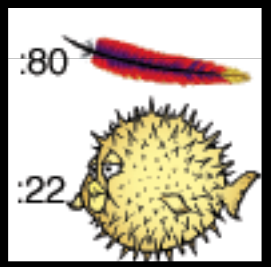








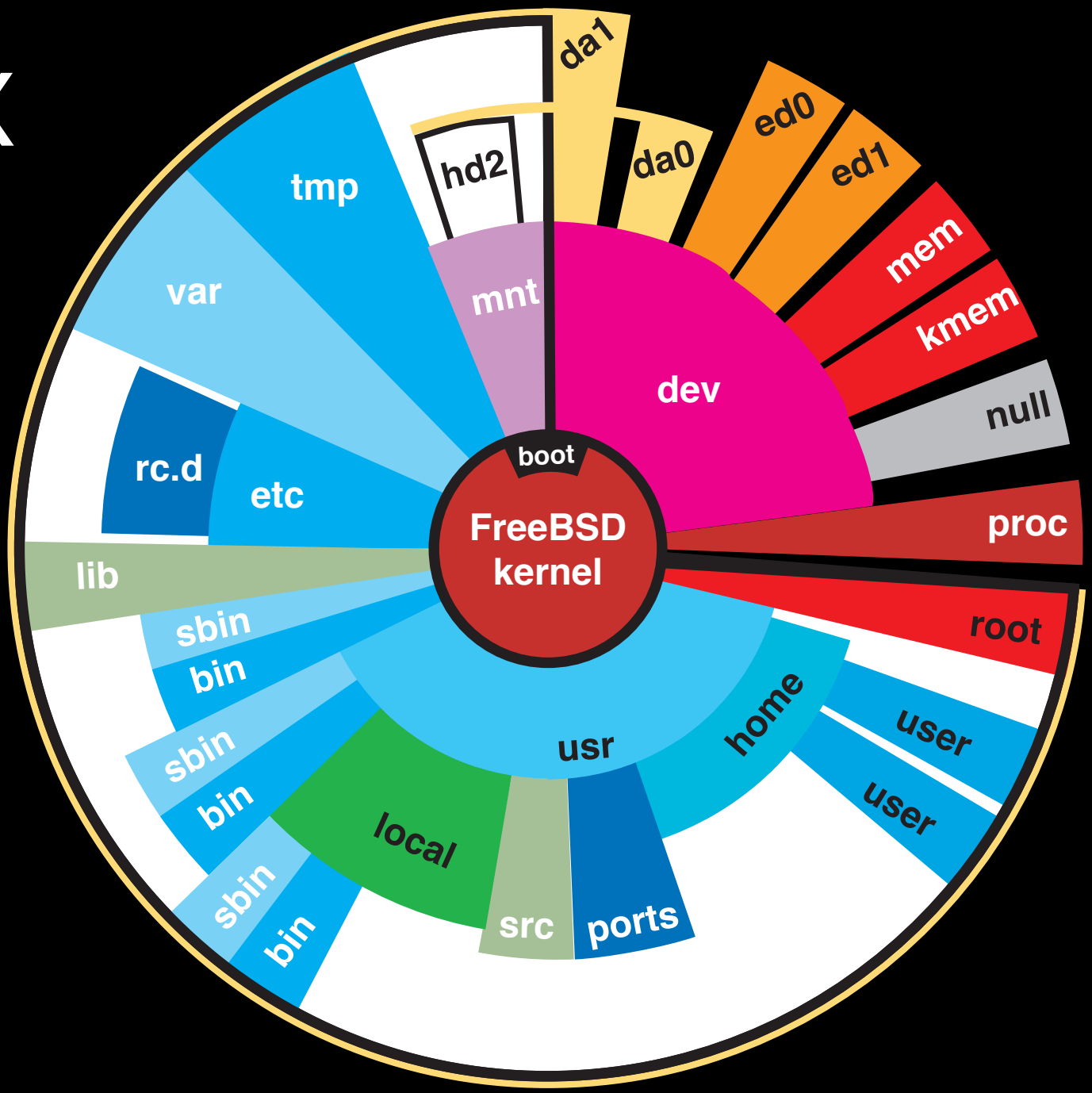




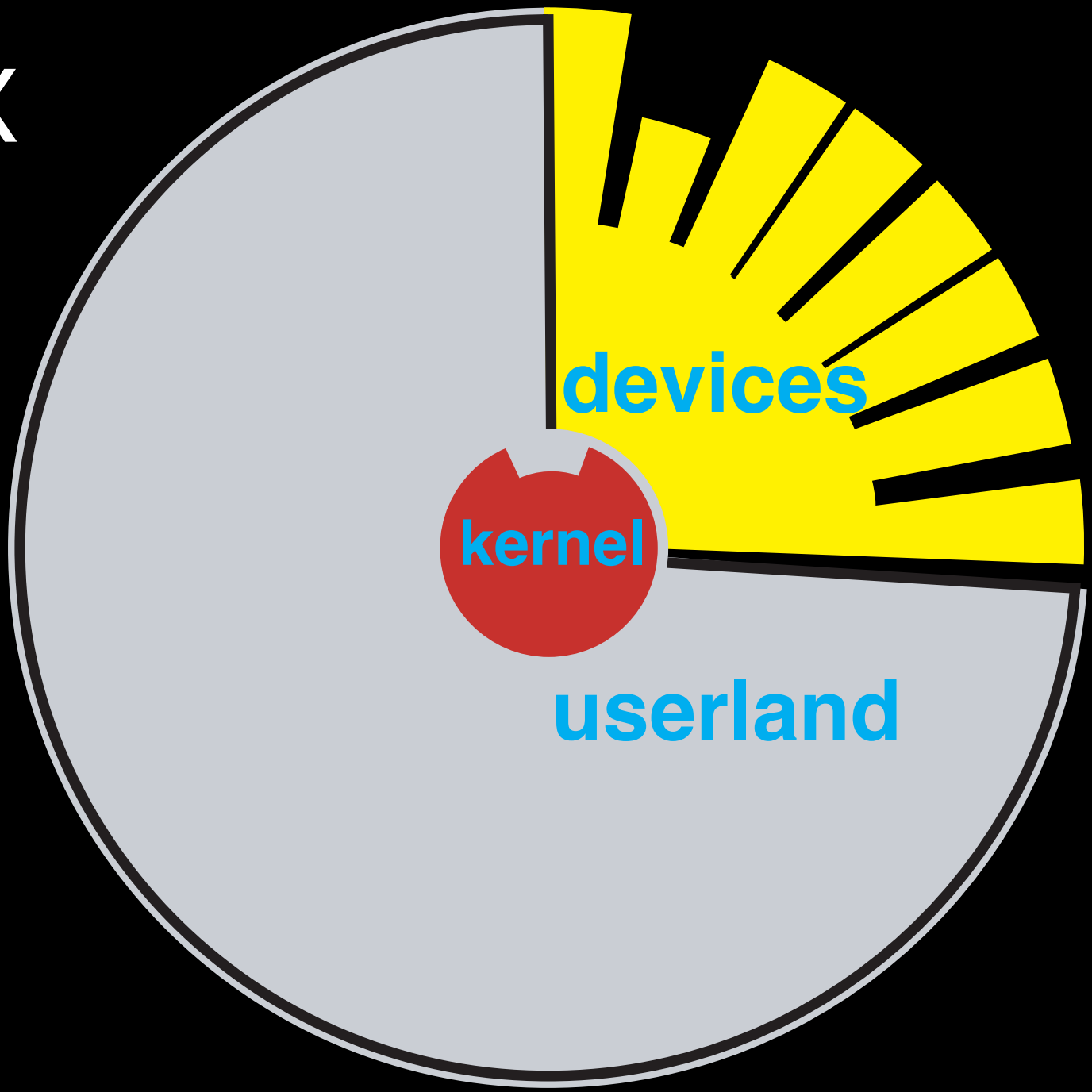

BSD UNIX

Internet universe, (according to ike, today.)

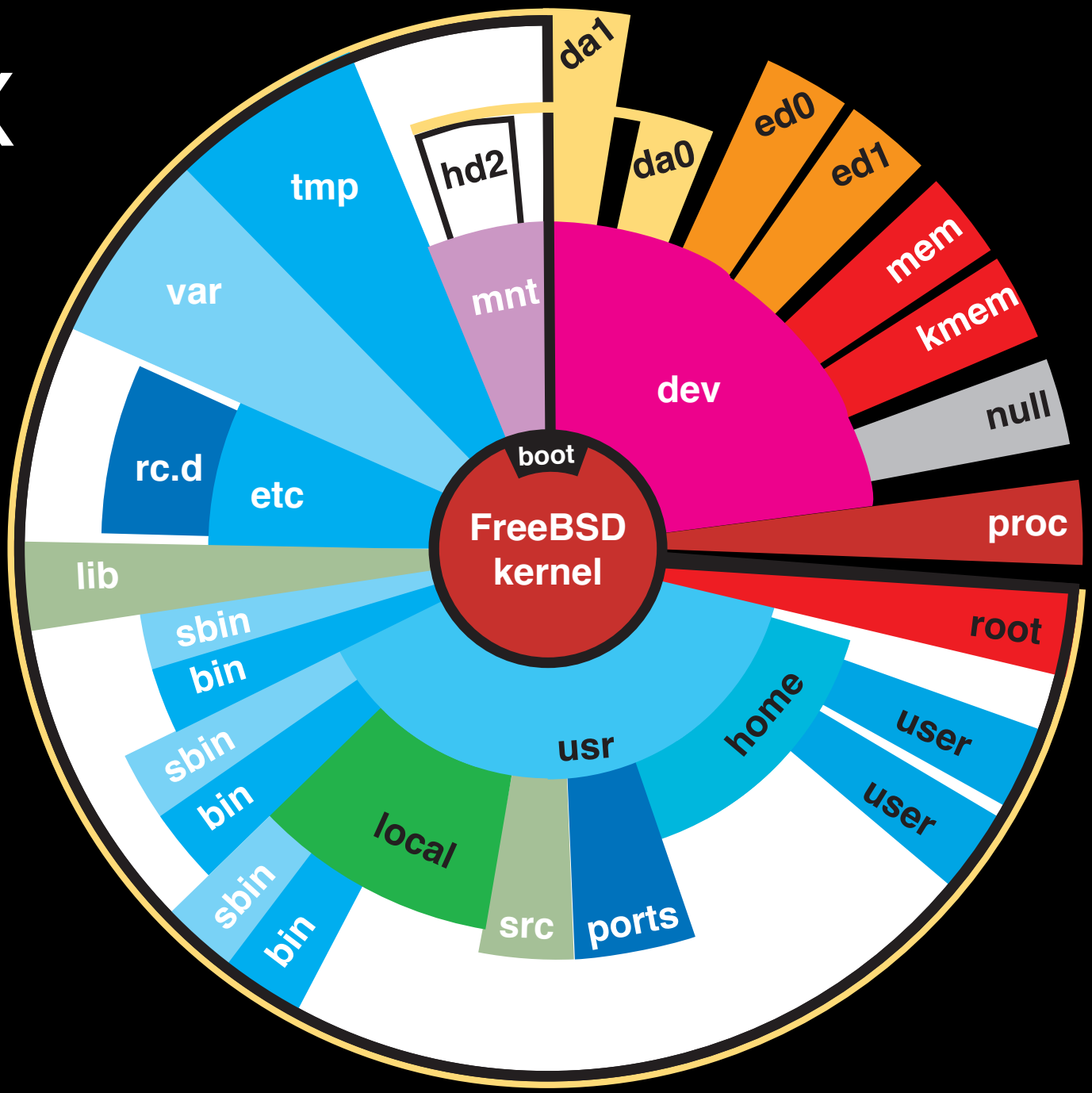
UNIX


FreeBSD kernel
boot
dev
proc
root
usr
home
user
user
local
src
ports
bin
sbin
bin
sbin
bin
sbin
lib
etc
rc.d
var
tmp
mnt
hd2
da1
da0
ed0
ed1
mem
kmem
null

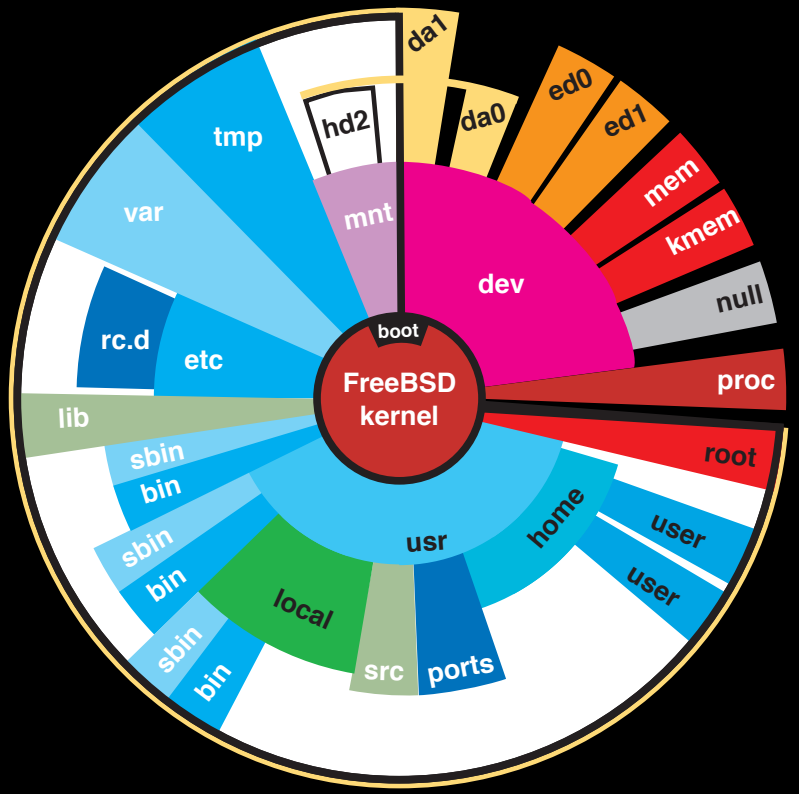
UNIX



UNIX


FreeBSD kernel
boot
dev
proc
root
usr
home
user
user
local
src
ports
bin
sbin
bin
sbin
bin
sbin
lib
etc
rc.d
var
tmp
mnt
hd2
da1
da0
ed0
ed1
mem
kmem
null

UNIX



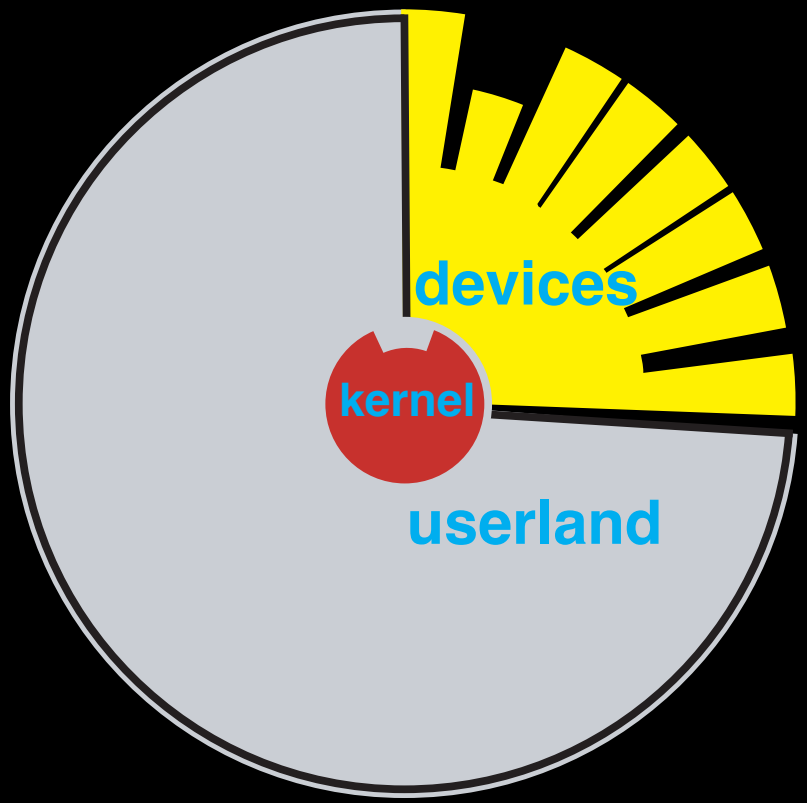
Spiral Galaxy

Spiral Galaxy NGC 1232

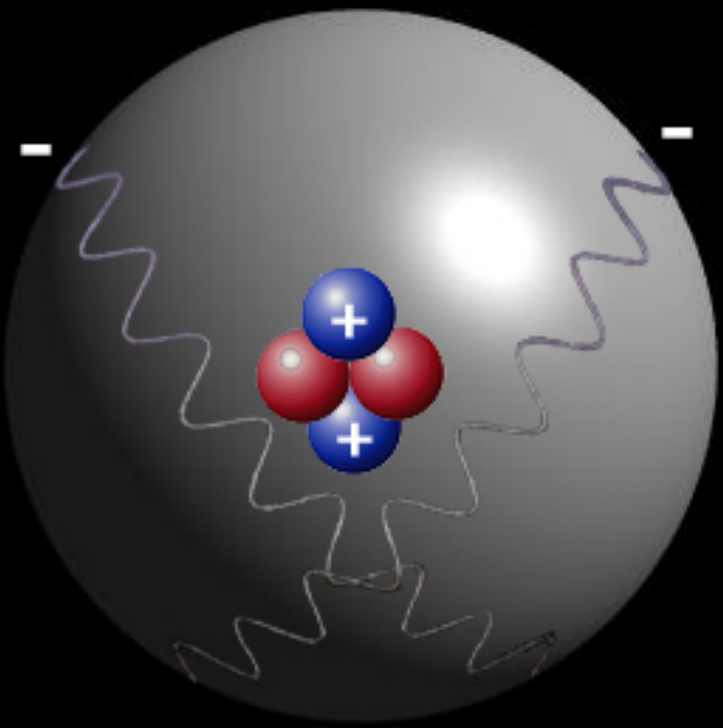
Our world is complex

(thx Dan Geer & ShmooCon).

UNIX

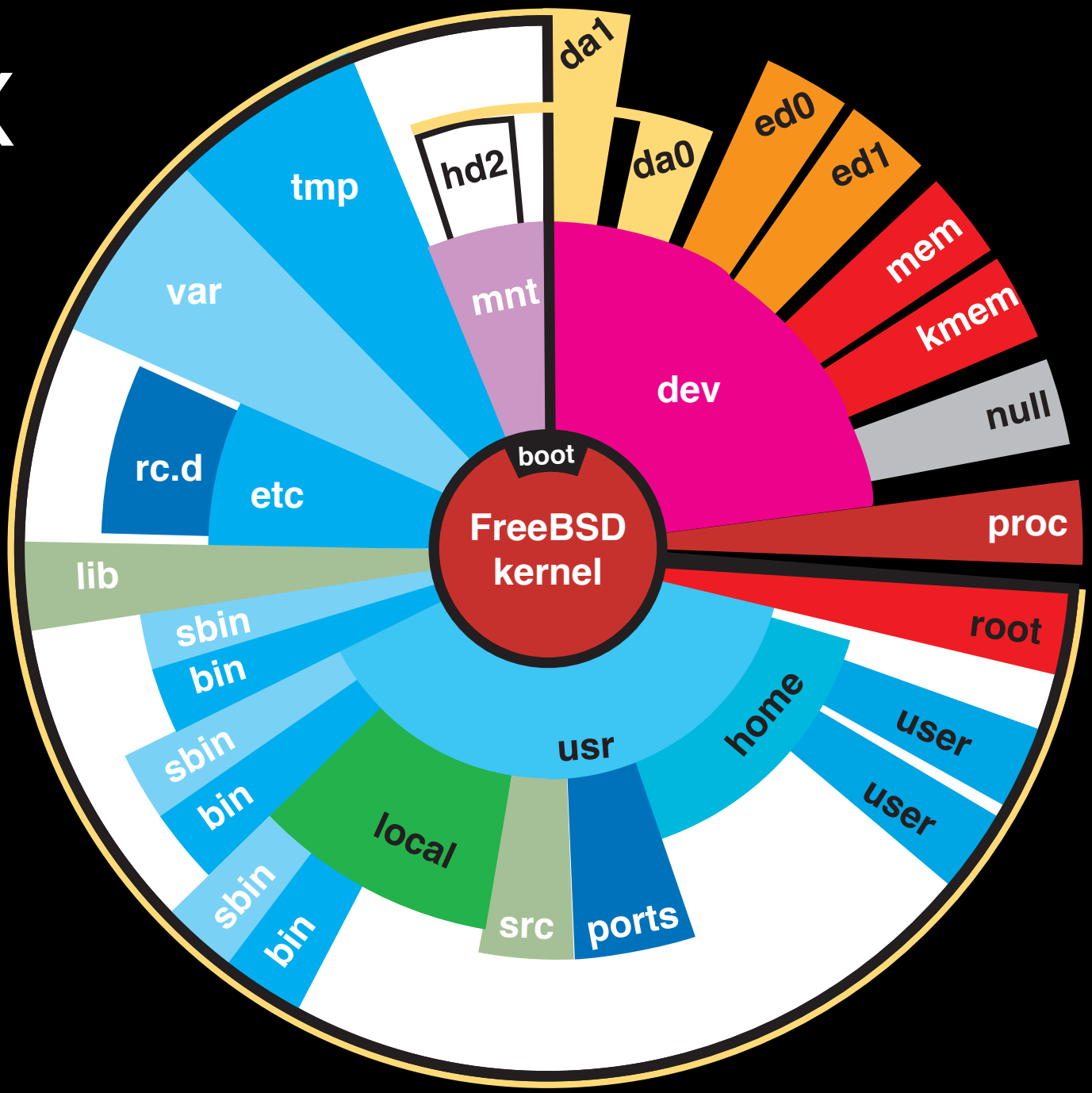


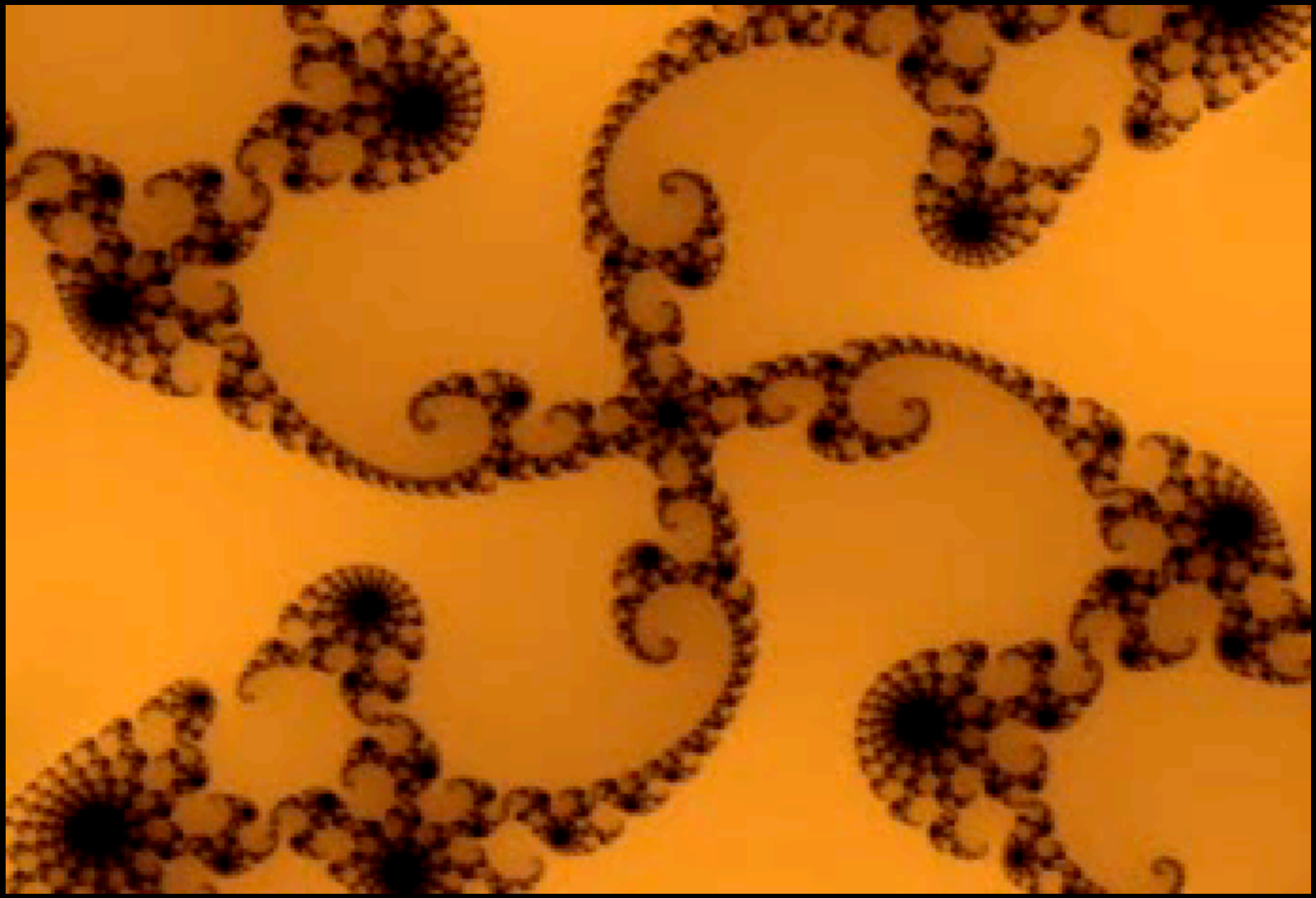
Helium Atom



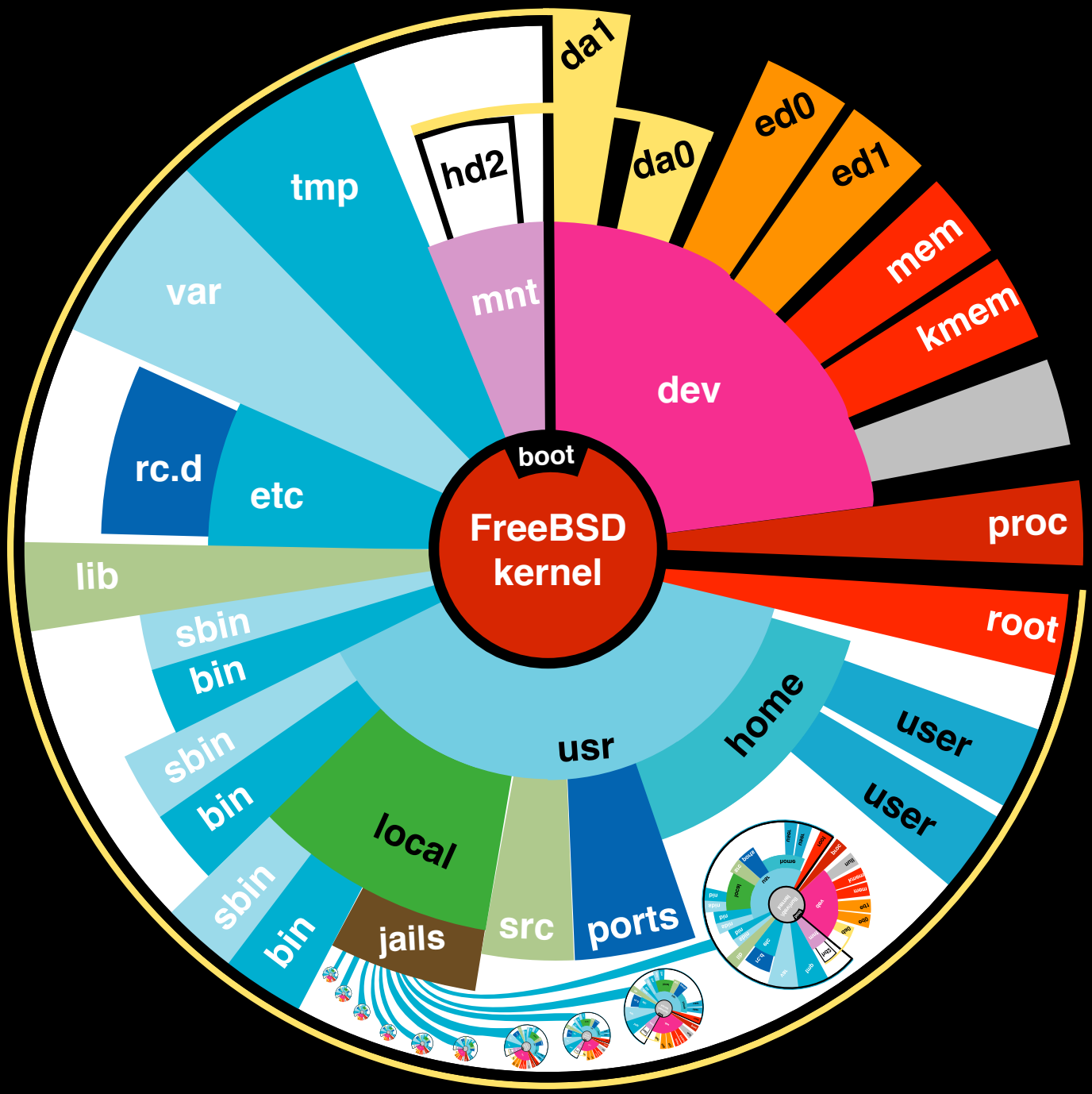
Our world is simple too...

UNIX

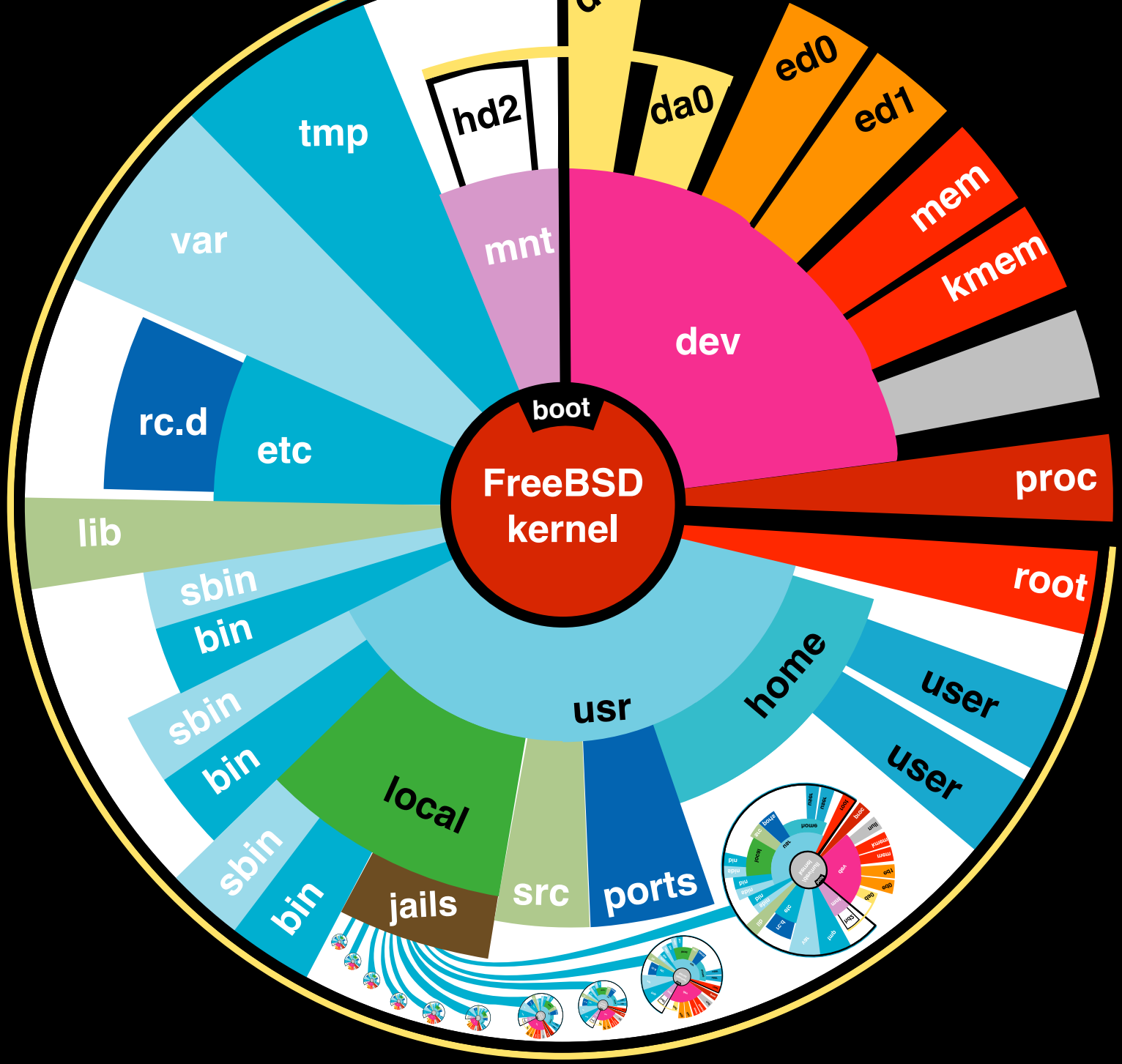

FreeBSD kernel
boot
dev
proc
root
usr
home
user
user
local
src
ports
bin
sbin
bin
sbin
bin
sbin
lib
etc
rc.d
var
tmp
mnt
hd2
da1
da0
ed0
ed1
mem
kmem
null

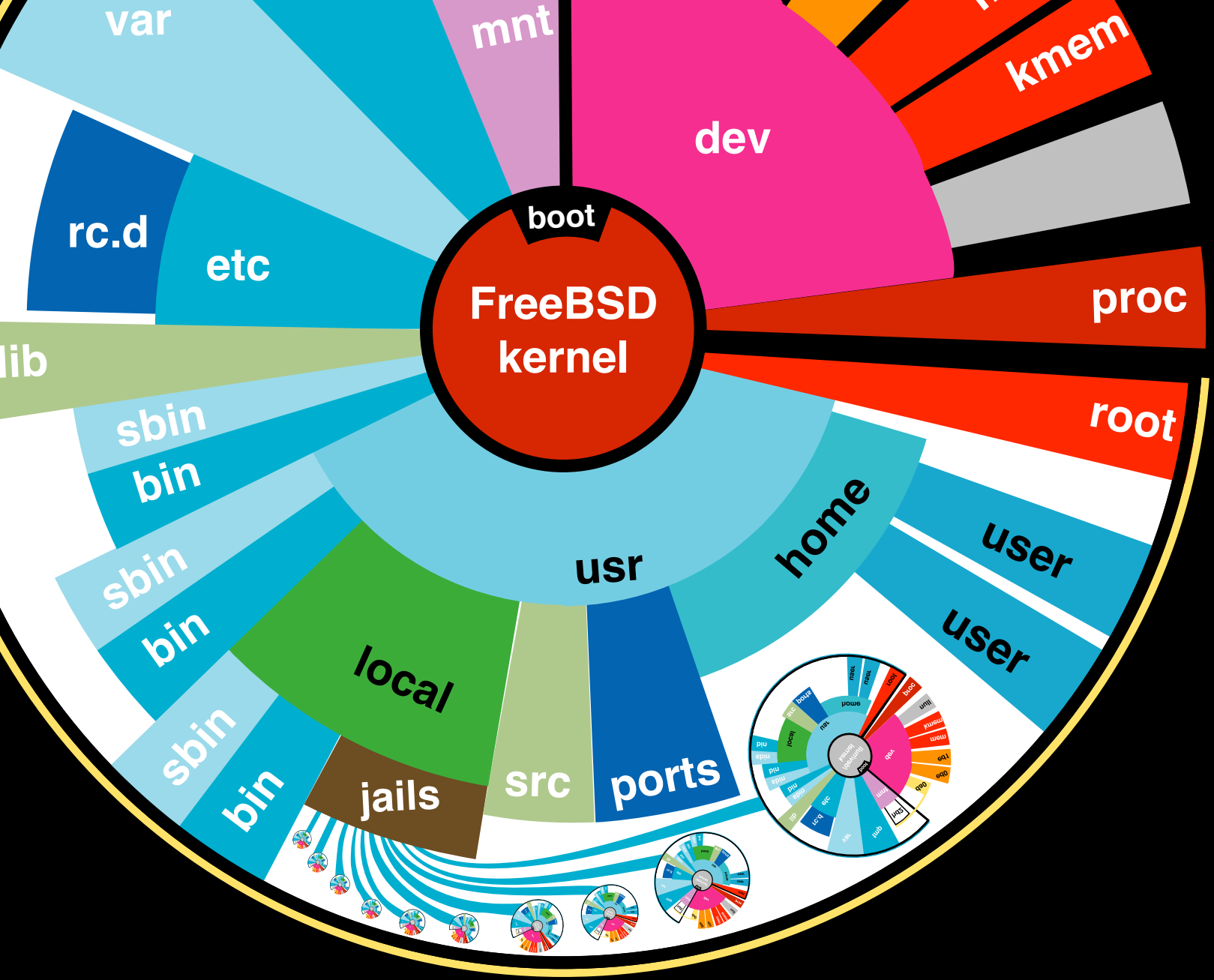


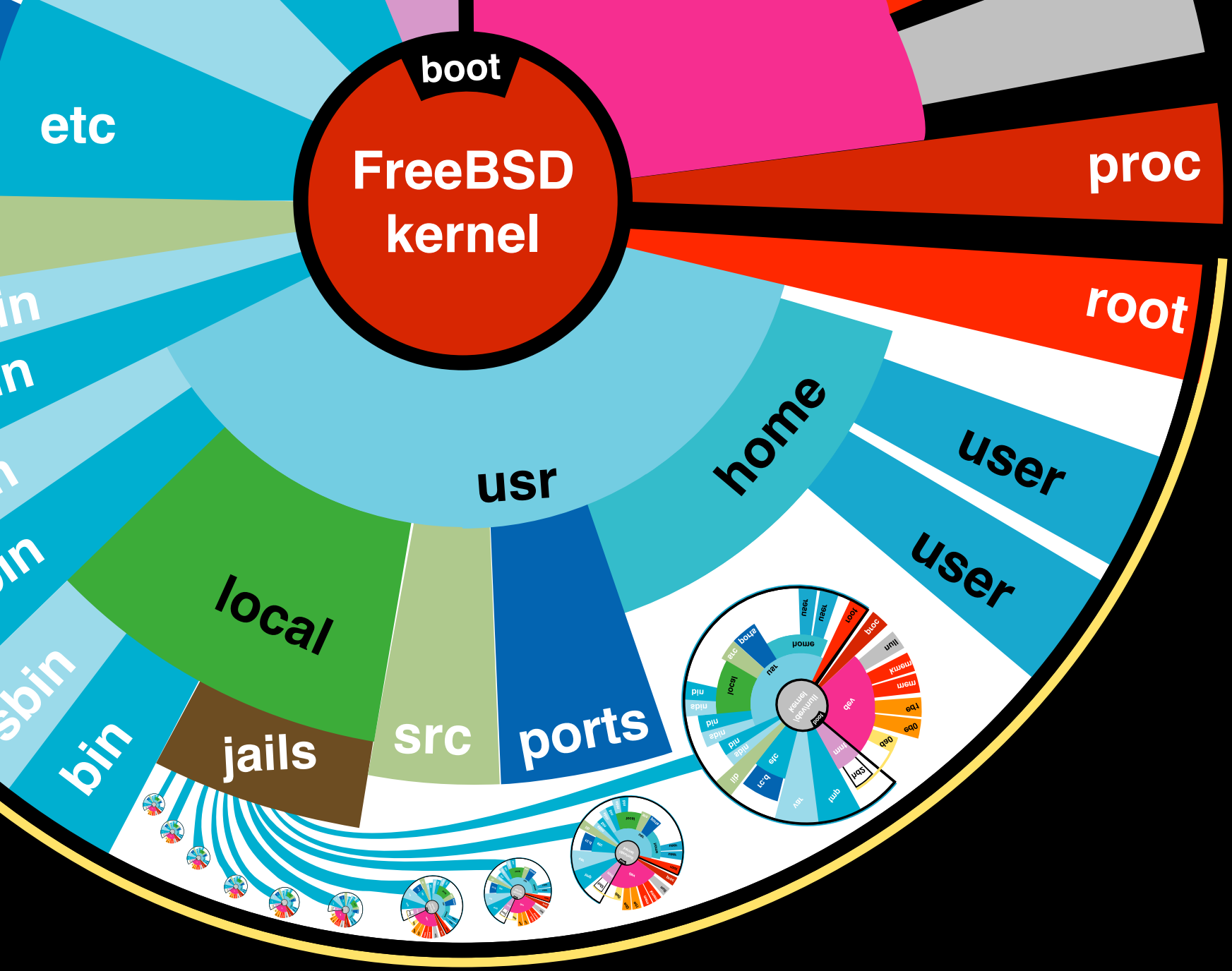
Mandelbrot Fractal - Julia set

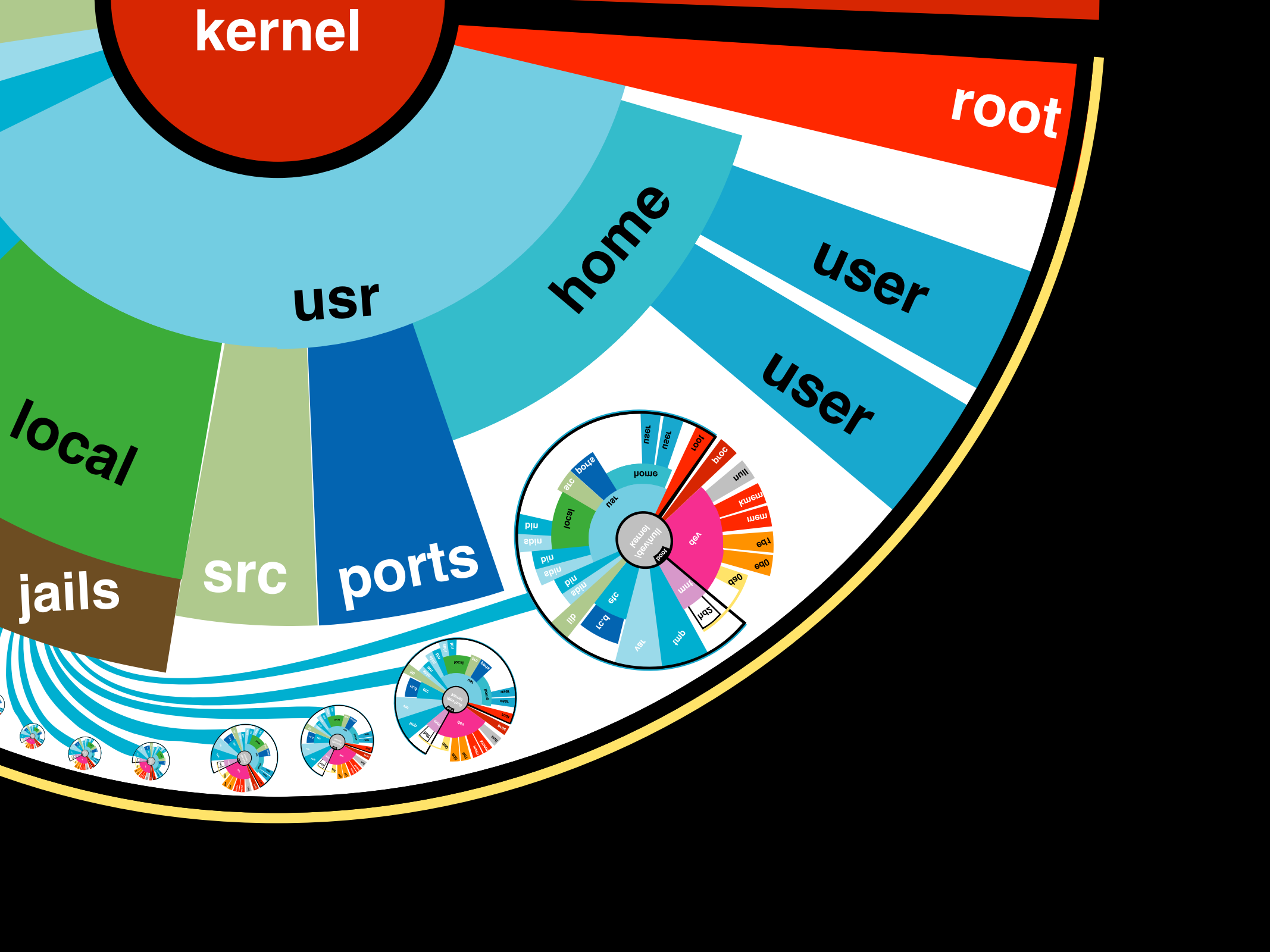


virtual
UNIX's


FreeBSD kernel
boot
dev
proc
root
home
usr
mnt
tmp
var
etc
rc.d
lib
sbin
bin
sbin
bin
sbin
bin
local
jails
src
ports
user
user
hd2
da0
ed0
ed1
mem
kmem


var
mnt
dev
kmem
rc.d
etc
boot
FreeBSD
kernel
proc
lib
root
sbin
bin
usr
home
user
user
sbin
bin
local
src
ports
sbin
bin
jails


boot
FreeBSD kernel
etc
proc
root
home
usr
user
user
local
jails
src
ports
bin
sbin


kernel
root
usr
home
user
user
local
src
ports
jails

root

usr

home

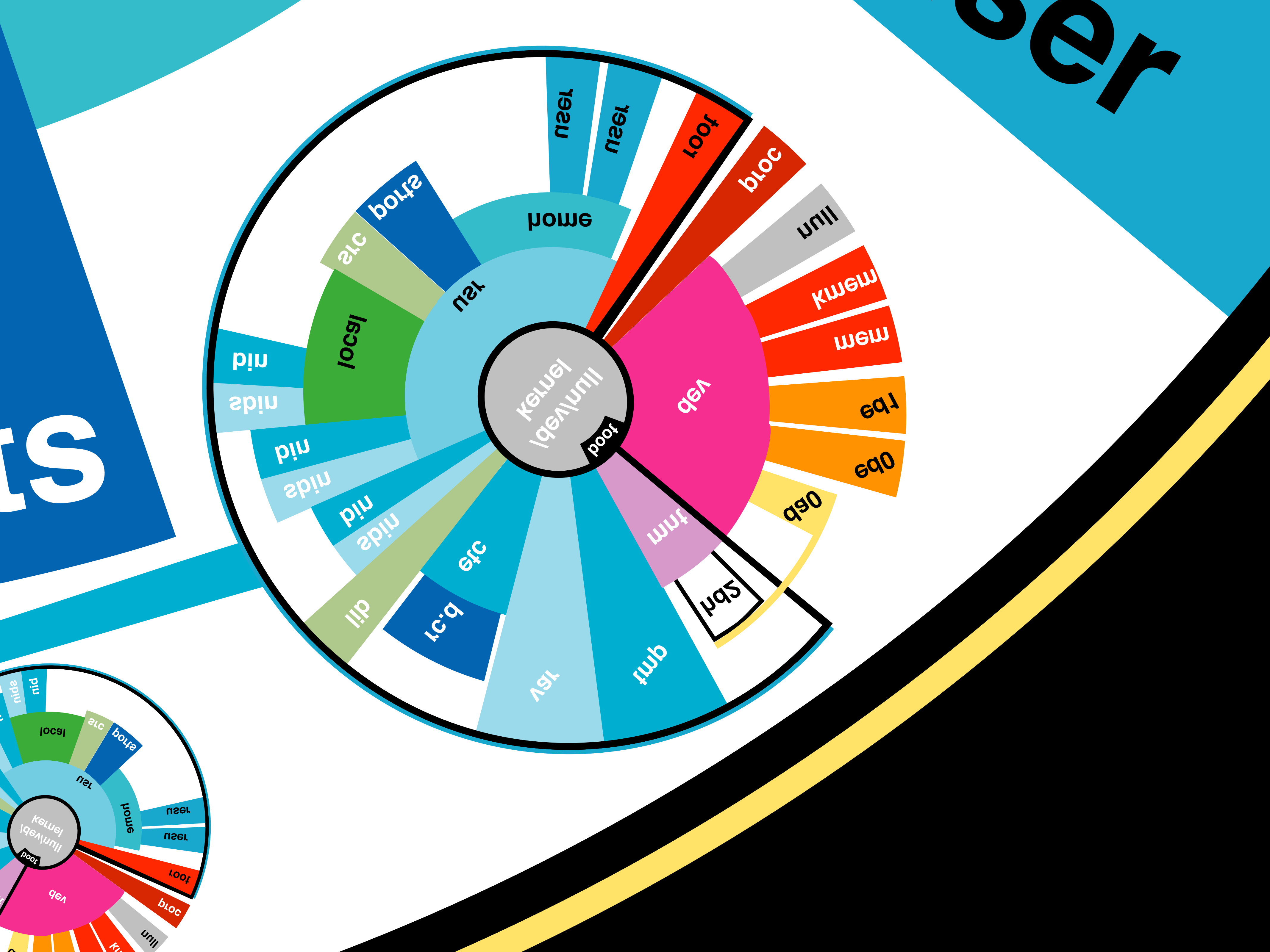
user

user

src

ports


user
orts
/dev/null
kernel
boot
dev
usr
home
user
user
root
proc
null
kmem
mem
local
src
ports
bin
sbin
bin
sbin
bin
sbin
lib
etc
rc.d
var
tmp
mnt



/dev/null
kernel
boot
dev
home
usr
user
user
root
proc
null
kmem
mem
ed1
ed0
dsp
mnt
sd2
tmp
var
rc.d
etc
lib
sbin
bin
local
src
ports

virtual UNIX's

The UNIX Time-Sharing System*

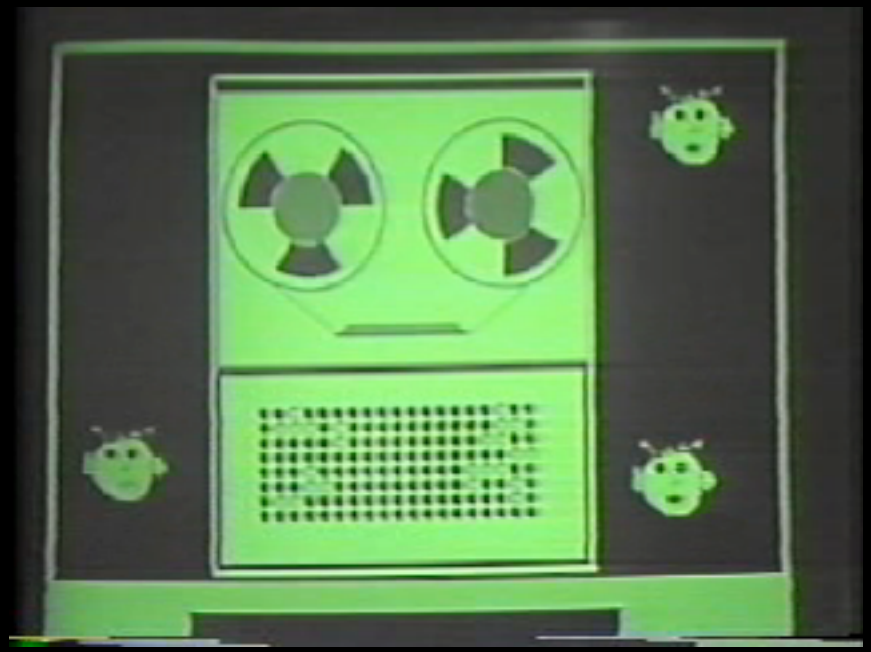
D. M. Ritchie and K. Thompson

ABSTRACT

Unix is a general-purpose, multi-user, interactive operating system for the larger Digital Equipment Corporation PDP-11 and the Interdata 8/32 computers. It offers a number of features seldom found even in larger operating systems, including

i
A hierarchical file system incorporating demountable volumes,
ii
Compatible file, device, and inter-process I/O,
iii
The ability to initiate asynchronous processes,
iv
System command language selectable on a per-user basis,
v
Over 100 subsystems including a dozen languages,
vi
High degree of portability.

This paper discusses the nature and implementation of the file system and of the user command interface.



You get the idea-

So what real-world contexts warrant virtualizing the ENTIRE operating system?

external
security
threats

development
messes

Mutually Untrusted Users

Mutually Untrusted Users

Mutually Untrusted Users

Mutually Untrusted Users

telnet forever!

Mutually Untrusted Users

login:admin
pass:love

Moron: Using an electric tool, in a pool, with a metal ladder.
Stupid Moron: Standing barefoot on the ladder while doing it.

su
24/7 ?

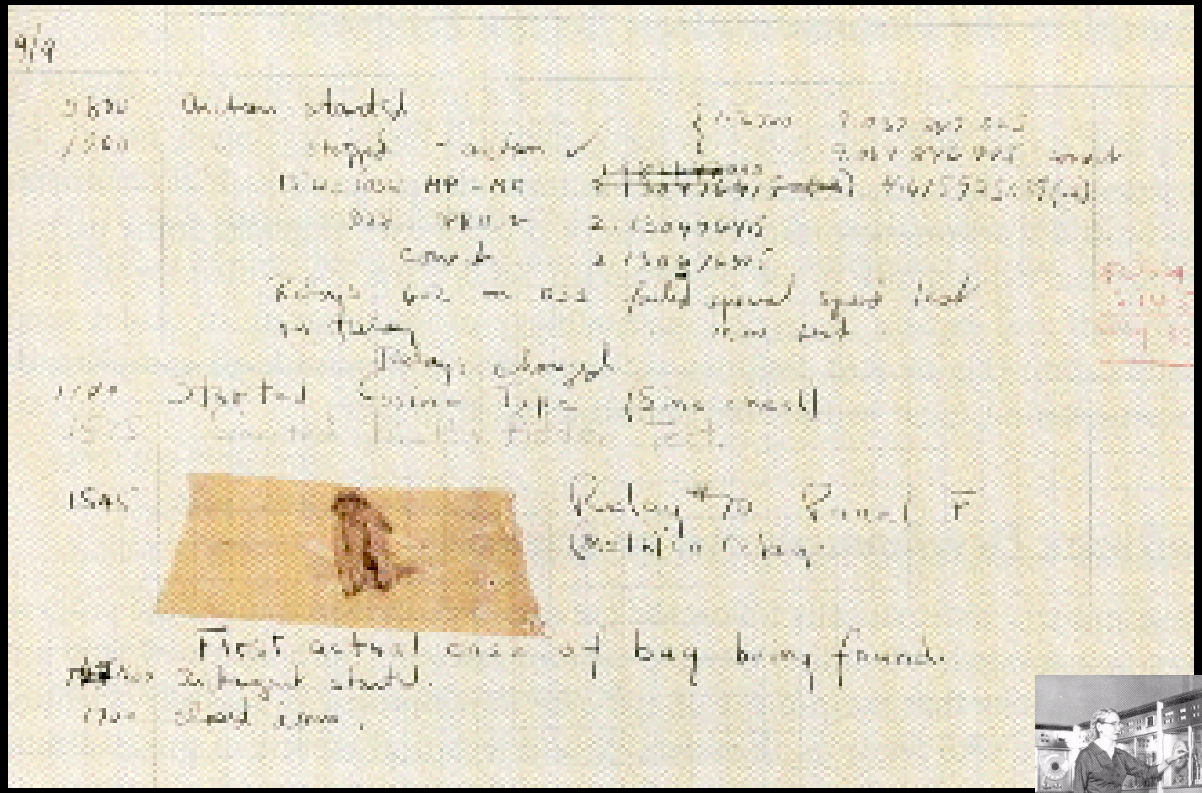
Mutually Untrusted Users

Mutually Untrusted Users

Mutually Untrusted Users

You run
WHAT as
CGI?

Mutually Untrusted Users



programs are users too...

Mutually Untrusted Users

muscle memory kills!


SOME ASSHOLE TALKING
ON HIS CELL PHONE
GOT CREAMED
9304

Harmony.

Once upon a time,
wasn't UNIX *fun*?

http://mckusick.com/beastie/

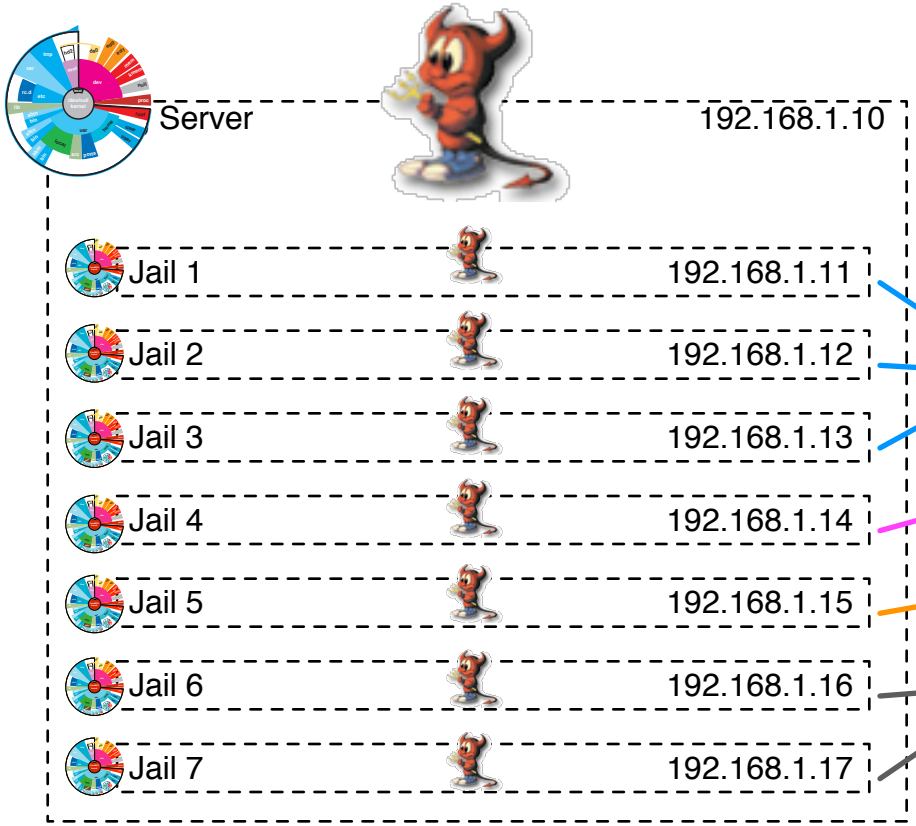
maintaining old junk?

Rack full of stuff Example:

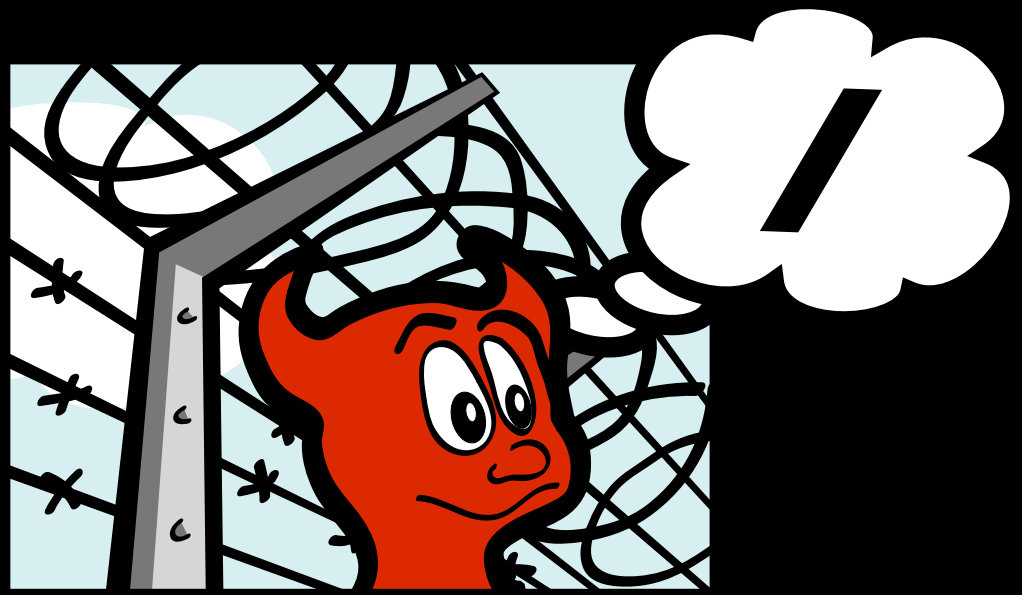
- 3 webservers
- 1 local-use dns cache
- fileserver (for 2 people)
- 2 dev servers

jail(8)!

Rack full of stuff ,
becomes 1u server!



3 webservers
1 local-use dns cache
fileserver (for 2 people)
2 dev servers



jail(8)

Definitions

- what is a jail(8):
  - a user space utility, like ifconfig(8)
  - produces a virtual system image
  - process tree based
- what is jail(2):
  - a system call to imprison a process
  - it calls chroot and attaches to IP
  - a very few lines of source code!

Definitions

- what jail is ***not***:
  - it is **not** a classical machine emulator
  - it is **not** chroot (‘jail’ vocabulary is commonly misused with other *NIX cultures)

Great Uses for **jail(8)**

- hardware resource sharing, an entire OS can be dedicated to a given service
- securely separate untrusted users/processes
- learning/development/testing/hacking
- insane high availability possibilities
- honeypots
- highly vulnerable network services

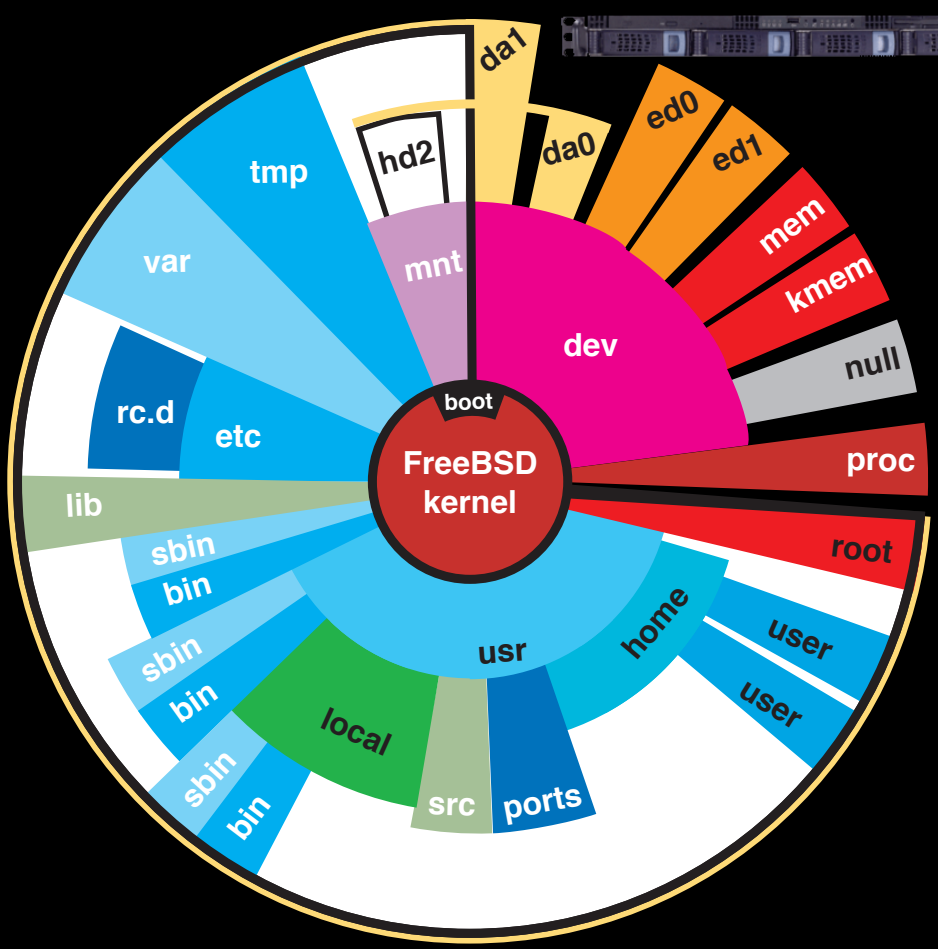
Poor Uses for **jail(8)**

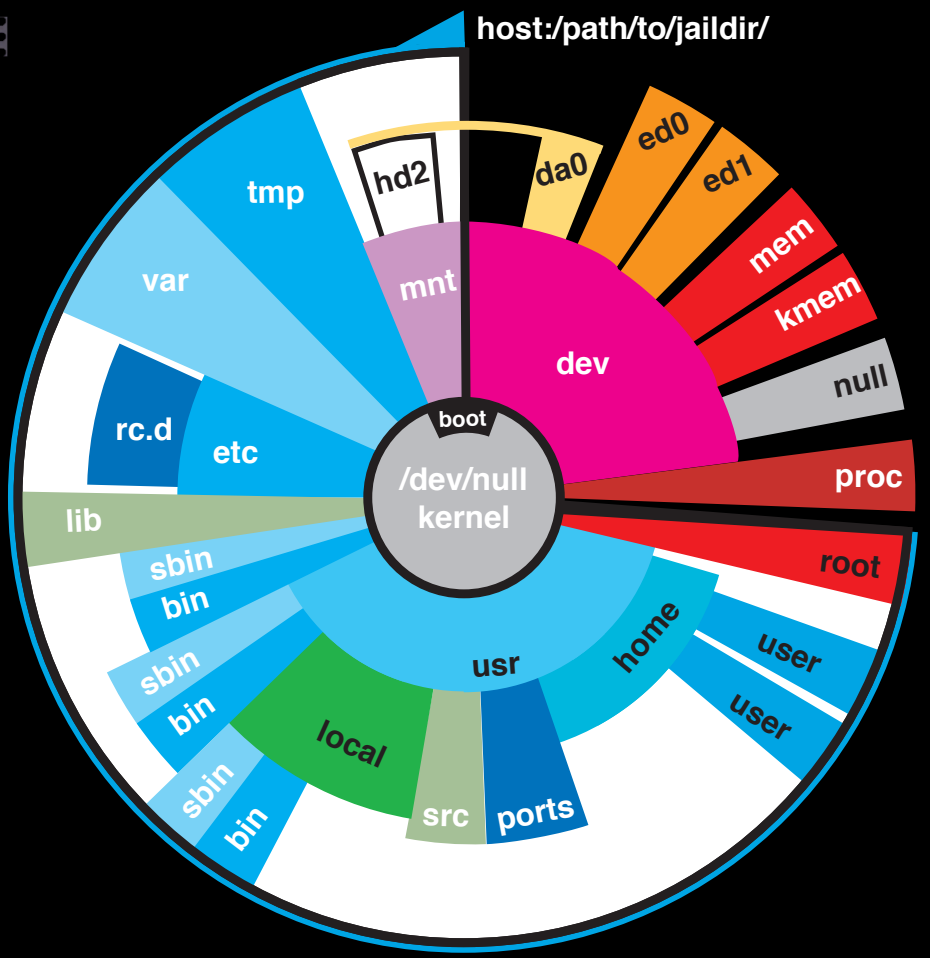
- kernel access (you don’t get a kernel)
- limited network interface access
- limited device driver access
- when chroot(8) will simply do the job
- some applications require particular low-level system calls:
  - Notably, PostgreSQL doesn’t run (securely) in jails based on SysV IPC

How To **jail(8)**

- DEFINITIVE instructions in **jail man pages**,
  1. compile a FreeBSD userland from source somewhere on host machine, minor tweaks.
  2. create an IP alias on a network interface
  3. run the jail(8) call with the IP, and userland, to ‘boot’ the jail, (so to speak).

Practical Comparison





making a jail

Host Machine

Terminal — ssh — 80x24

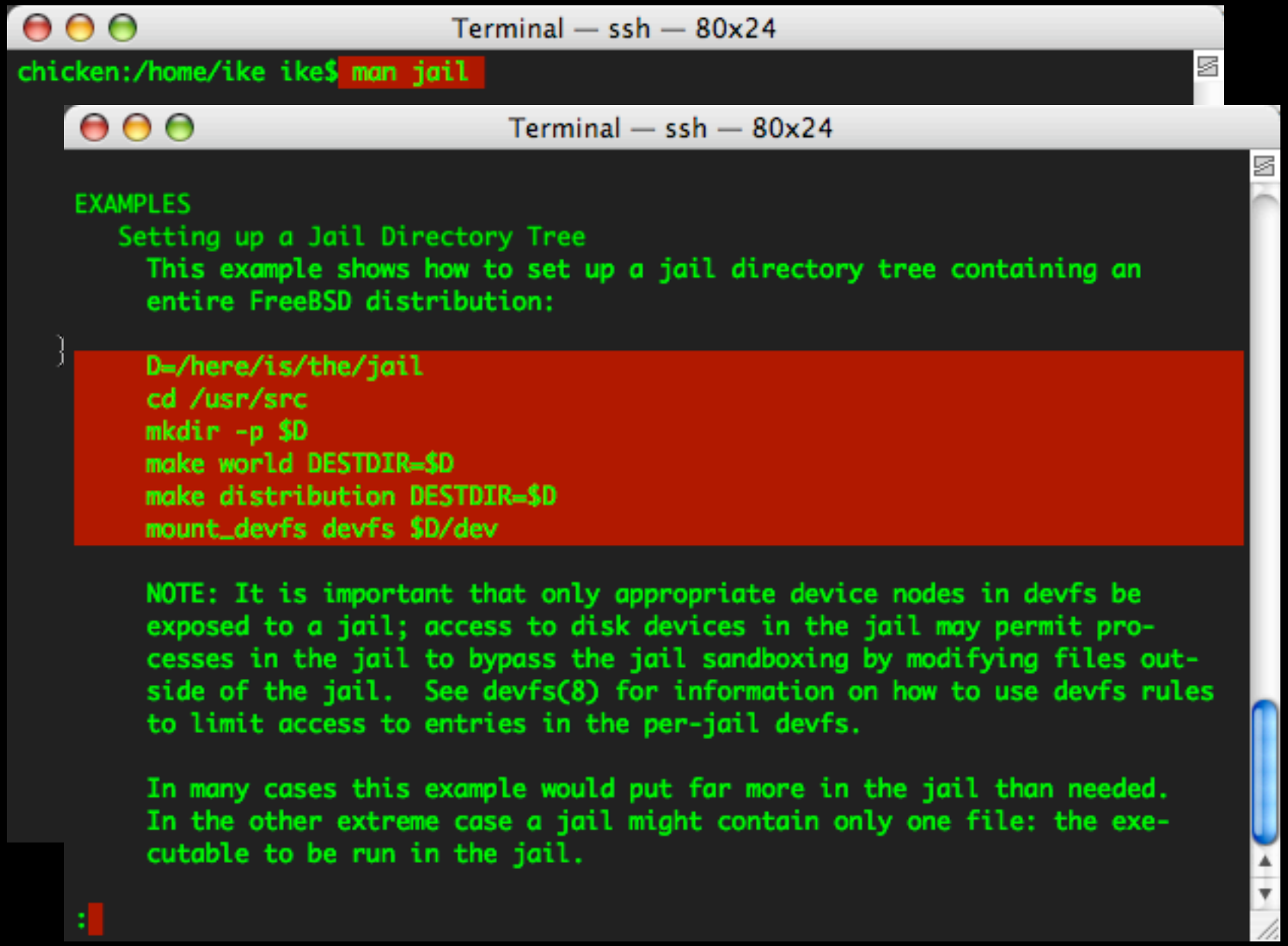
```
o  Security advisories and updated errata information for all relea
   at http://www.FreeBSD.org/releases/ - always consult the ERRATA
   for your release first as it's updated frequently.

o  The Handbook and FAQ documents are at http://www.FreeBSD.org/ and,
   along with the mailing lists, can be searched by going to
   http://www.FreeBSD.org/search/.  If the doc distribution has
   been installed, they're also available formatted in /usr/share/doc.

If you still have a question or problem, please take the output of
`uname -a', along with any relevant error messages, and email it
as a question to the questions@FreeBSD.org mailing list.  If you are
unfamiliar with FreeBSD's directory layout, please refer to the hier(7)
manual page.  If you are not familiar with manual pages, type `man man'.

You may also use sysinstall(8) to re-enter the installation and
configuration utility.  Edit /etc/motd to change this login announcement.

chicken:/home/ike ike$ uname -a
FreeBSD chicken.diversaform.net 6.1-RELEASE-p3 FreeBSD 6.1-RELEASE-p3 #3: Fri Ju
l 07 11:43:08 EDT 2006     root@chicken.diversaform.net:/usr/obj/usr/src/sys/GEN
ERIC  i386
chicken:/home/ike ike$
```

preflight (simple)

1. get source to build with (cvsup is great)
2. make somewhere for the jails to live (partitions, disk mounts, etc...)
3. make somewhere for jail-related start/mgmt scripts to live

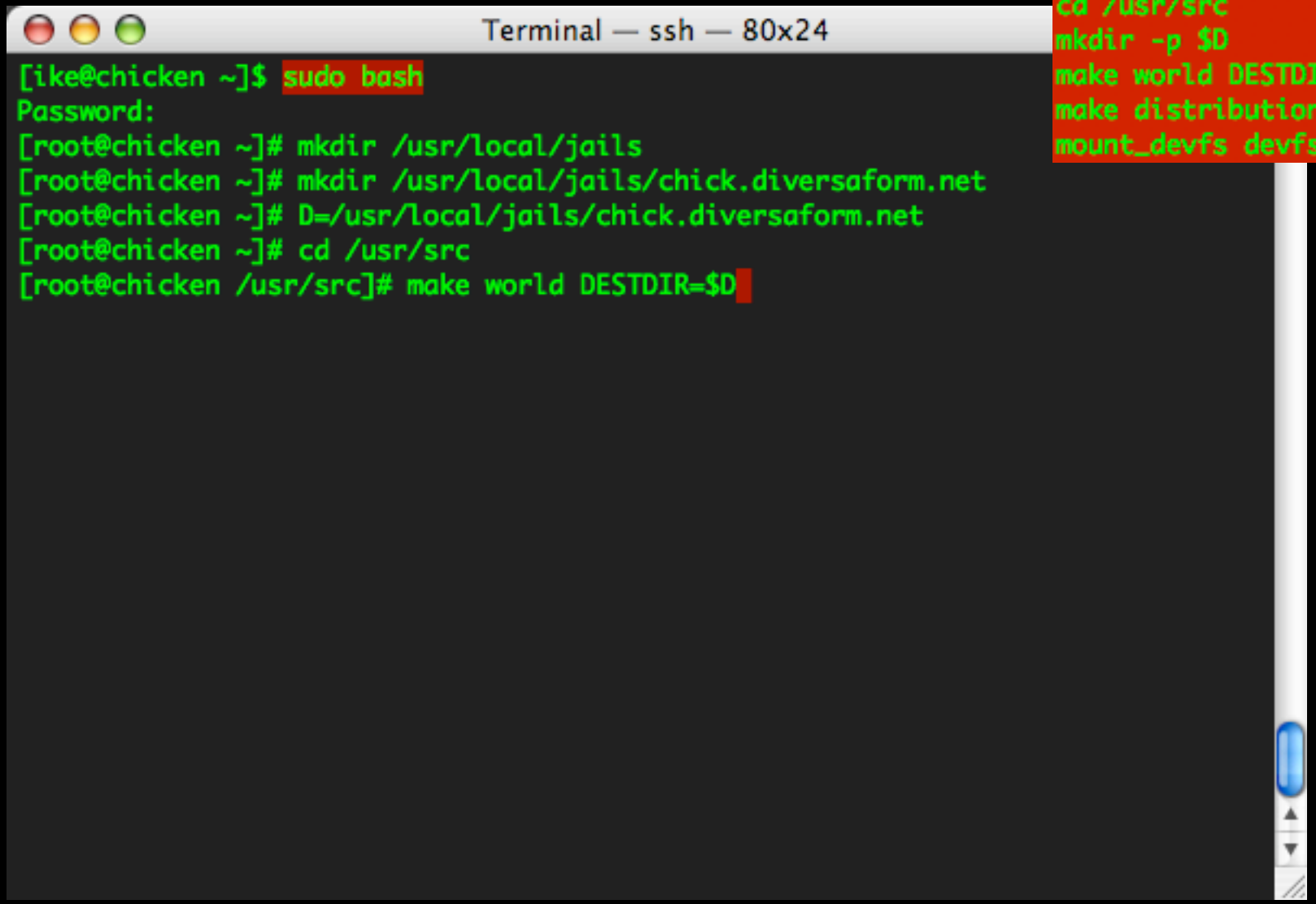
(starting jails from **/etc/rc.d/jail** can **thrash violently** in most contexts! Bad!)

preflight- (man, definitive)



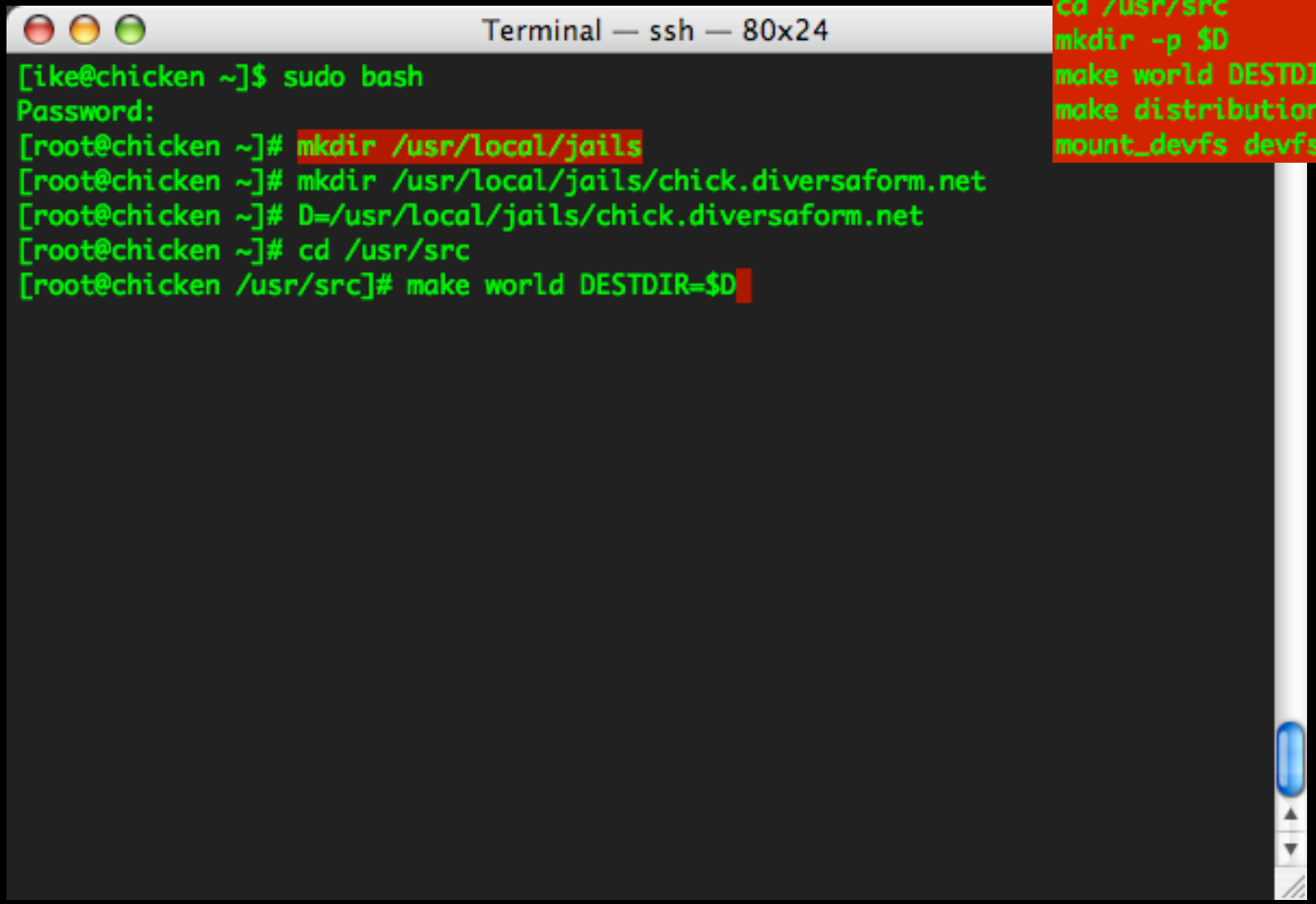
preflight- (build from src)

```
D=/here/is/the/jail
cd /usr/src
mkdir -p $D
make world DESTDIR=$D
make distribution DESTDIR=$D
mount_devfs devfs $D/dev
```



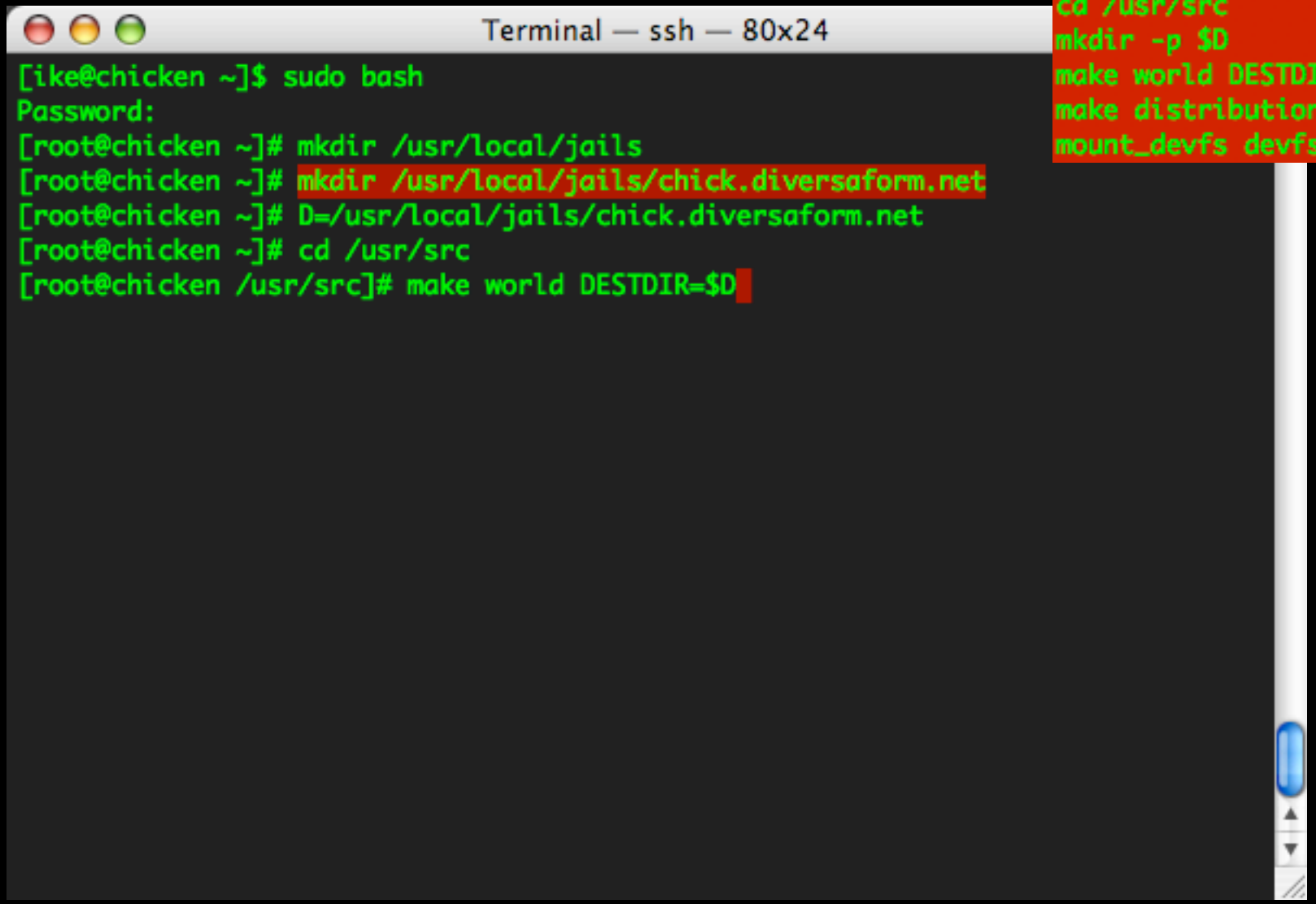
preflight- (build from src)

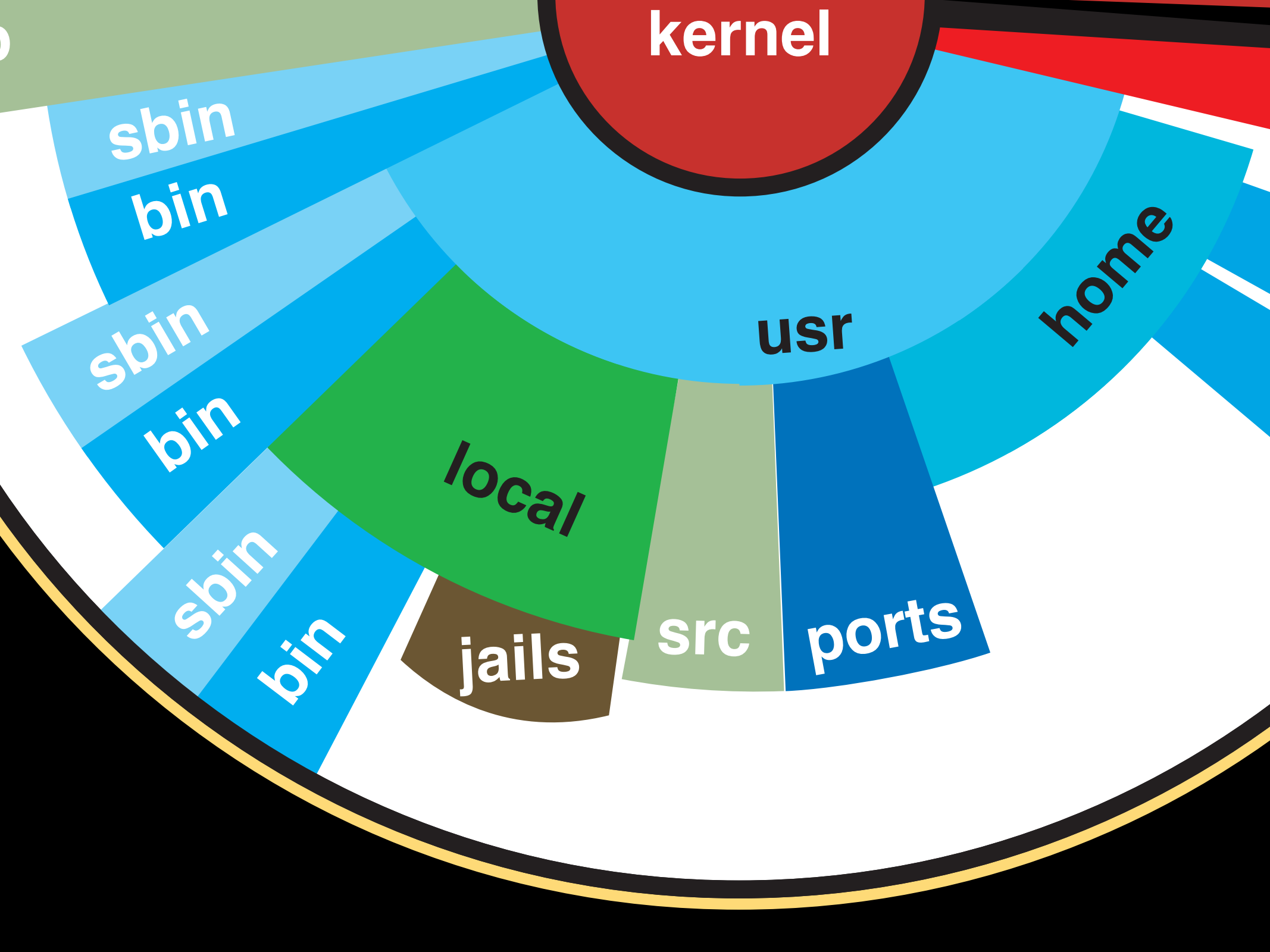
```
D=/here/is/the/jail
cd /usr/src
mkdir -p $D
make world DESTDIR=$D
make distribution DESTDIR=$D
mount_devfs devfs $D/dev
```



preflight- (build from src)

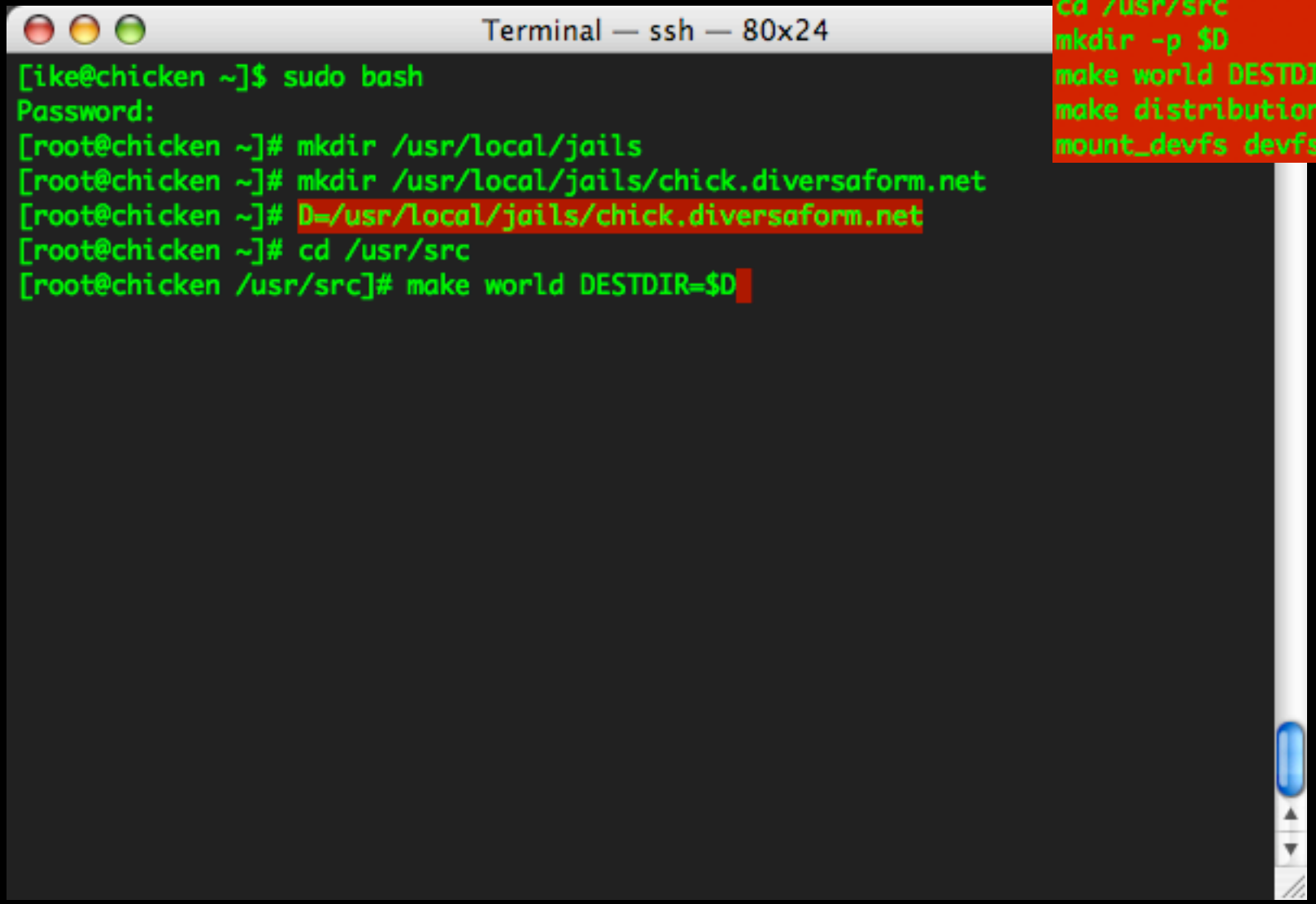
```
D=/here/is/the/jail
cd /usr/src
mkdir -p $D
make world DESTDIR=$D
make distribution DESTDIR=$D
mount_devfs devfs $D/dev
```

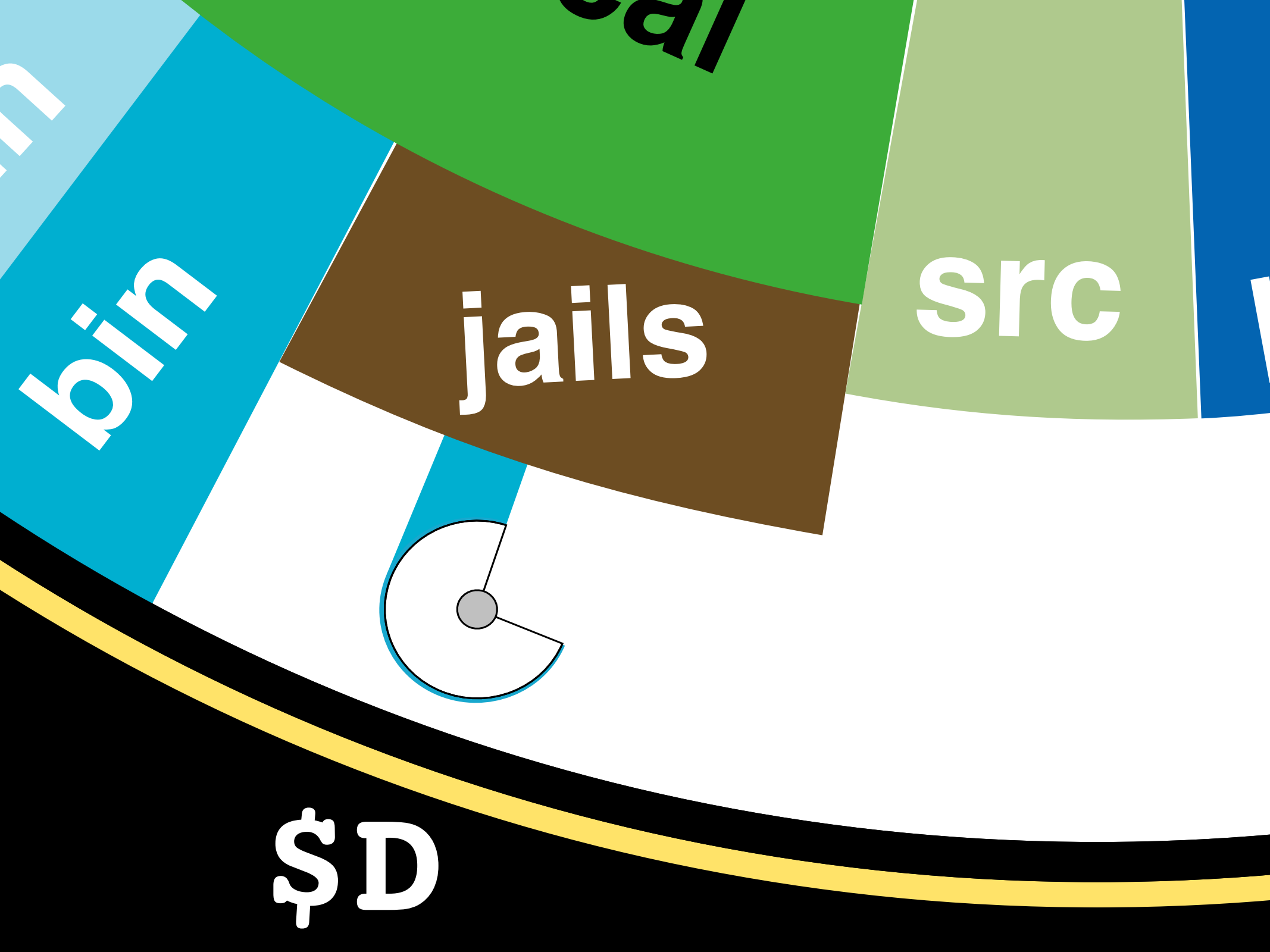



kernel
sbin
bin
sbin
bin
sbin
bin
usr
home
local
jails
src
ports

preflight- (build from src)

```
D=/here/is/the/jail
cd /usr/src
mkdir -p $D
make world DESTDIR=$D
make distribution DESTDIR=$D
mount_devfs devfs $D/dev
```




bin
jails
src
$D

preflight- (build from src)

```
D=/here/is/the/jail
cd /usr/src
mkdir -p $D
make world DESTDIR=$D
make distribution DESTDIR=$D
mount_devfs devfs $D/dev
```

Terminal — ssh — 80x24

```
[ike@chicken ~]$ sudo bash
Password:
[root@chicken ~]# mkdir /usr/local/jails
[root@chicken ~]# mkdir /usr/local/jails/chick.diversaform.net
[root@chicken ~]# D=/usr/local/jails/chick.diversaform.net
[root@chicken ~]# cd /usr/src
[root@chicken /usr/src]# make world DESTDIR=$D
--------------------------------------------------------------
>>> make world started on Mon Jul 10 14:39:21 EDT 2006
--------------------------------------------------------------


--------------------------------------------------------------
>>> Rebuilding the temporary build tree
--------------------------------------------------------------
rm -rf /usr/obj/usr/src/tmp
mkdir -p /usr/obj/usr/src/tmp/legacy/usr/bin
mkdir -p /usr/obj/usr/src/tmp/legacy/usr/games
mkdir -p /usr/obj/usr/src/tmp/legacy/usr/include/c++/3.4
mkdir -p /usr/obj/usr/src/tmp/legacy/usr/include/sys
mkdir -p /usr/obj/usr/src/tmp/legacy/usr/lib
mkdir -p /usr/obj/usr/src/tmp/legacy/usr/libexec
mkdir -p /usr/obj/usr/src/tmp/legacy/usr/sbin
mkdir -p /usr/obj/usr/src/tmp/legacy/usr/share/dict
mkdir -p /usr/obj/usr/src/tmp/legacy/usr/share/groff_font/devX100
```

compile!

preflight- (build from src)

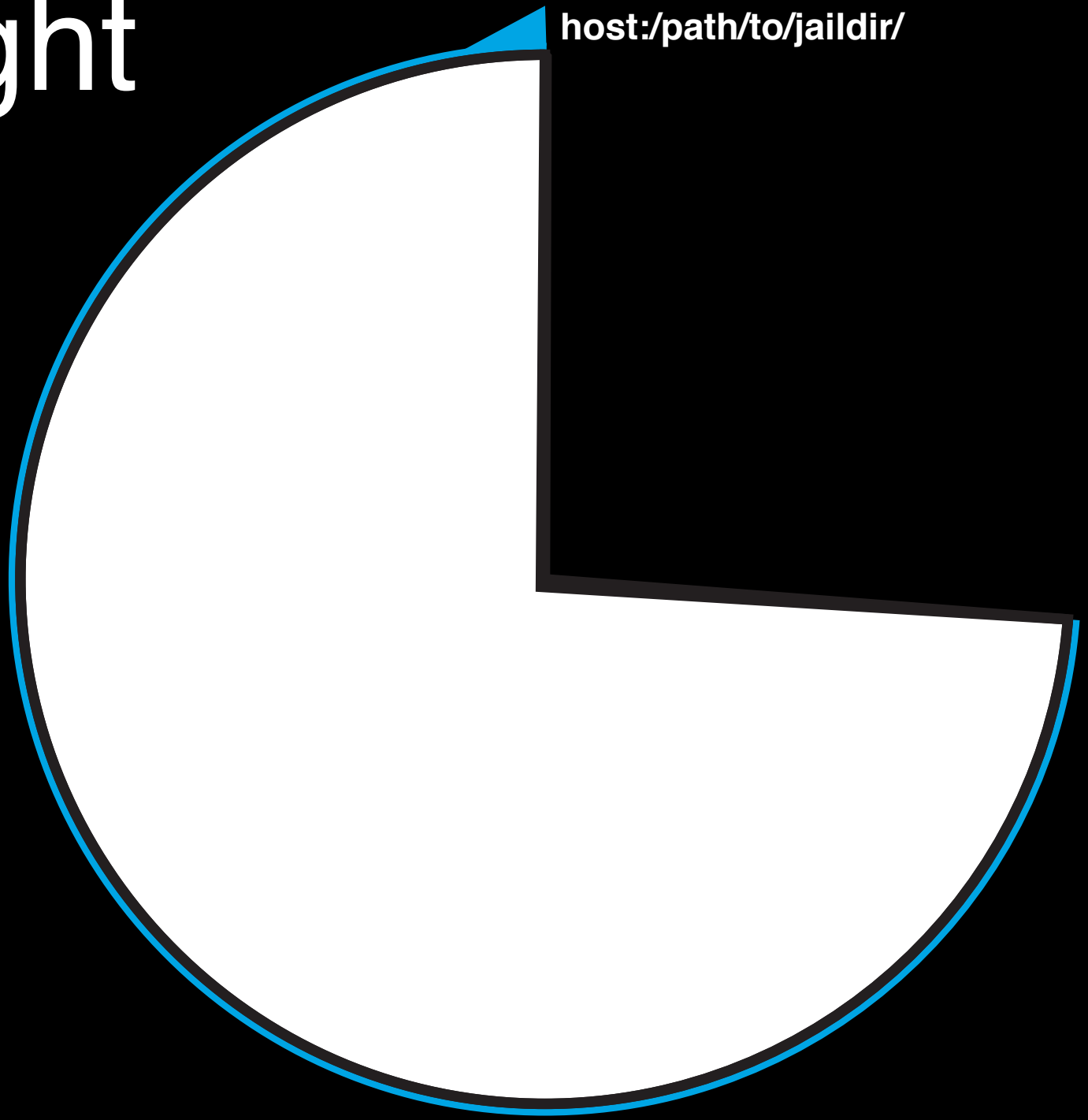
```
D=/here/is/the/jail
cd /usr/src
mkdir -p $D
make world DESTDIR=$D
make distribution DESTDIR=$D
mount_devfs devfs $D/dev
```

Terminal — ssh — 80x24

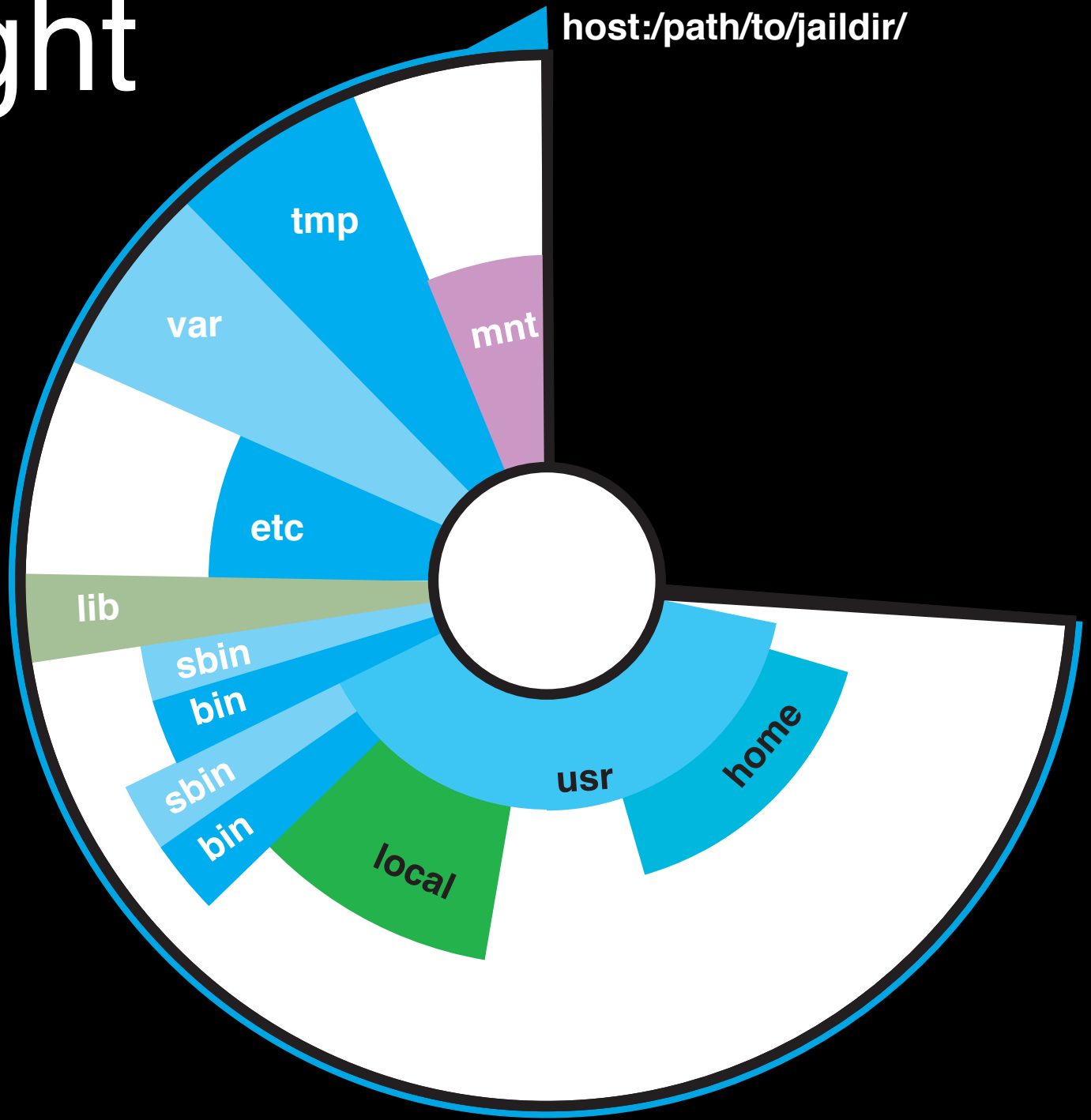
```
[root@chicken /usr/src]# make distribution DESTDIR=$D
cd /usr/src/etc; MAKEOBJDIRPREFIX=/usr/obj  MACHINE_ARCH=i386  MACH
TYPE=  GROFF_BIN_PATH=/usr/obj/usr/src/tmp/legacy/usr/bin  GROFF_FONT_PATH=/usr/
obj/usr/src/tmp/legacy/usr/share/groff_font  GROFF_TMAC_PATH=/usr/obj/usr/src/tm
p/legacy/usr/share/tmac PATH=/usr/obj/usr/src/tmp/legacy/usr/sbin:/usr/obj/usr/s
rc/tmp/legacy/usr/bin:/usr/obj/usr/src/tmp/legacy/usr/games:/usr/obj/usr/src/tmp
/usr/sbin:/usr/obj/usr/src/tmp/usr/bin:/usr/obj/usr/src/tmp/usr/games:/sbin:/bin
:/usr/sbin:/usr/bin /usr/obj/usr/src/make.i386/make distribution
cd /usr/src/etc;  install -o root -g wheel -m 644  amd.map apmd.conf auth.conf
crontab csh.cshrc csh.login csh.logout devd.conf devfs.conf  dhclient.conf diskt
ab fbtab ftpusers gettytab group  hosts hosts.allow hosts.equiv hosts.lpd  inetd
.conf login.access login.conf  mac.conf motd netconfig network.subr
syslog.conf  portsnap.conf pf.conf pf.os phones profile protocol
nded rc.firewall rc.firewall6 rc.initdiskless  rc.sendmail rc.shutdo
remote rpc services shells  snmpd.config sysctl.conf syslog.conf usb
i386/ttys  /usr/src/etc/../gnu/usr.bin/man/manpath/manpath.config  /
../usr.bin/mail/misc/mail.rc  /usr/src/etc/../usr.bin/locate/locate/
intcap /usr/local/jails/chick.diversaform.net/etc;  cap_mkdb -l /usr
/chick.diversaform.net/etc/login.conf;  install -o root -g wheel -m
t pccard_ether rc.suspend rc.resume /usr/local/jails/chick.diversafo
 install -o root -g wheel -m 600  master.passwd nsmb.conf opieaccess
jails/chick.diversaform.net/etc;  pwd_mkdb -L -i -p -d /usr/local/jc
versaform.net/etc  /usr/local/jails/chick.diversaform.net/etc/master
```

compile!

preflight

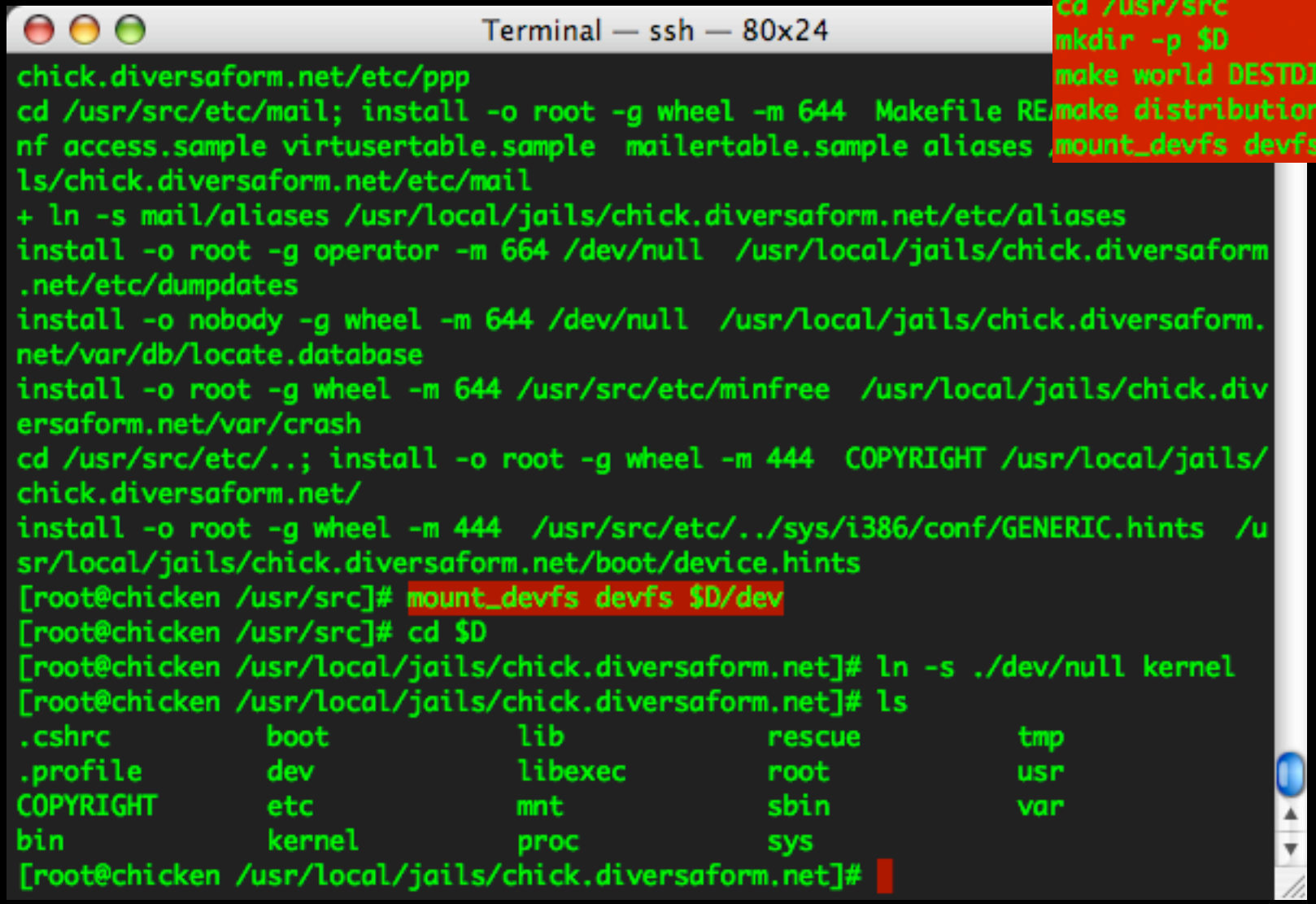


preflight


host:/path/to/jaildir/
tmp
var
mnt
etc
lib
sbin
bin
sbin
bin
usr
home
local

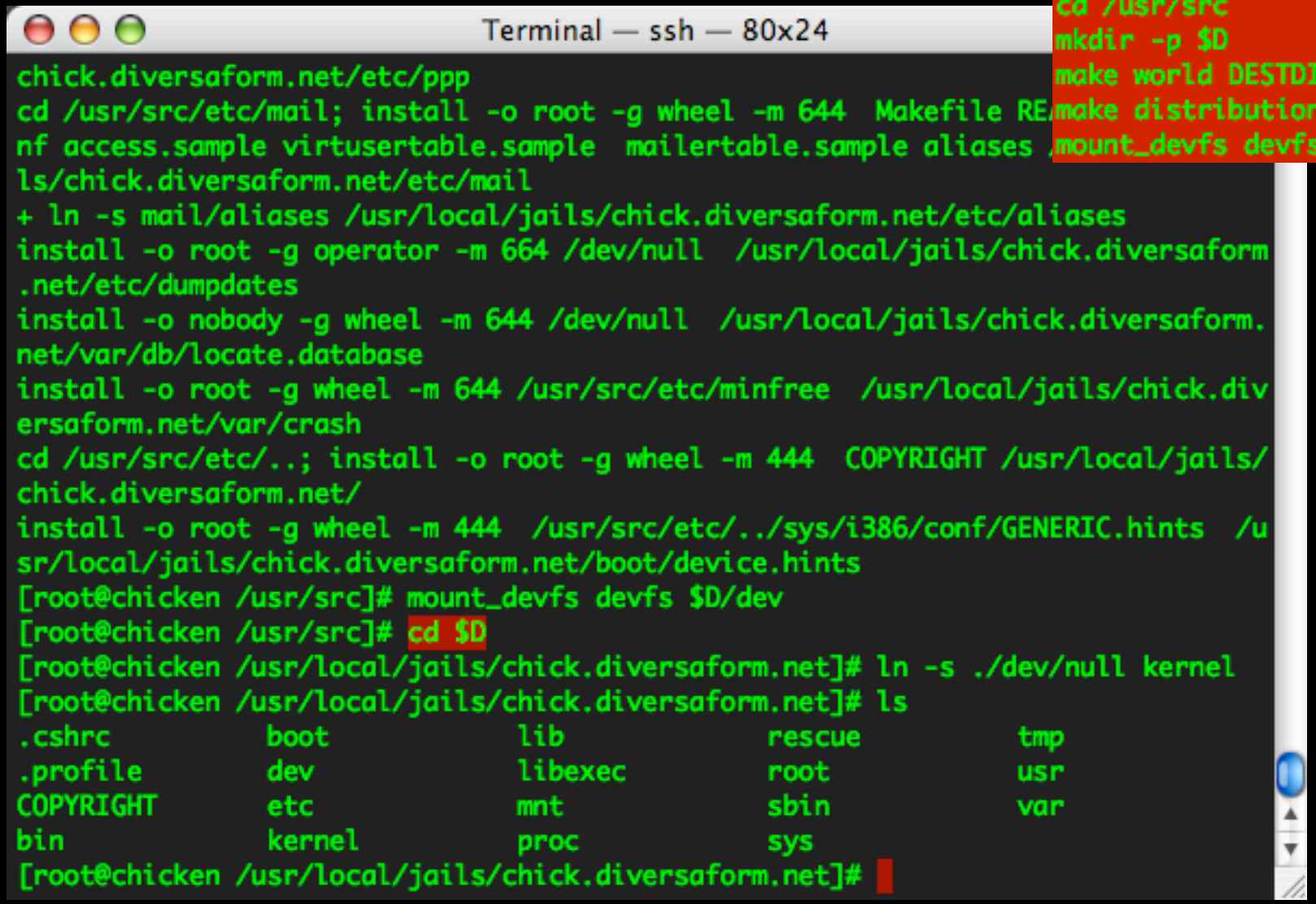
preflight- (mount /dev)

```
D=/here/is/the/jail
cd /usr/src
mkdir -p $D
make world DESTDIR=$D
make distribution DESTDIR=$D
mount_devfs devfs $D/dev
```

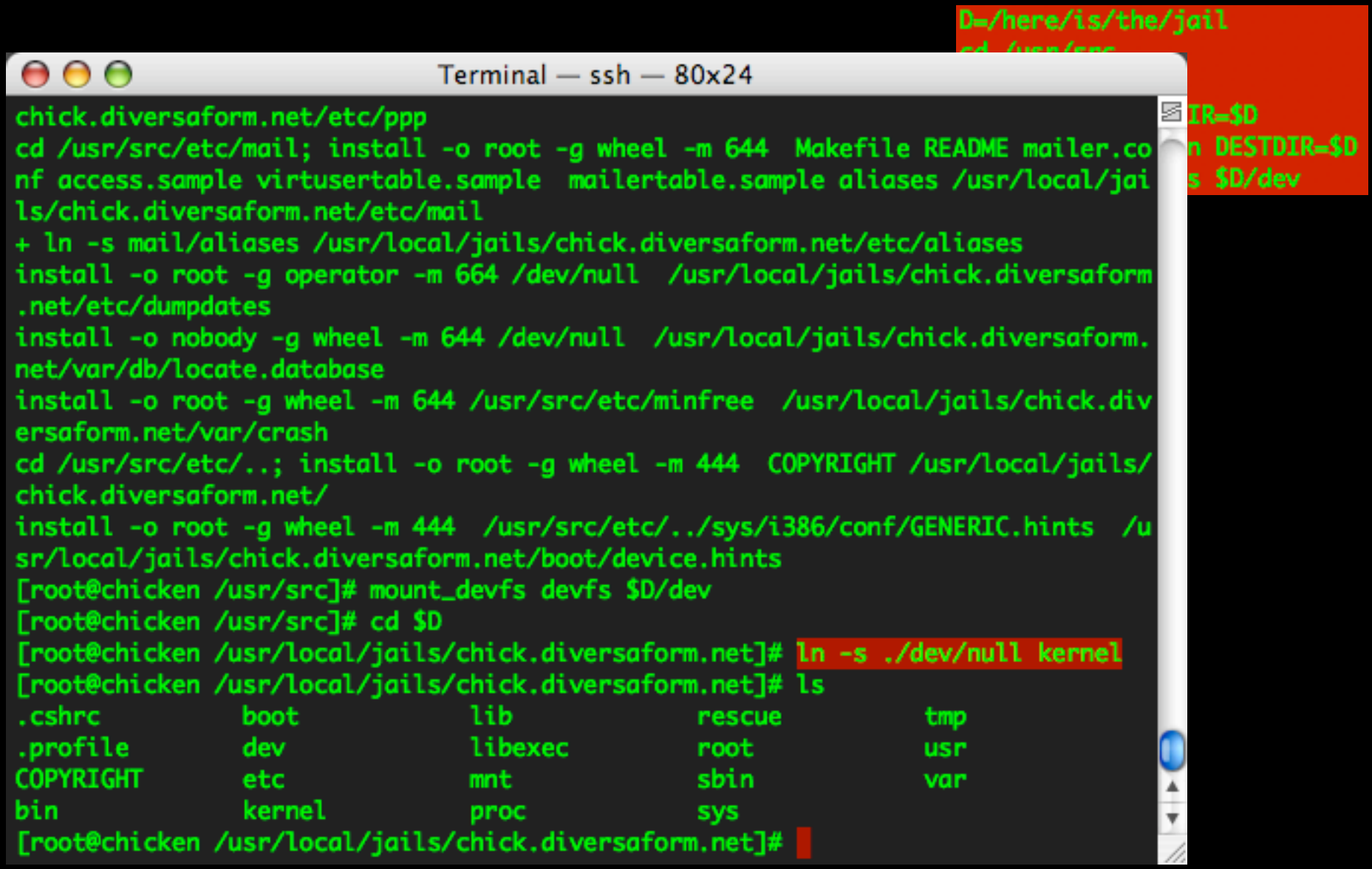


preflight- (mount /dev)

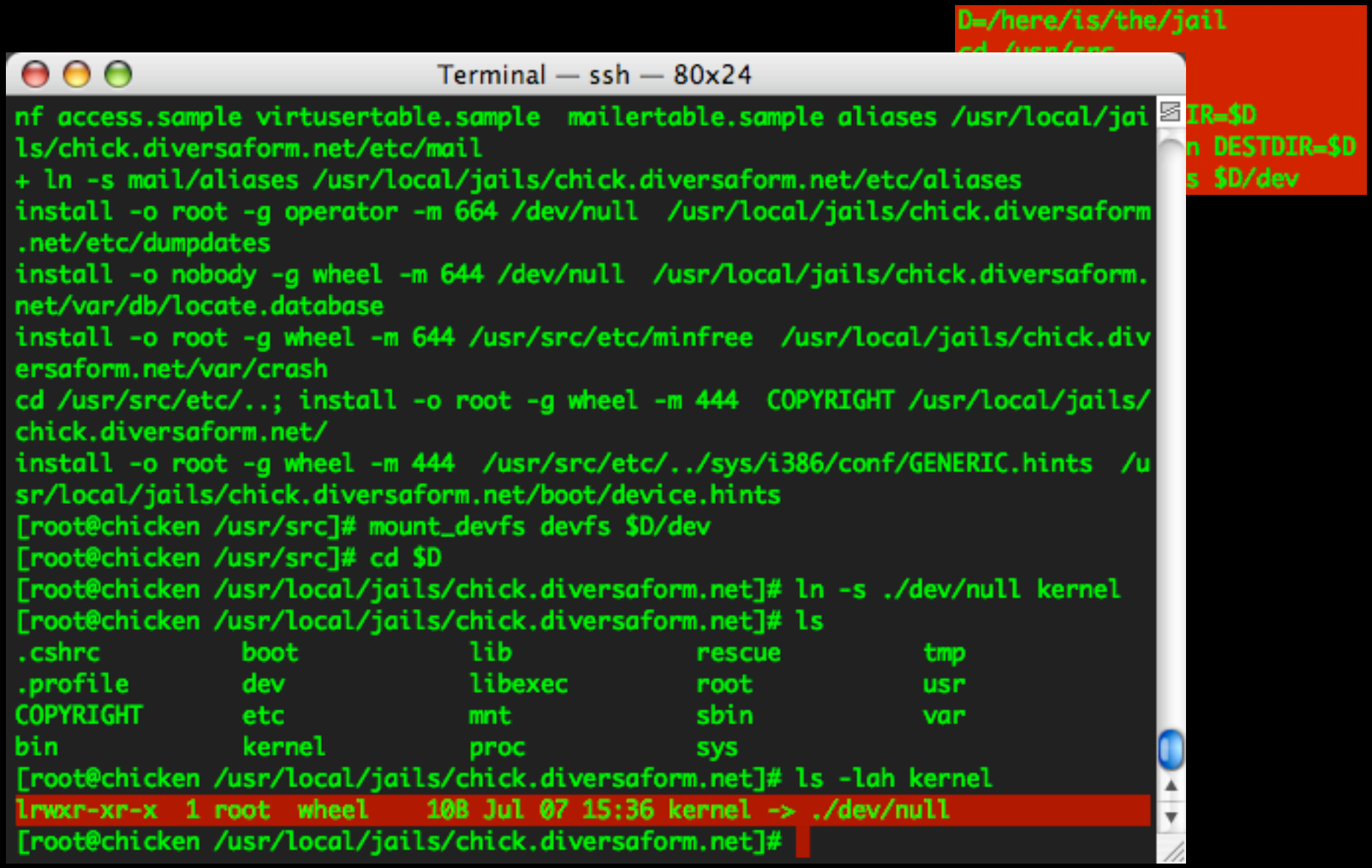
```
D=/here/is/the/jail
cd /usr/src
mkdir -p $D
make world DESTDIR=$D
make distribution DESTDIR=$D
mount_devfs devfs $D/dev
```



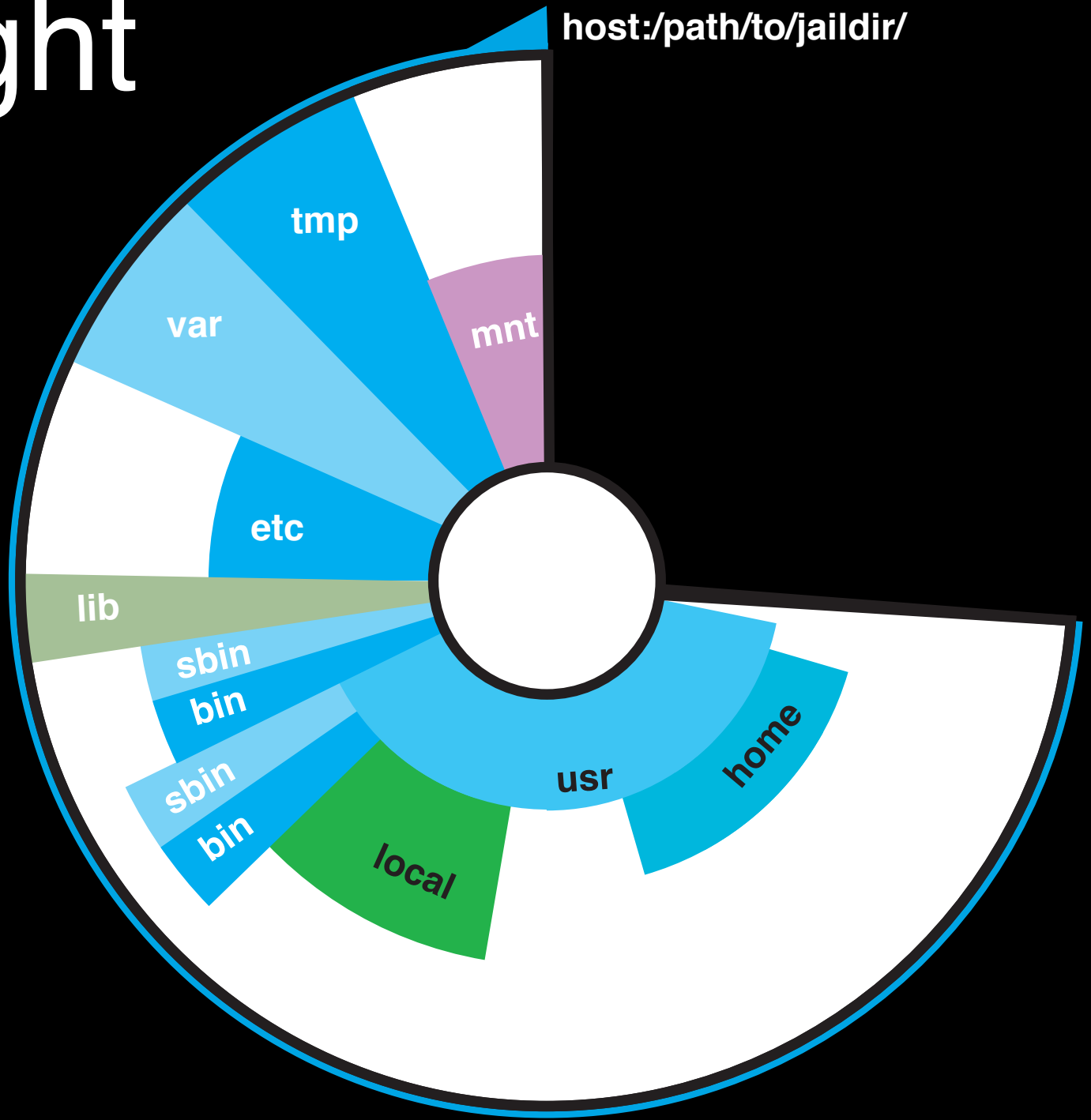
preflight- (null kernel)



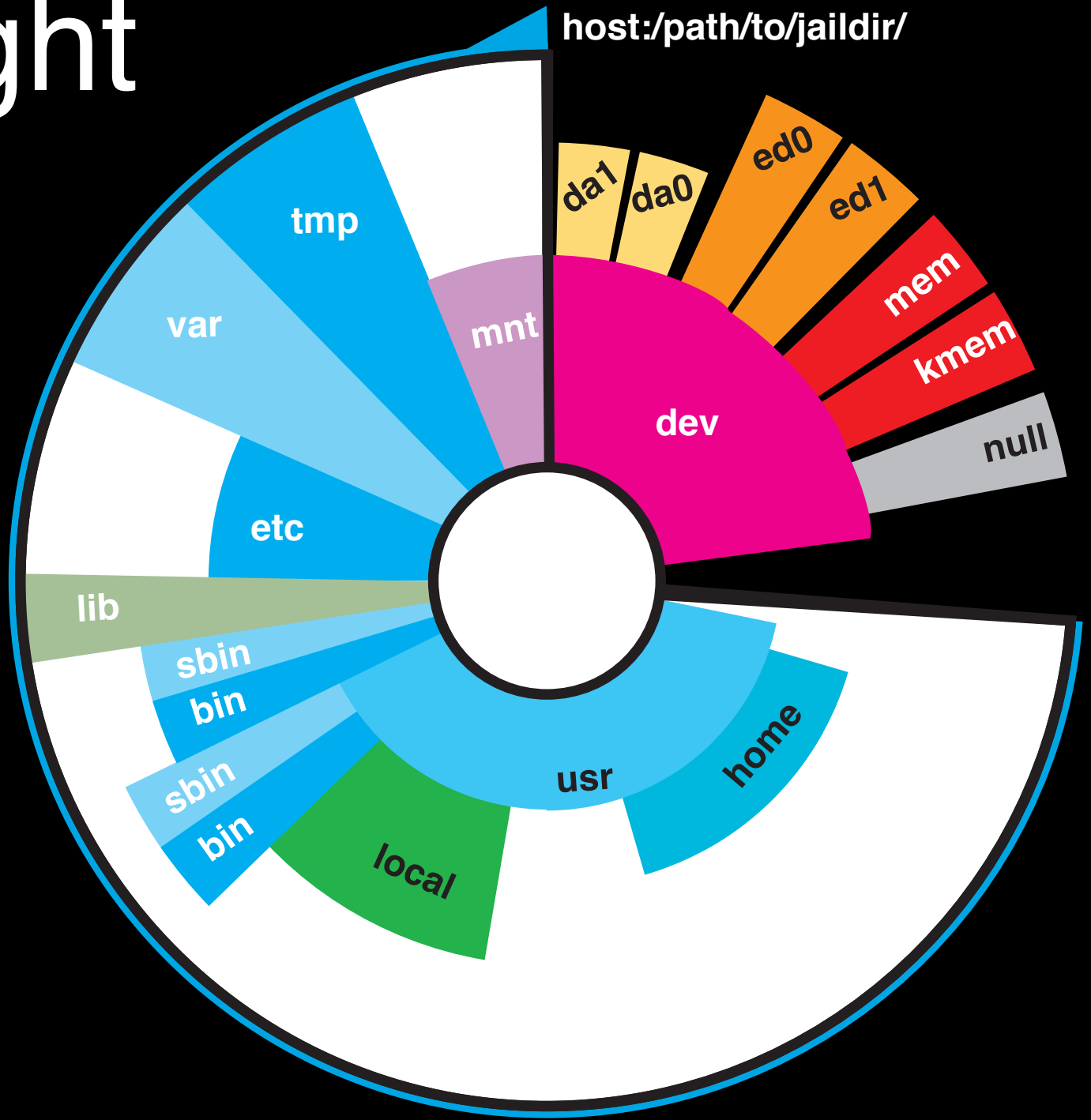
preflight- (null kernel)



preflight


host:/path/to/jaildir/
tmp
var
mnt
etc
lib
sbin
bin
sbin
bin
usr
home
local

preflight


host:/path/to/jaildir/
da1
da0
ed0
ed1
mem
kmem
null
dev
mnt
tmp
var
etc
lib
sbin
bin
sbin
bin
local
usr
home

preflight

```
D=/here/is/the/jail
cd /usr/src
mkdir -p $D
make world DESTDIR=$D
make distribution DESTDIR=$D
mount_devfs devfs $D/dev
```

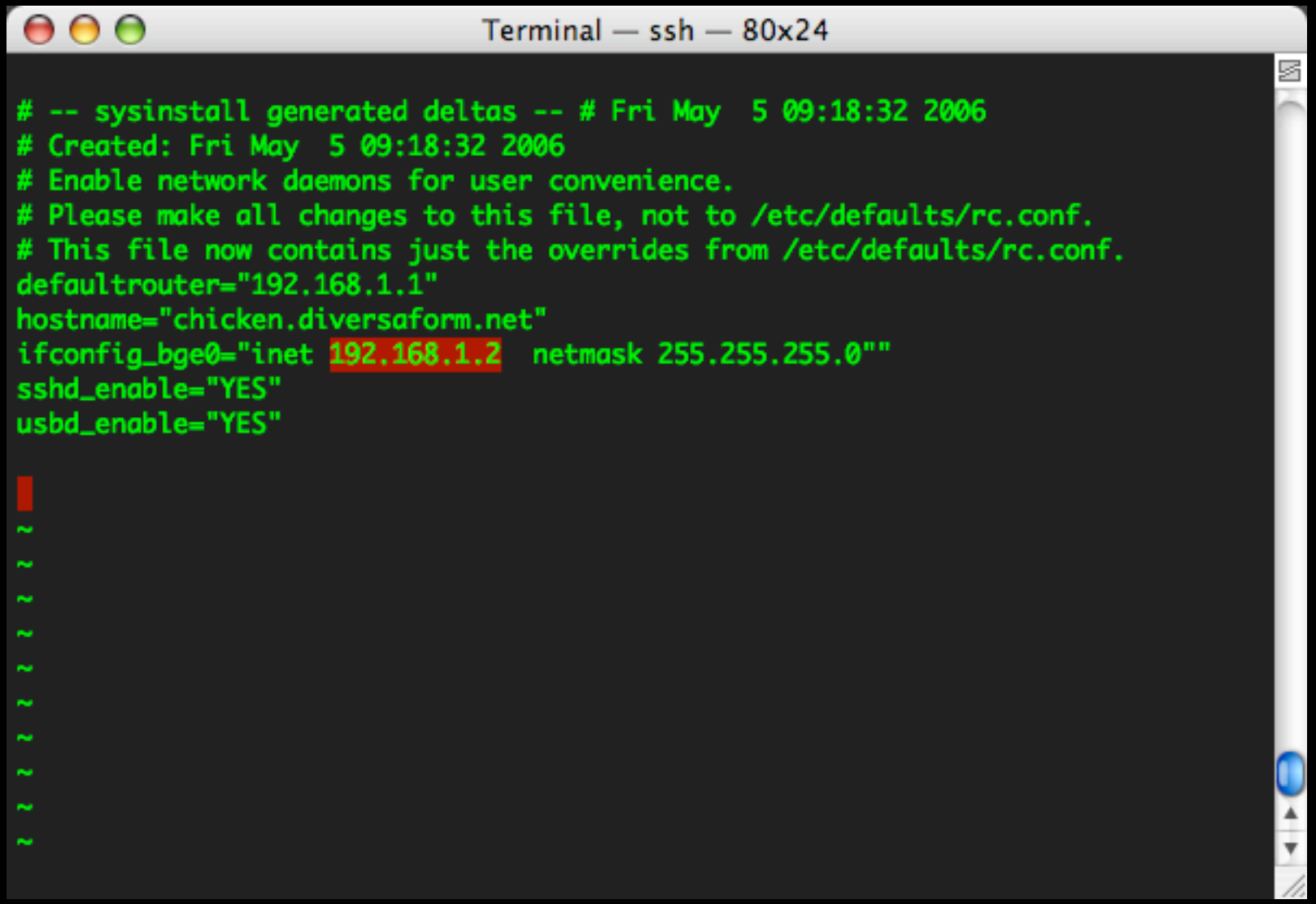
Common Question:

Why isn't there an automated build system for this stage?

- Take care with the build procedure, it's better to automate things later, once you have basics setup.

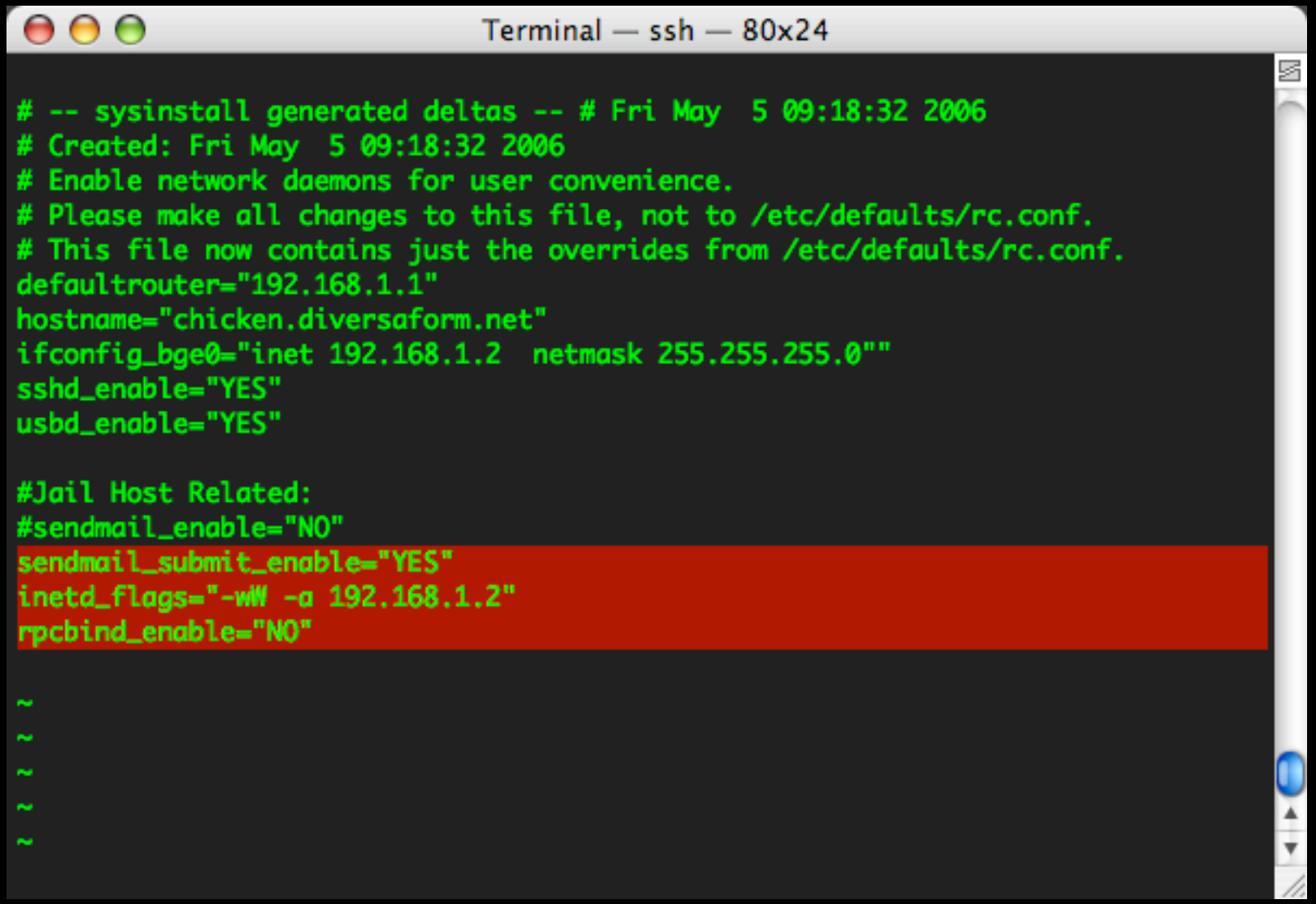
(network, users, packages, time, etc.)

preflight- (config host)



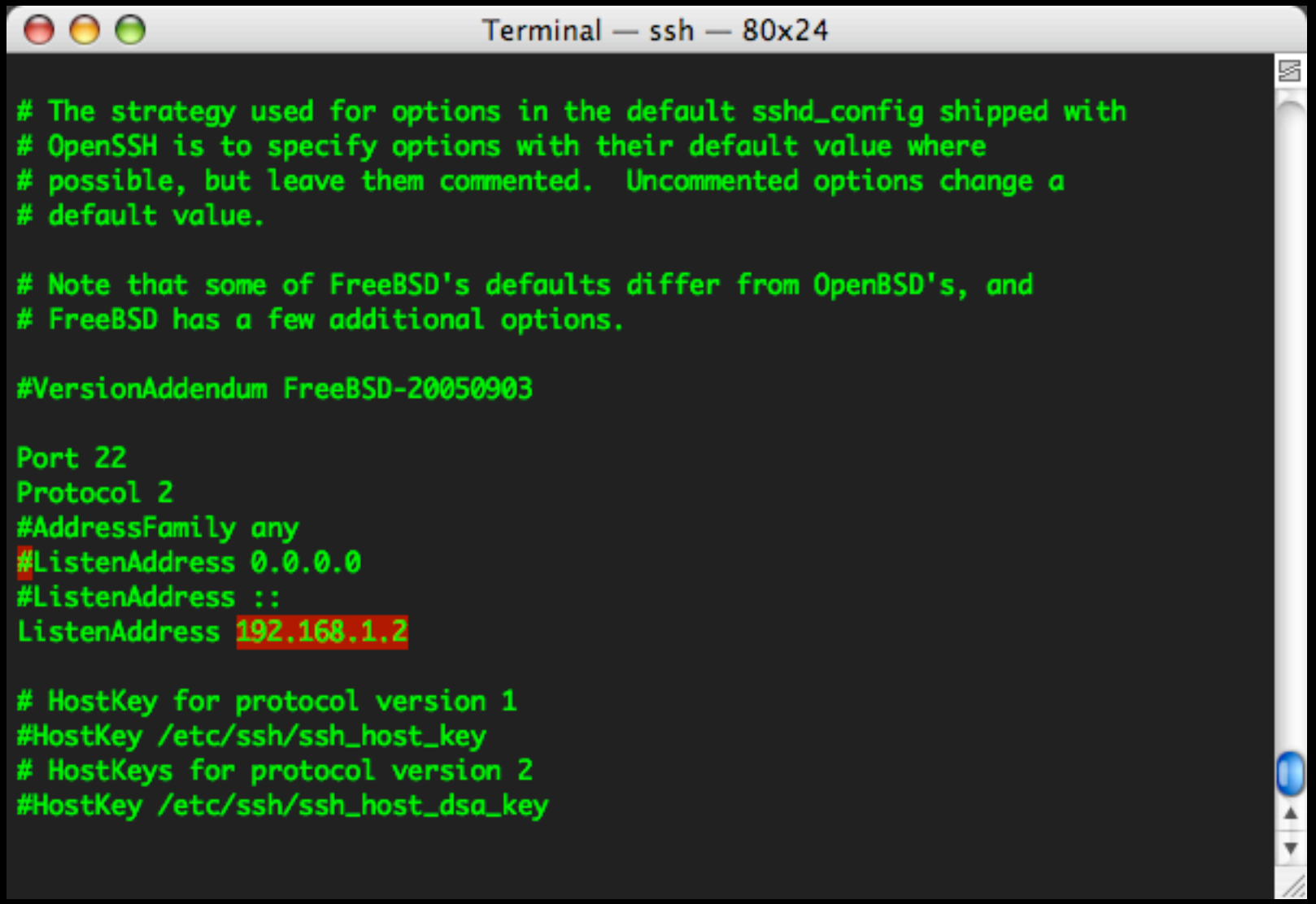
jailinghost:/etc/rc.conf (stock)

preflight- (config host)

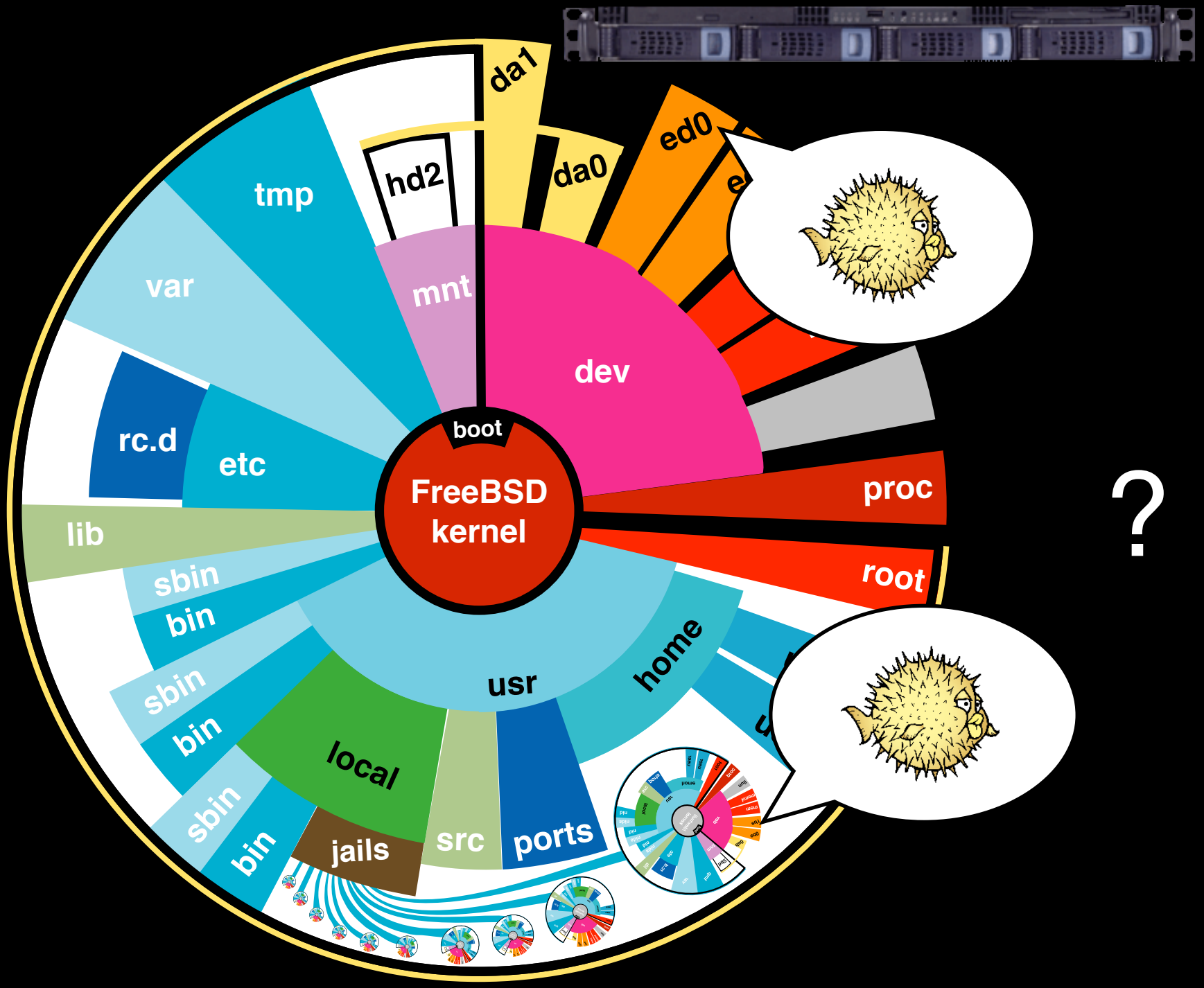


jailinghost:/etc/rc.conf

preflight- (master system)

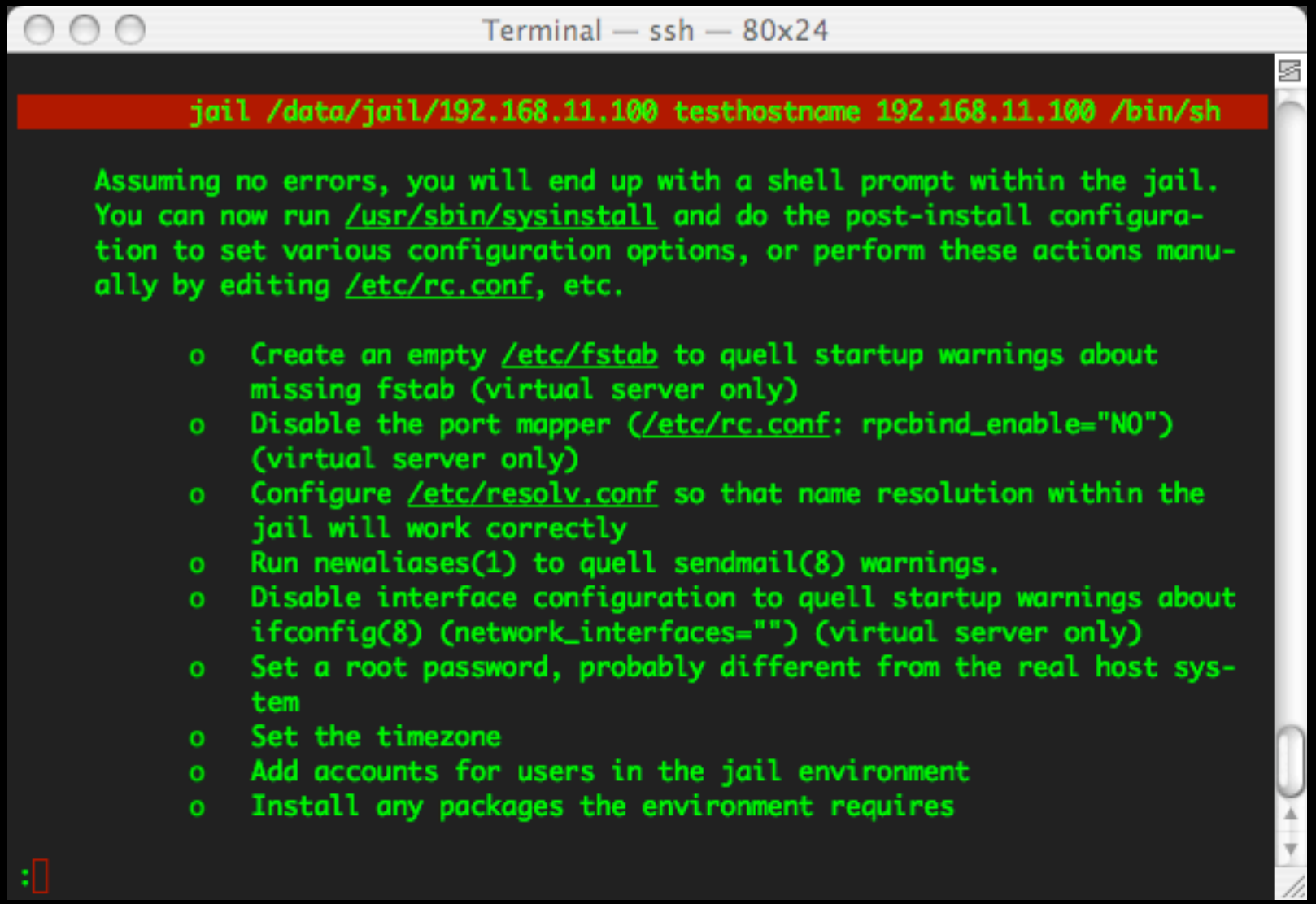


jailinghost:/etc/ssh/sshd_conf


da1
hd2
da0
ed0
tmp
mnt
var
dev
boot
rc.d
etc
FreeBSD
kernel
proc
lib
root
sbin
bin
usr
home
sbin
bin
local
sbin
bin
jails
src
ports
?

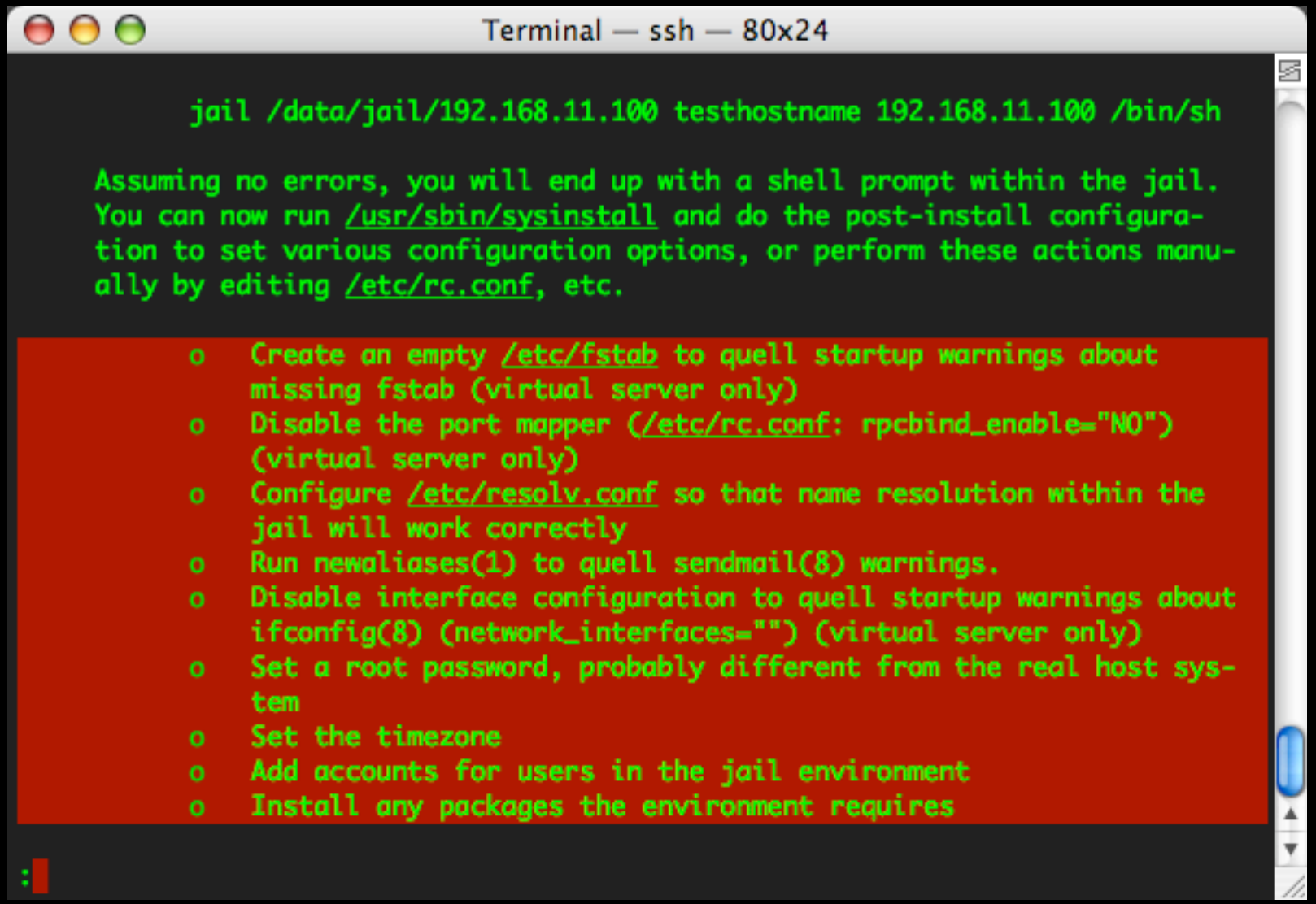
configure - call jailed sh

(analagous to booting a machine in su mode)



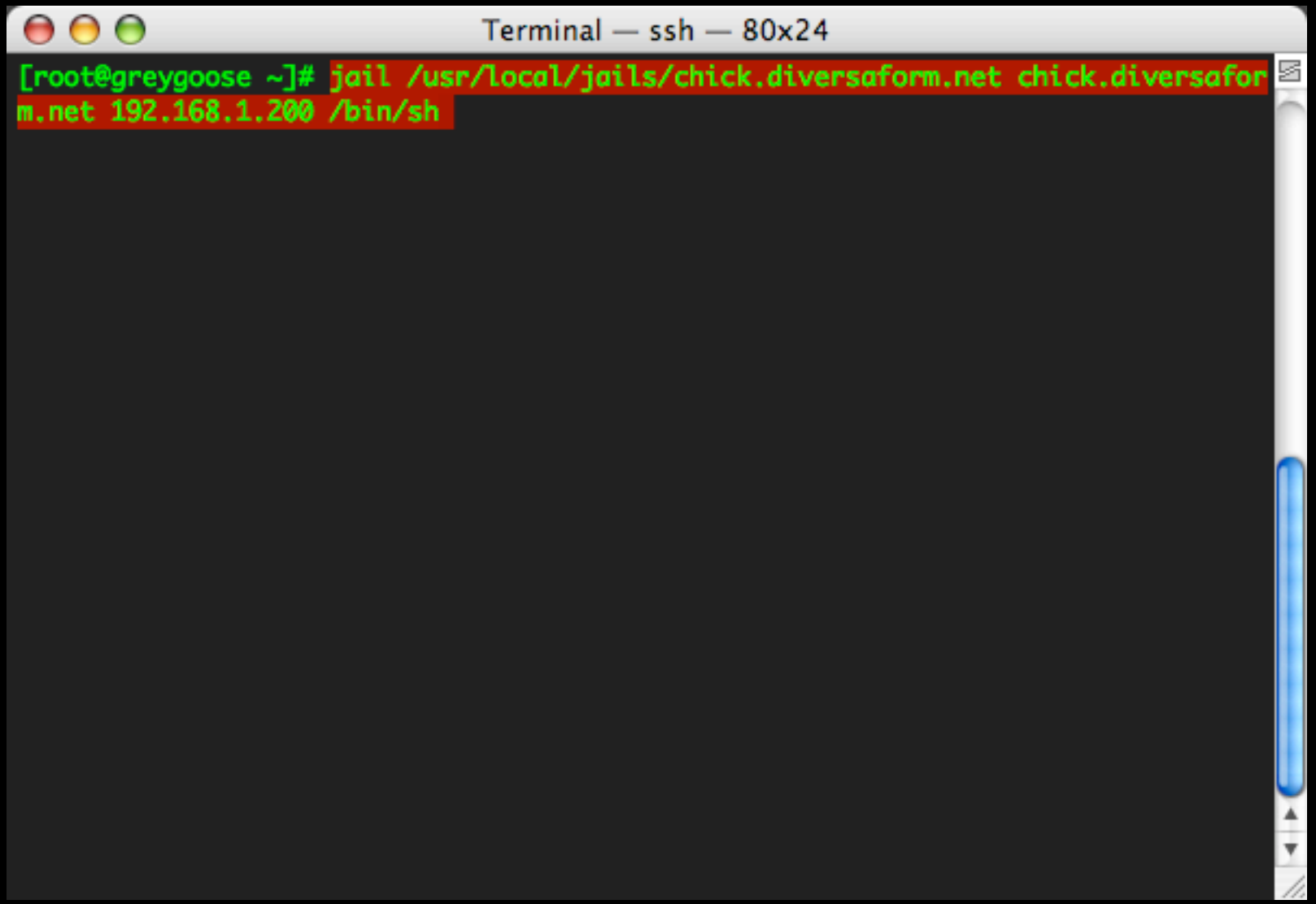
configure - call jailed sh

(analagous to booting a machine in su mode)



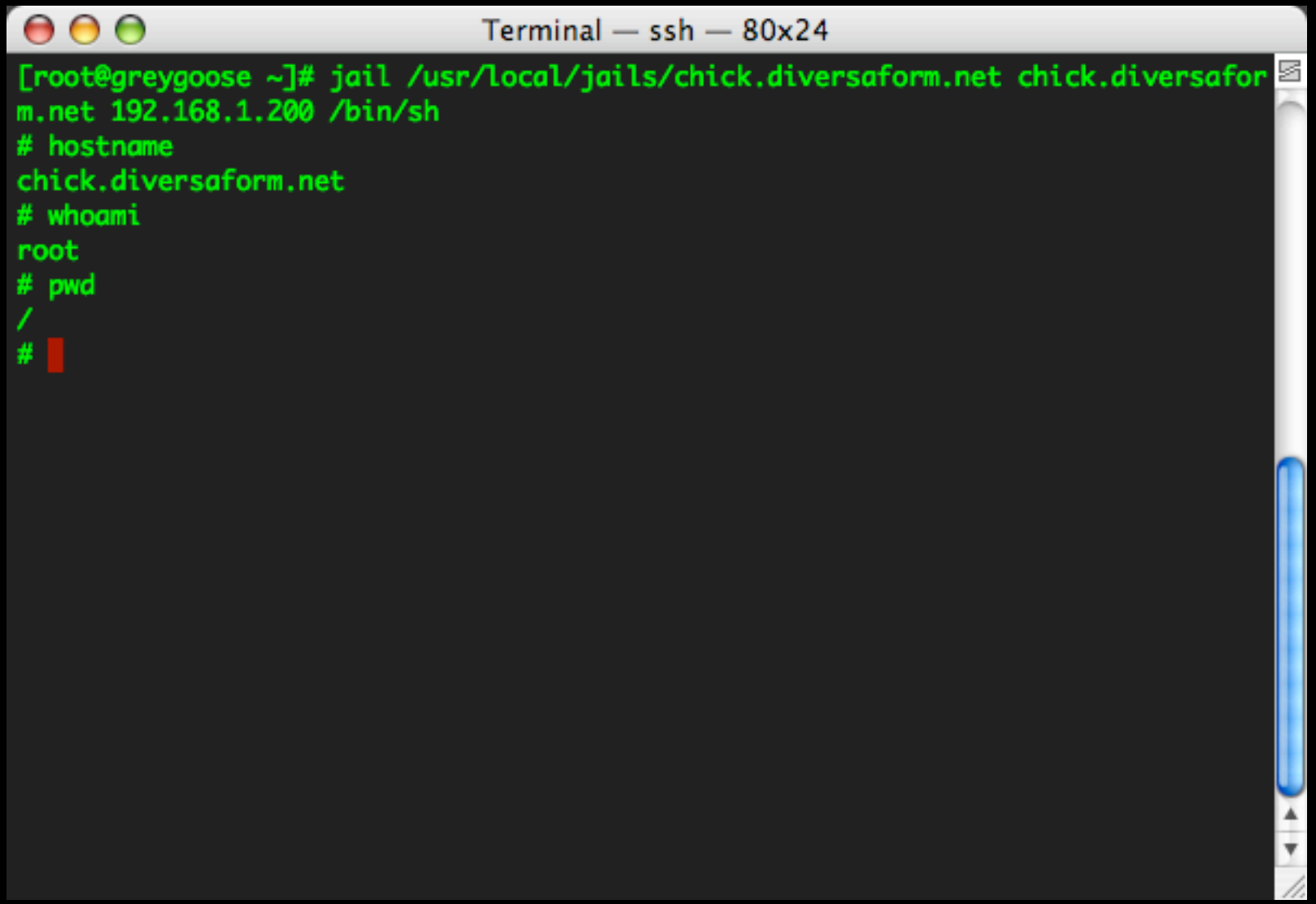
configure - call jailed sh

(analagous to booting a machine in su mode)



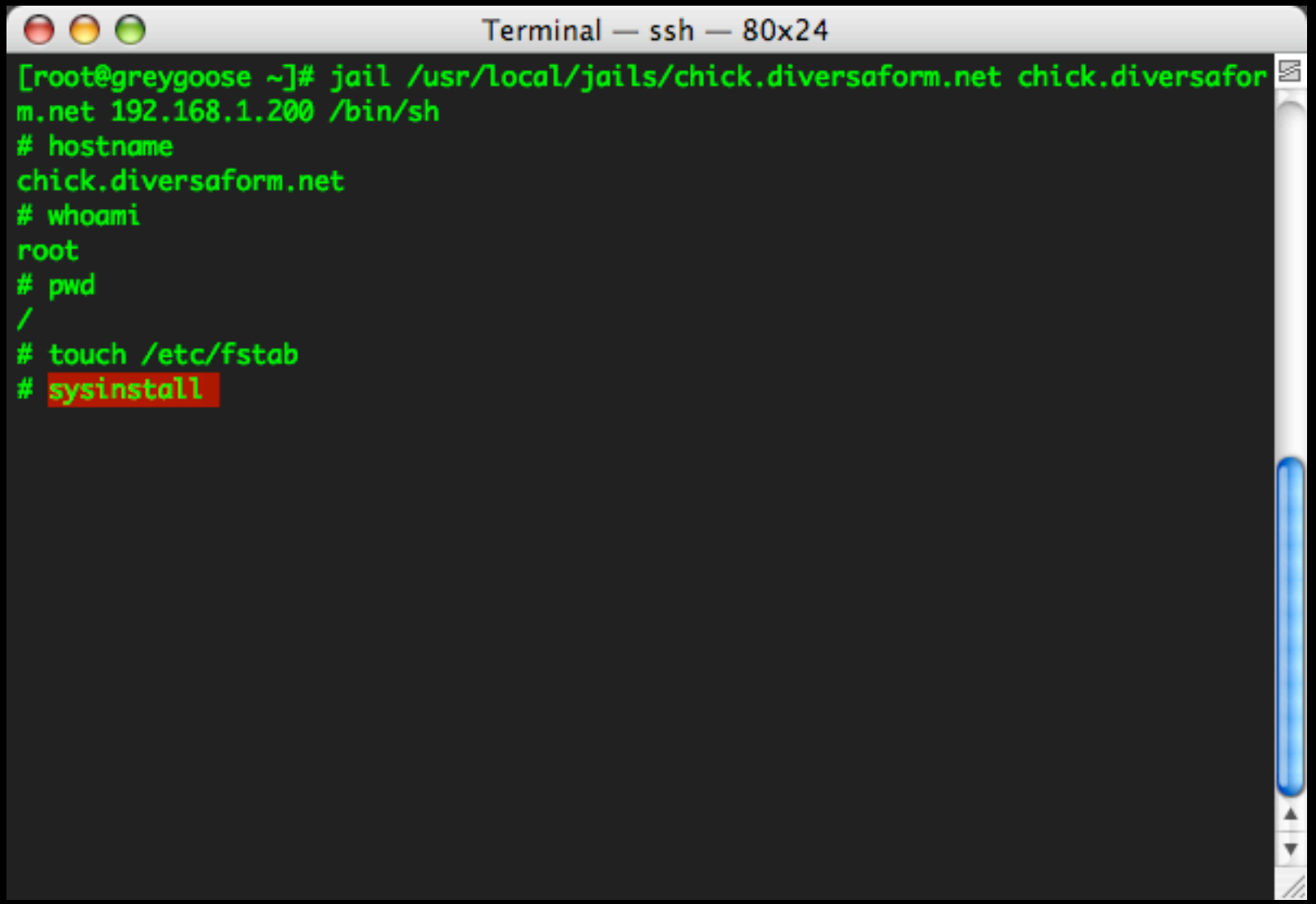
configure - call jailed sh

configure the jail, **inside the jail**



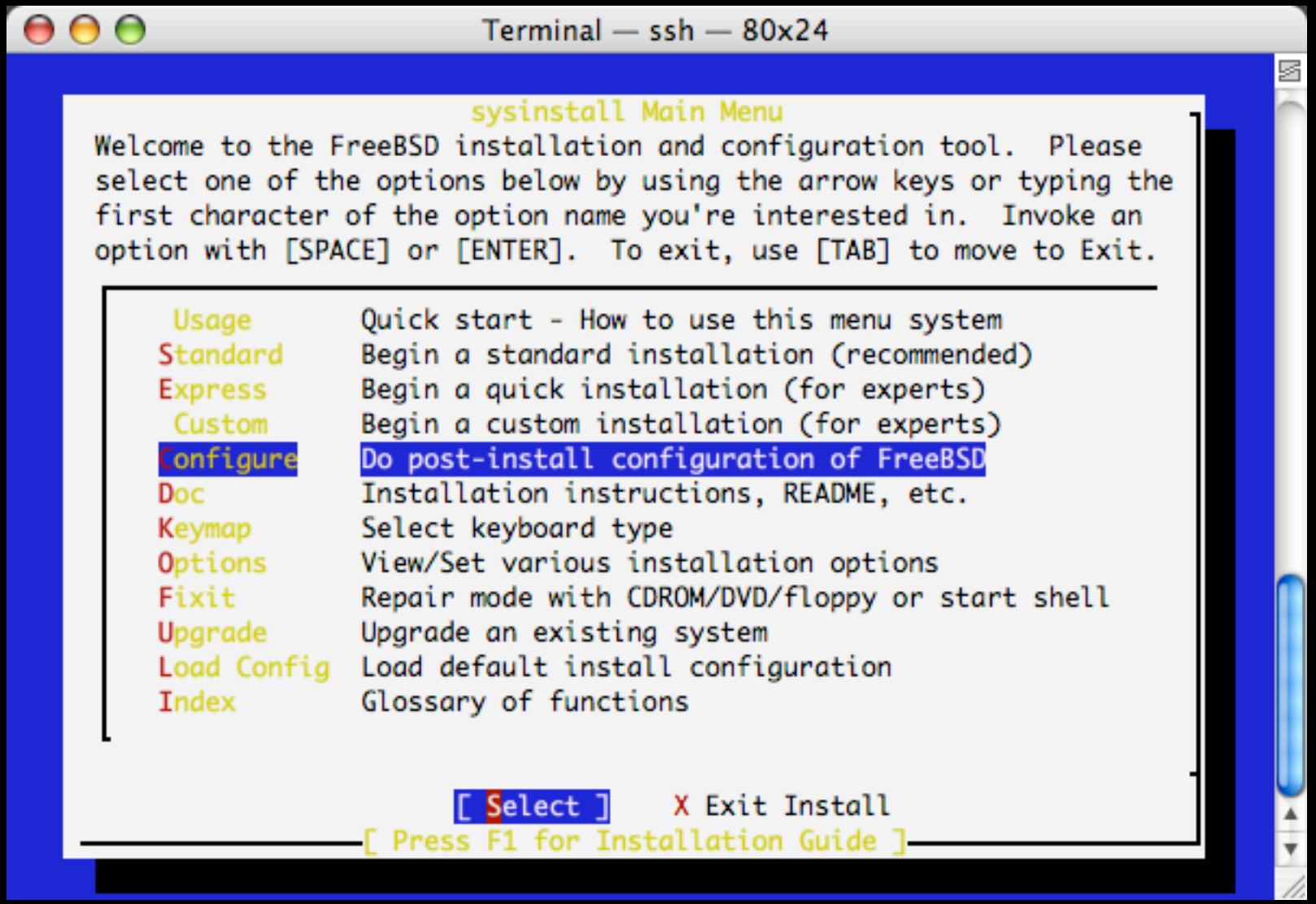
configure - call jailed sh

configure the jail, **inside the jail**



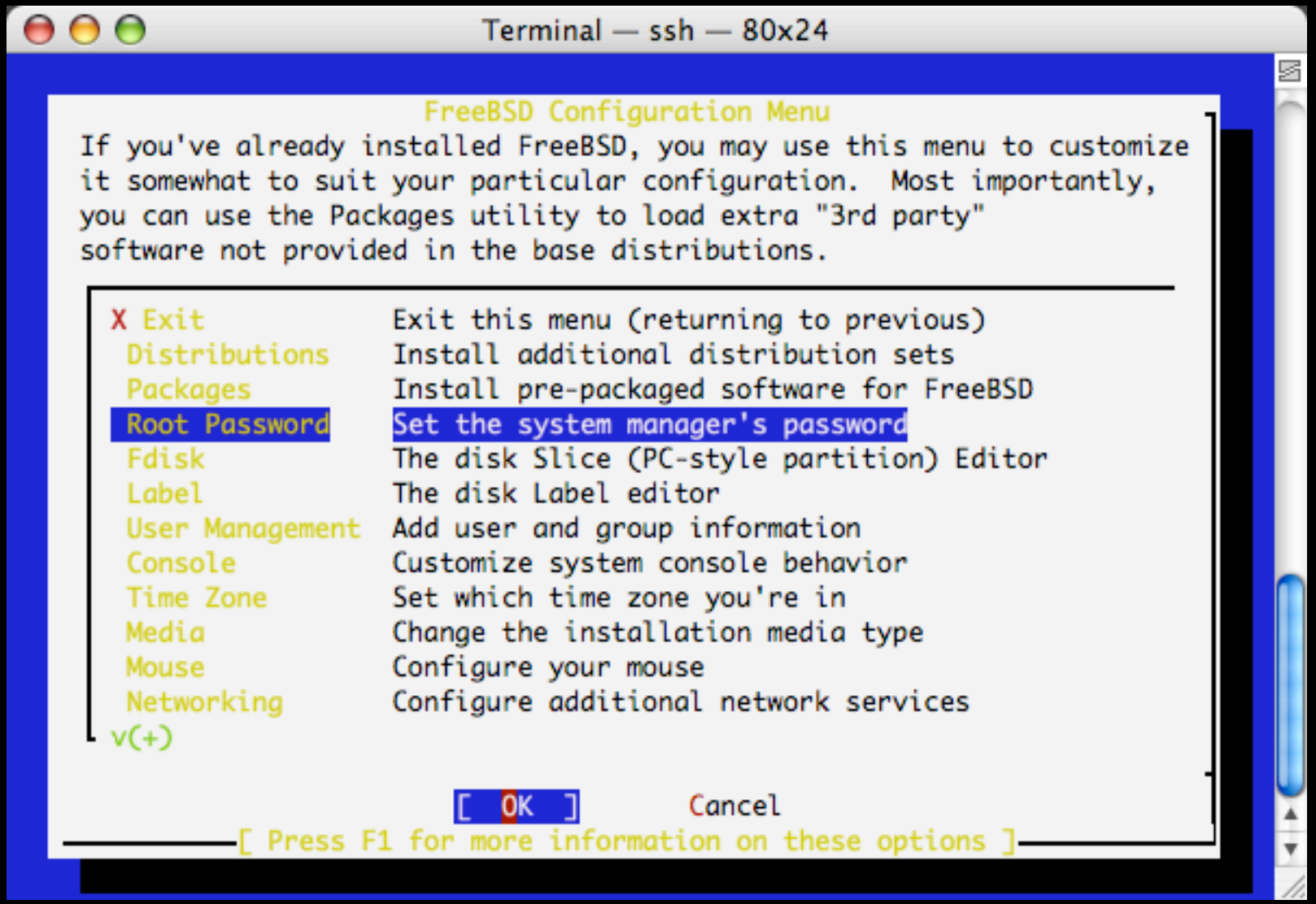
configure - call jailed sh

sysctl, whee!



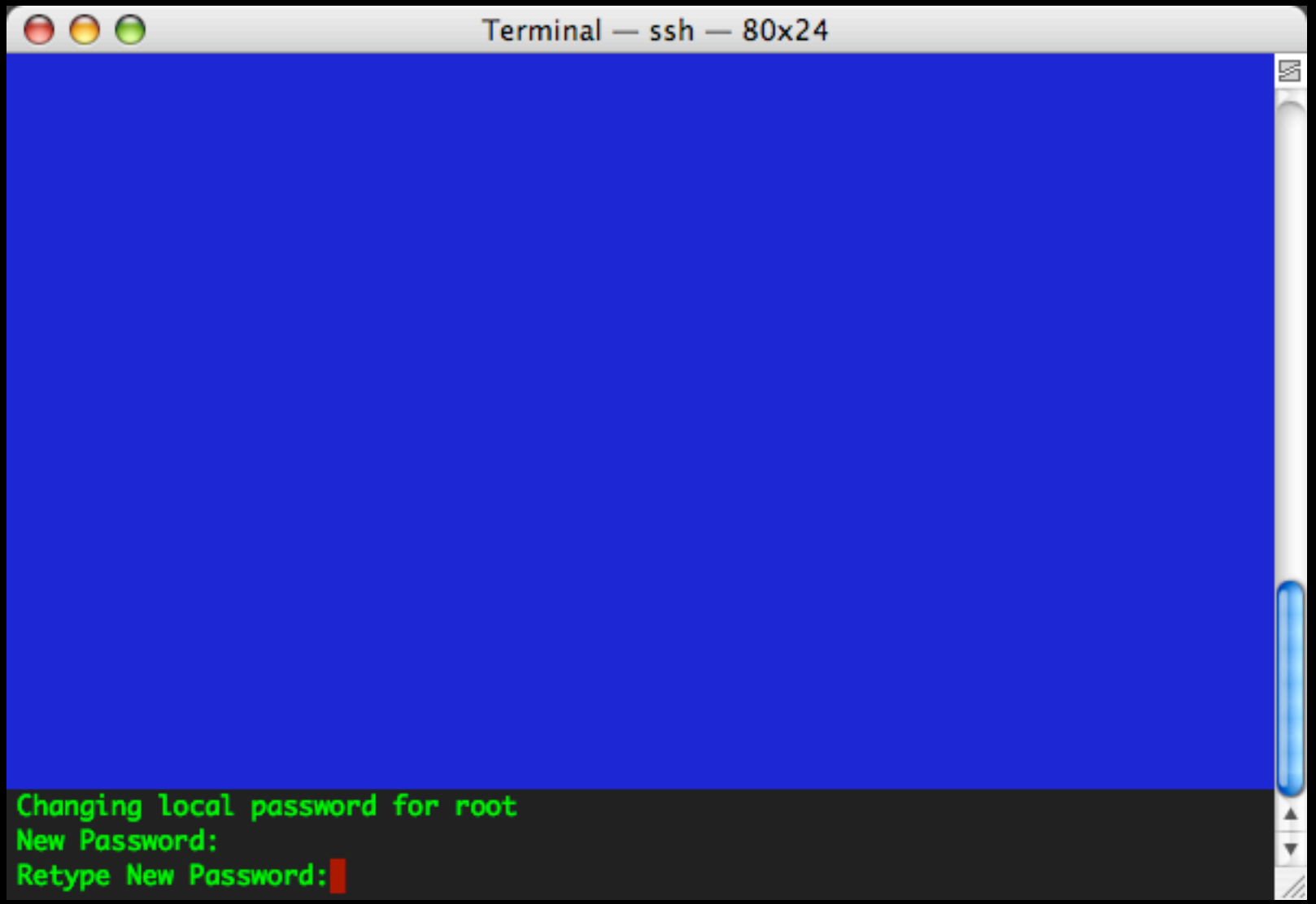
configure - call jailed sh

root pw



configure - call jailed sh

root pw

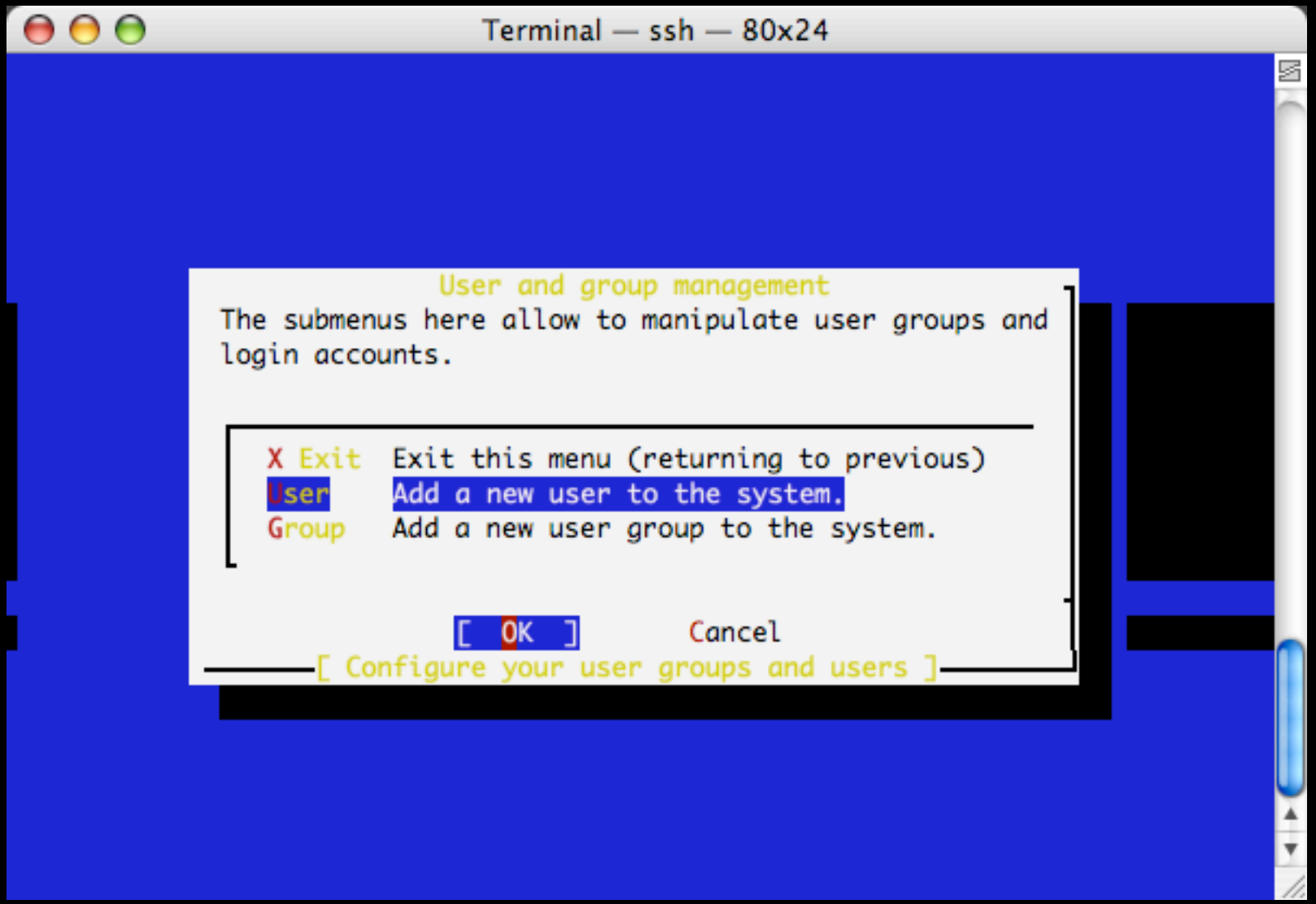


configure - call jailed sh

add users

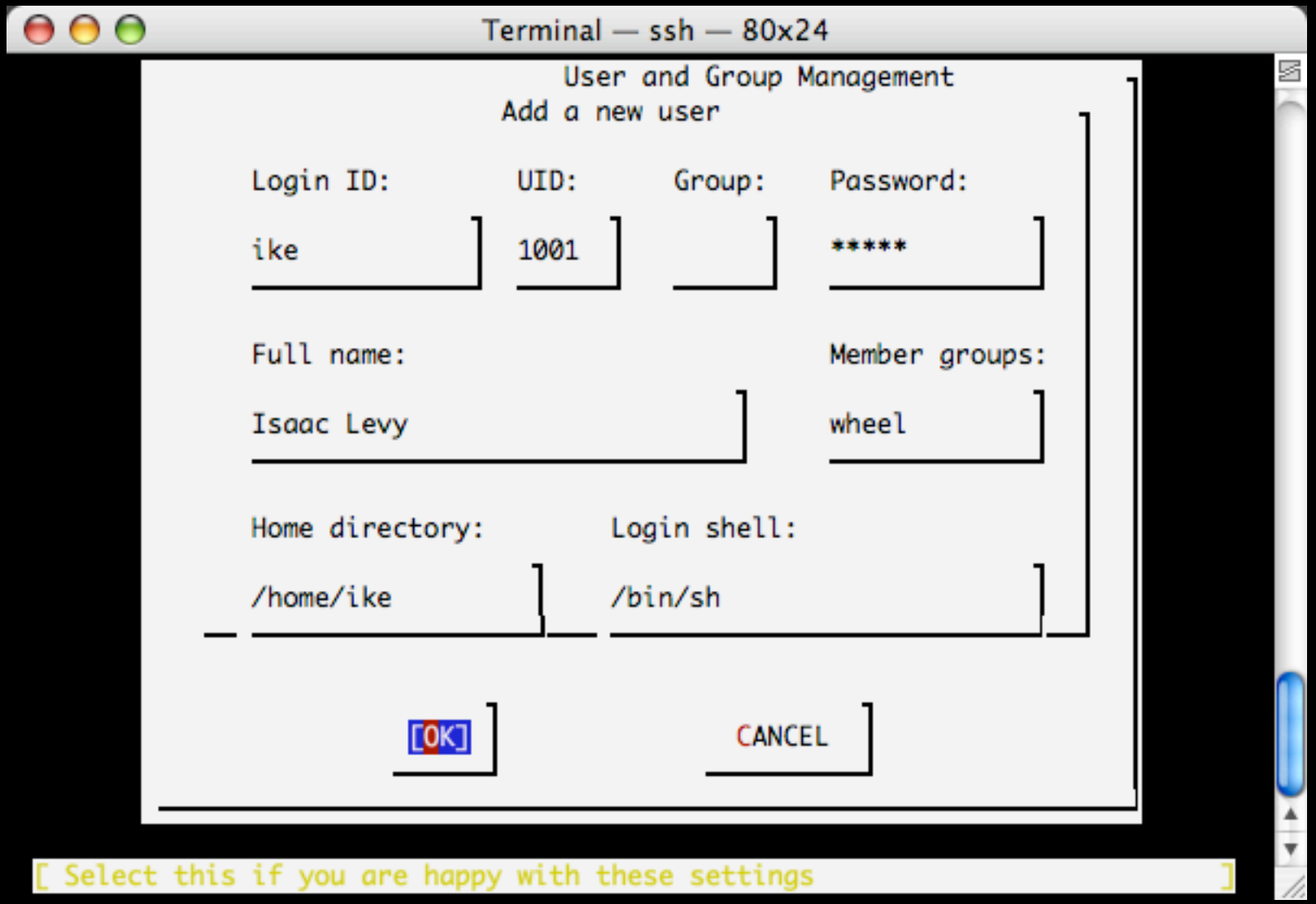
configure - call jailed sh

add users



configure - call jailed sh

add users

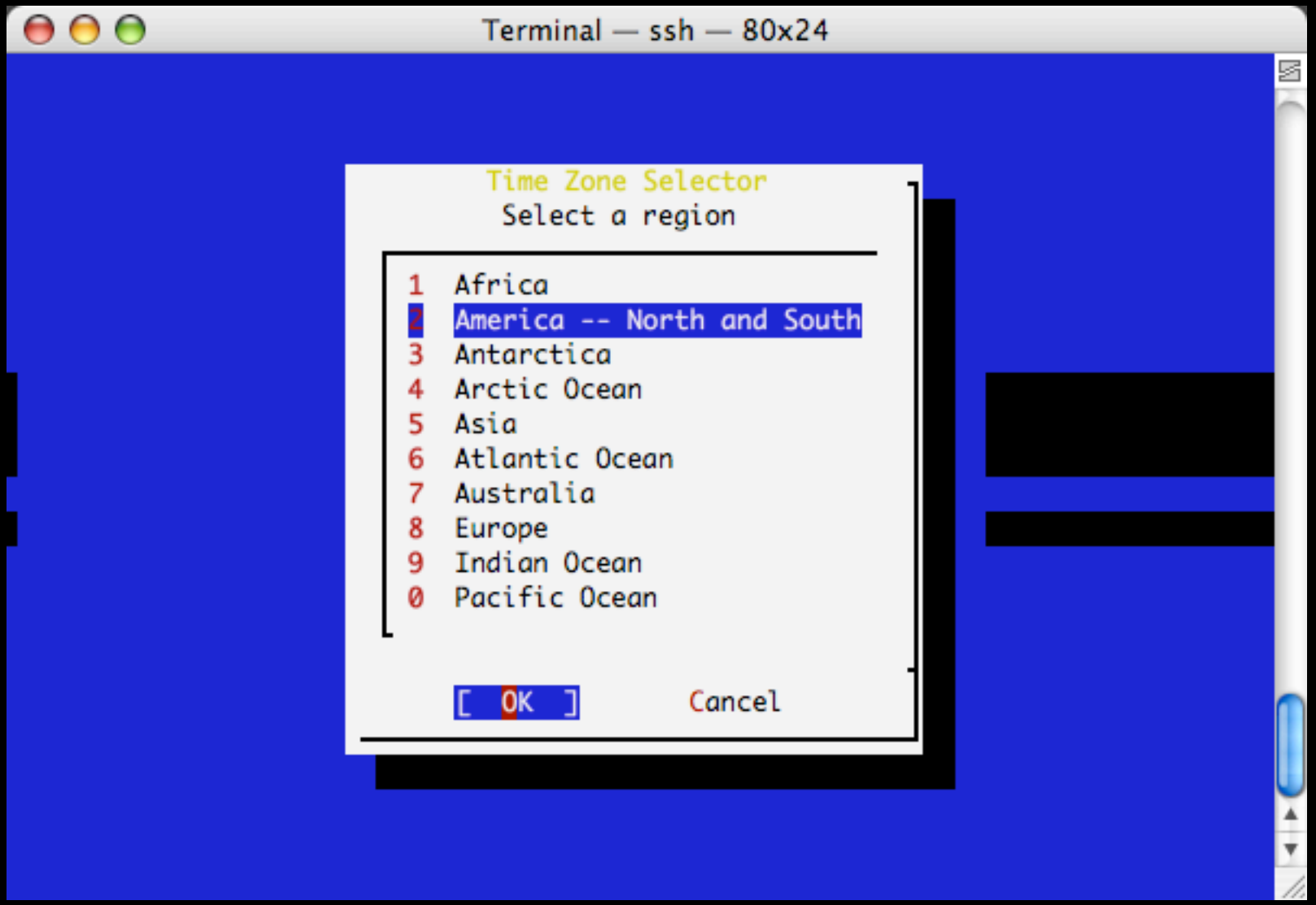


configure - call jailed sh

set timezone

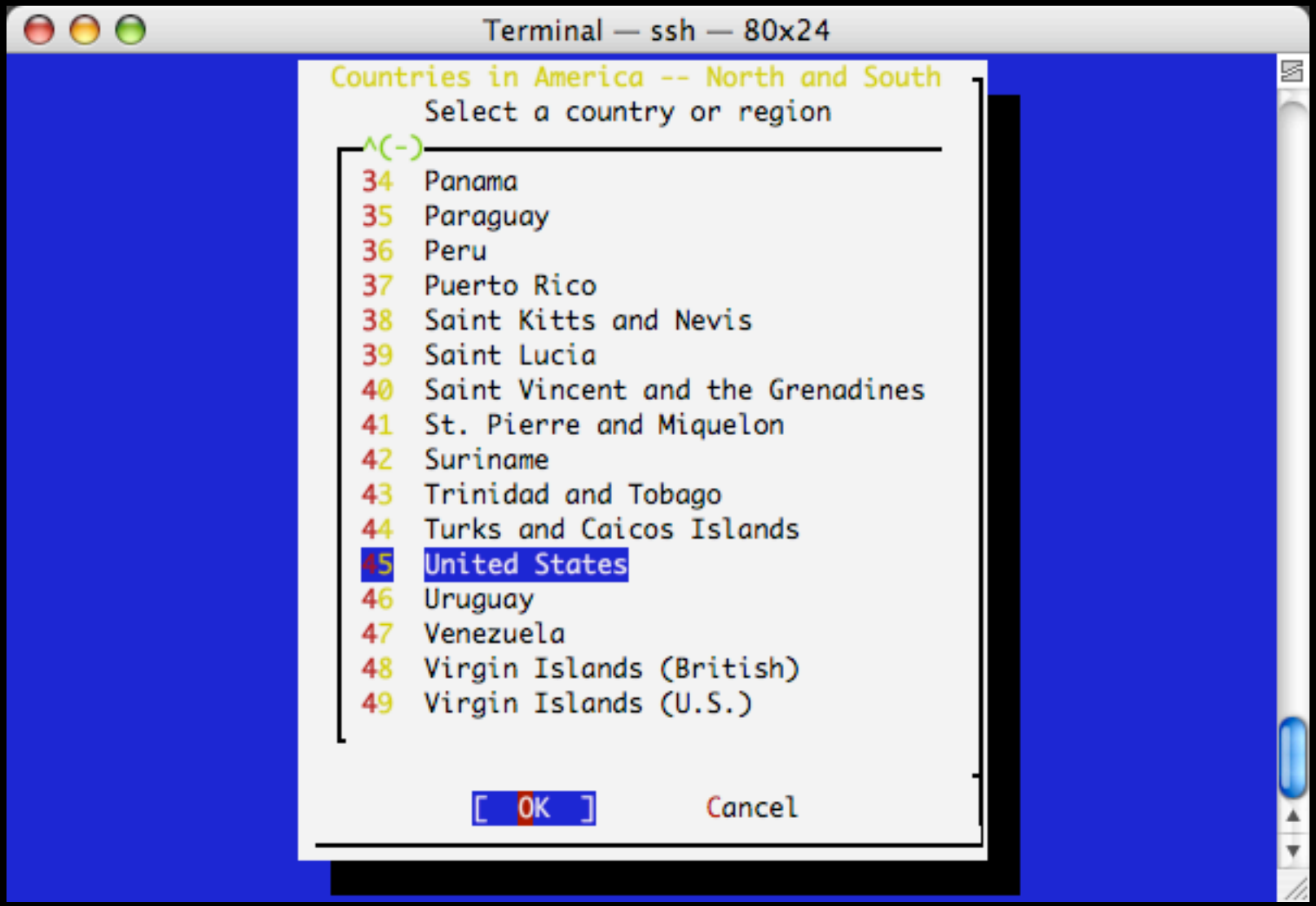
configure - call jailed sh

set timezone



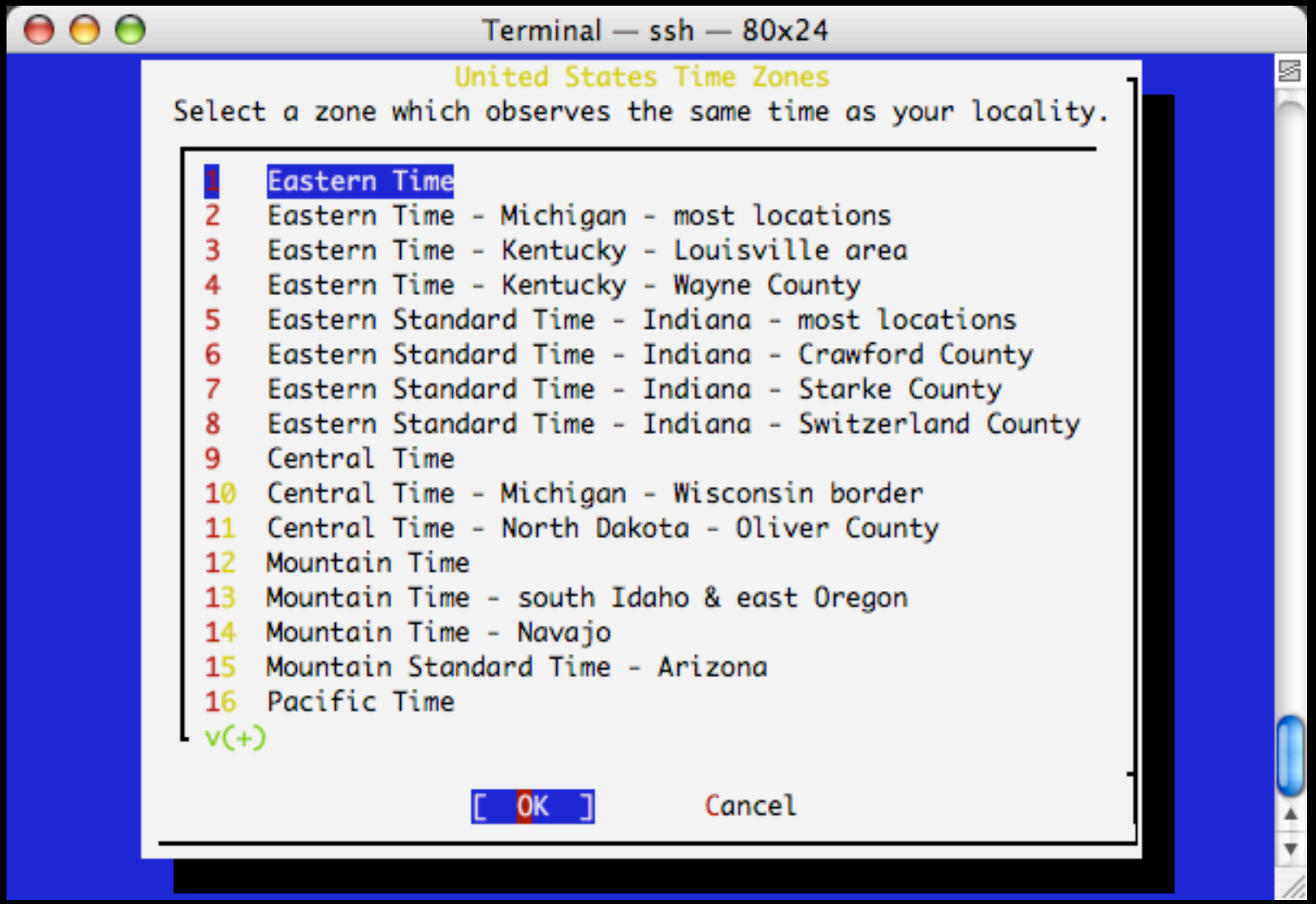
configure - call jailed sh

set timezone



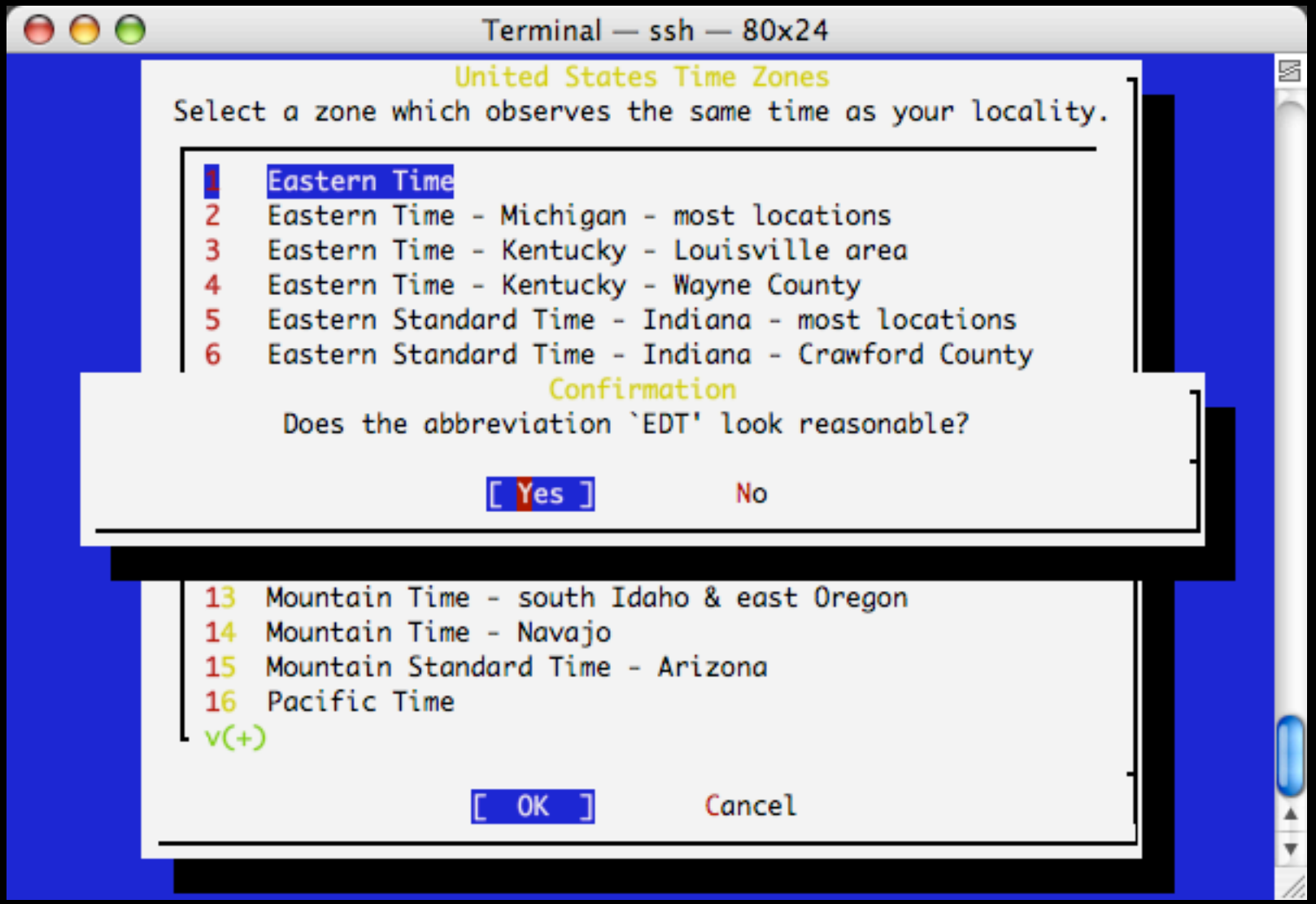
configure - call jailed sh

set timezone



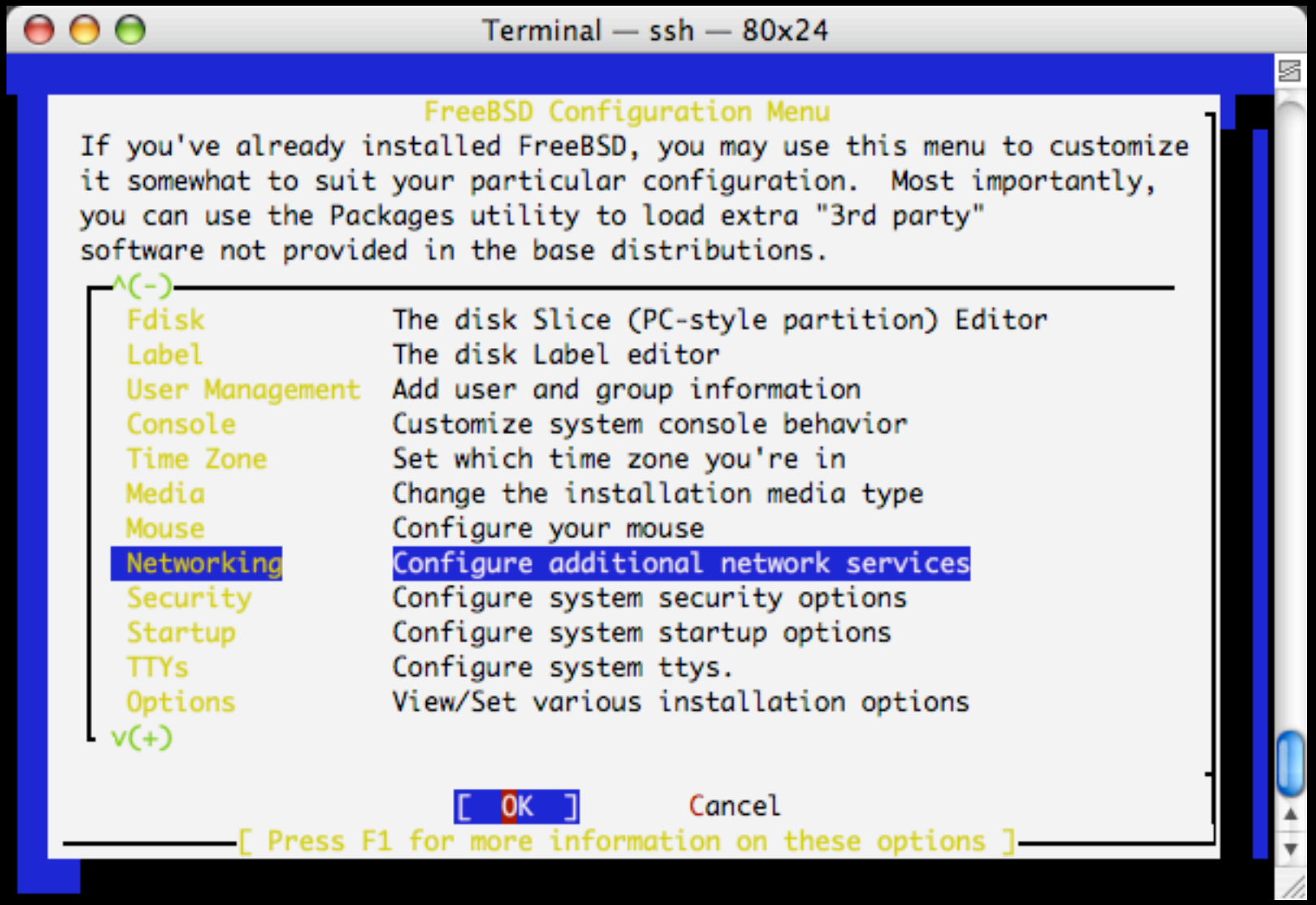
configure - call jailed sh

set timezone



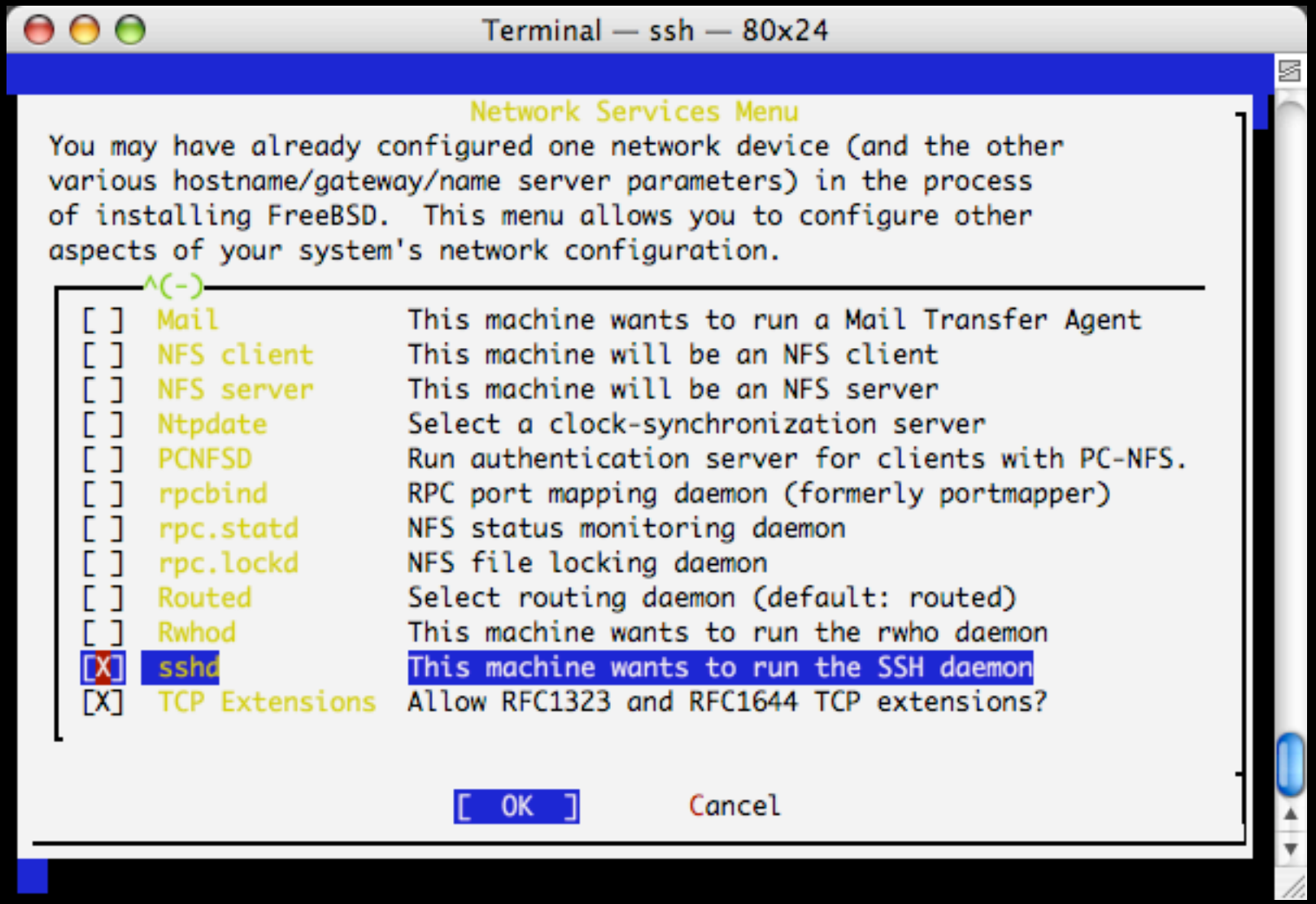
configure - call jailed sh

network options...



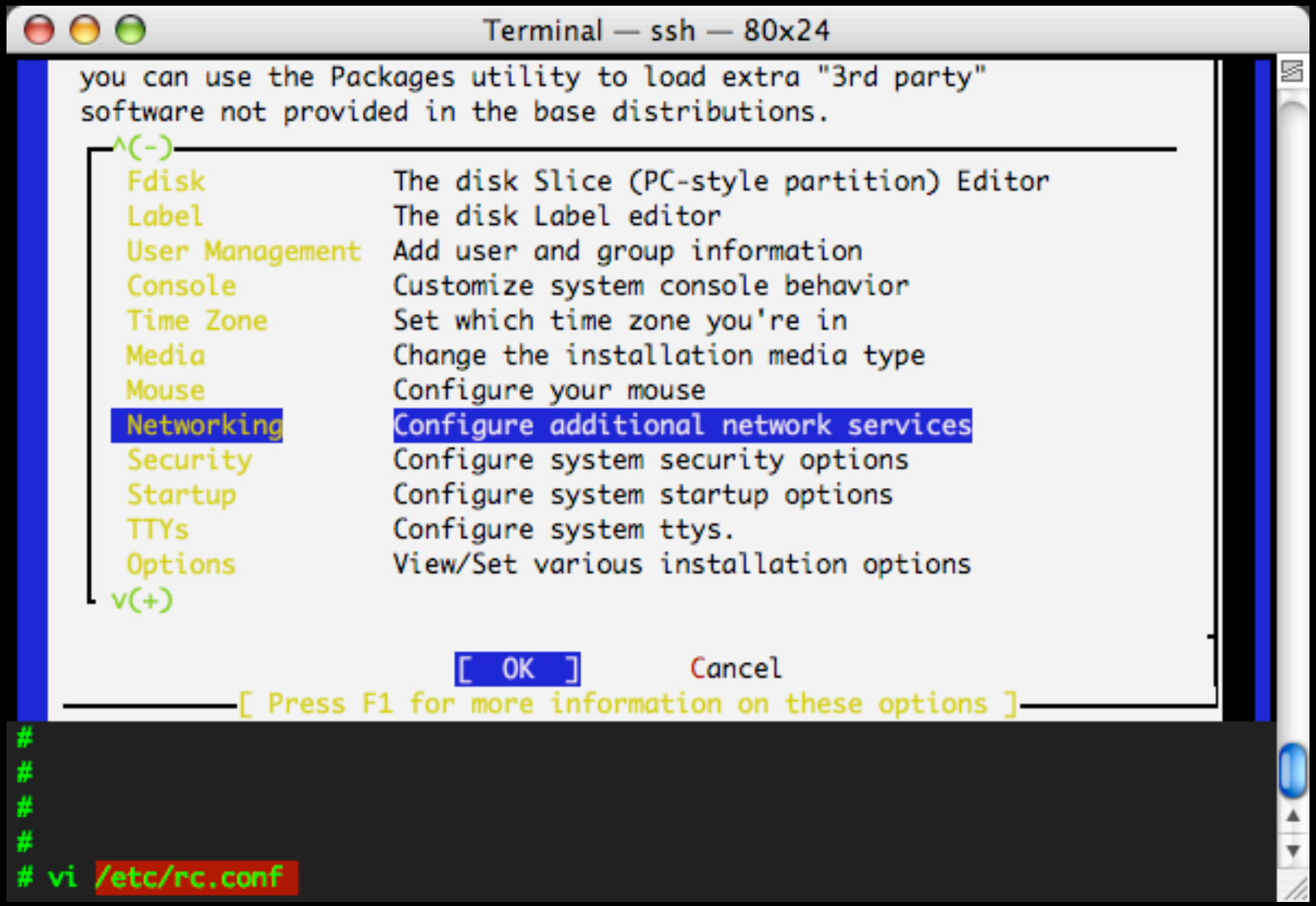
configure - call jailed sh

run ssh, **important**



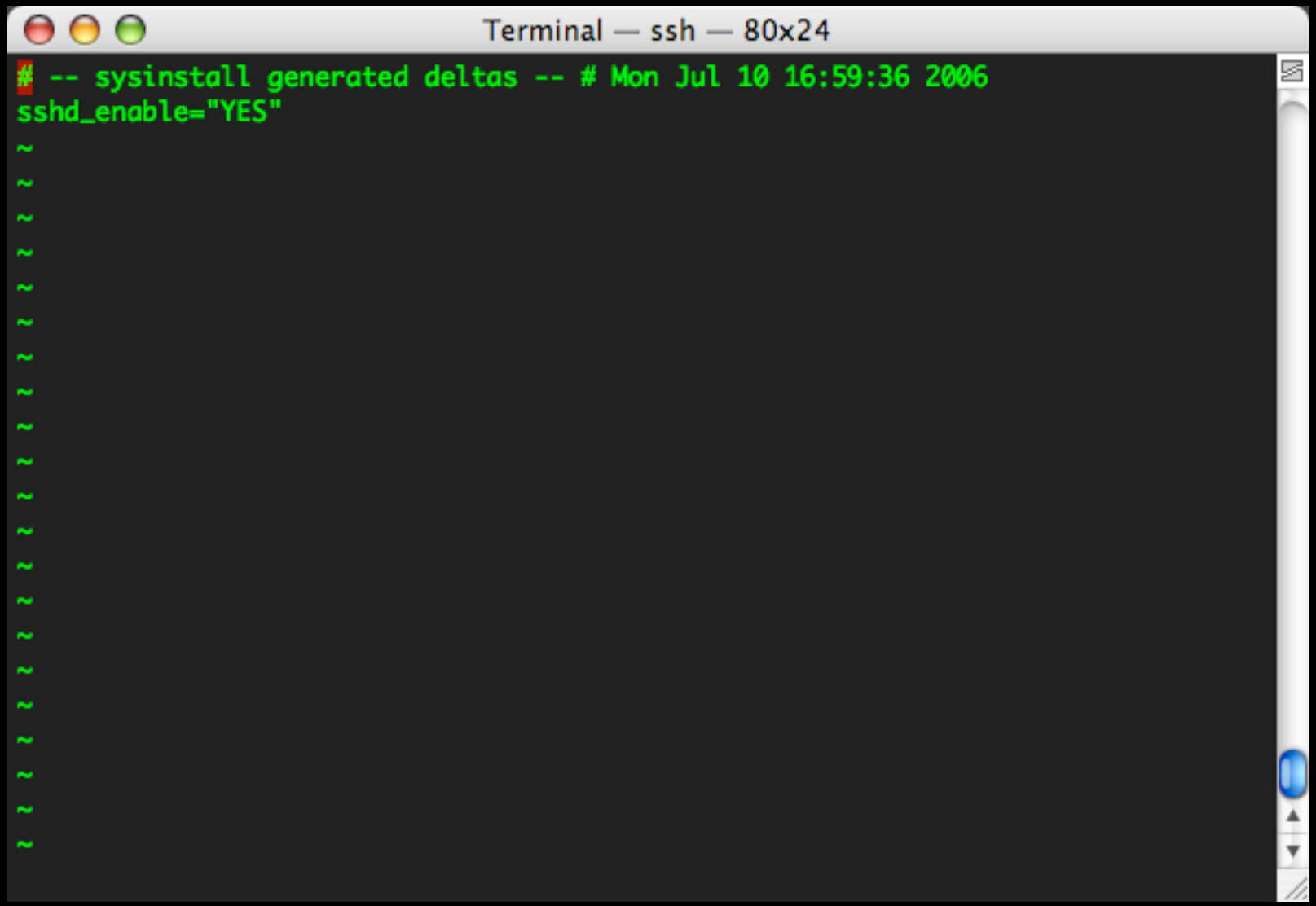
configure - call jailed sh

check rc.conf in jail



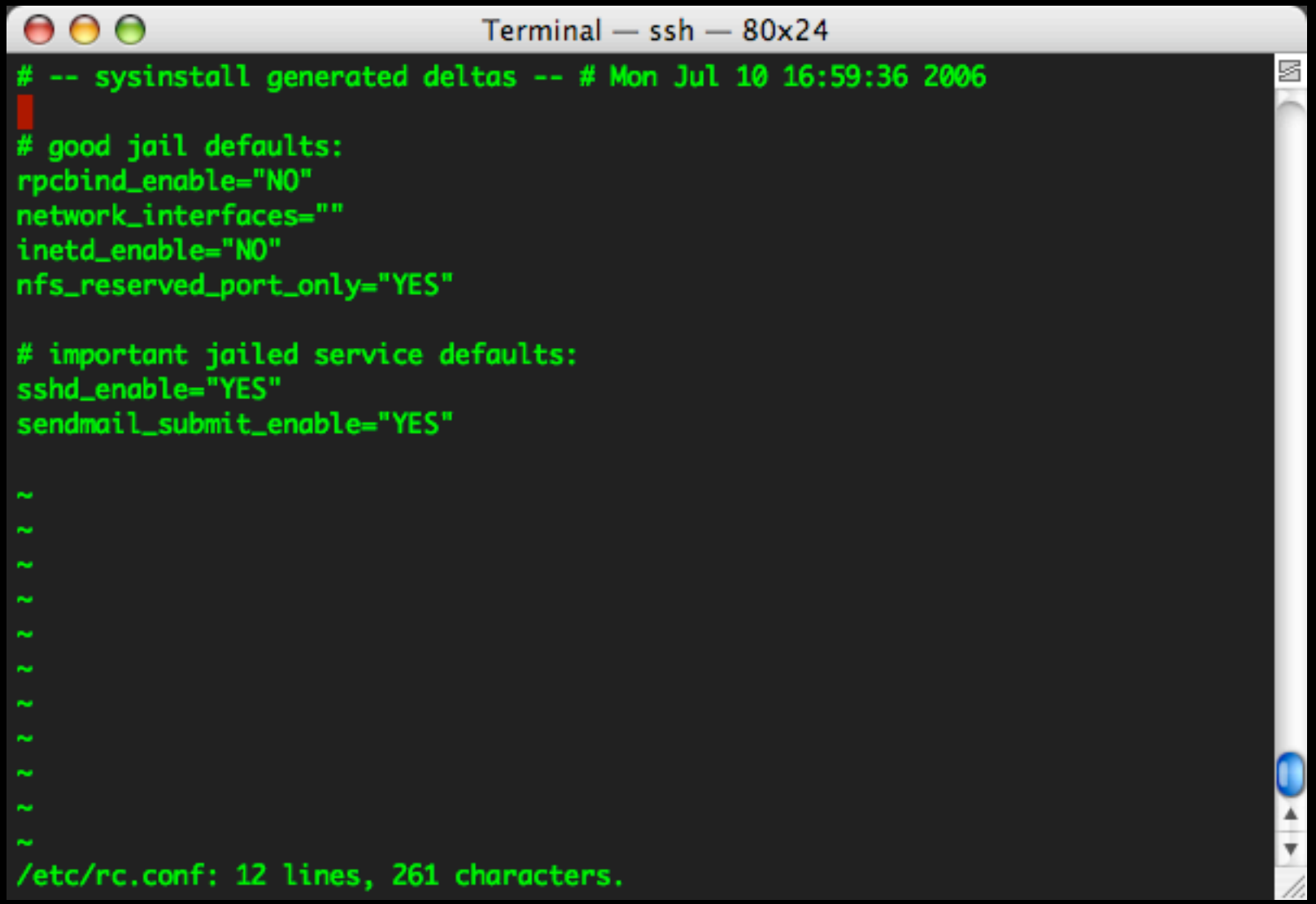
configure - call jailed sh

check rc.conf in jail

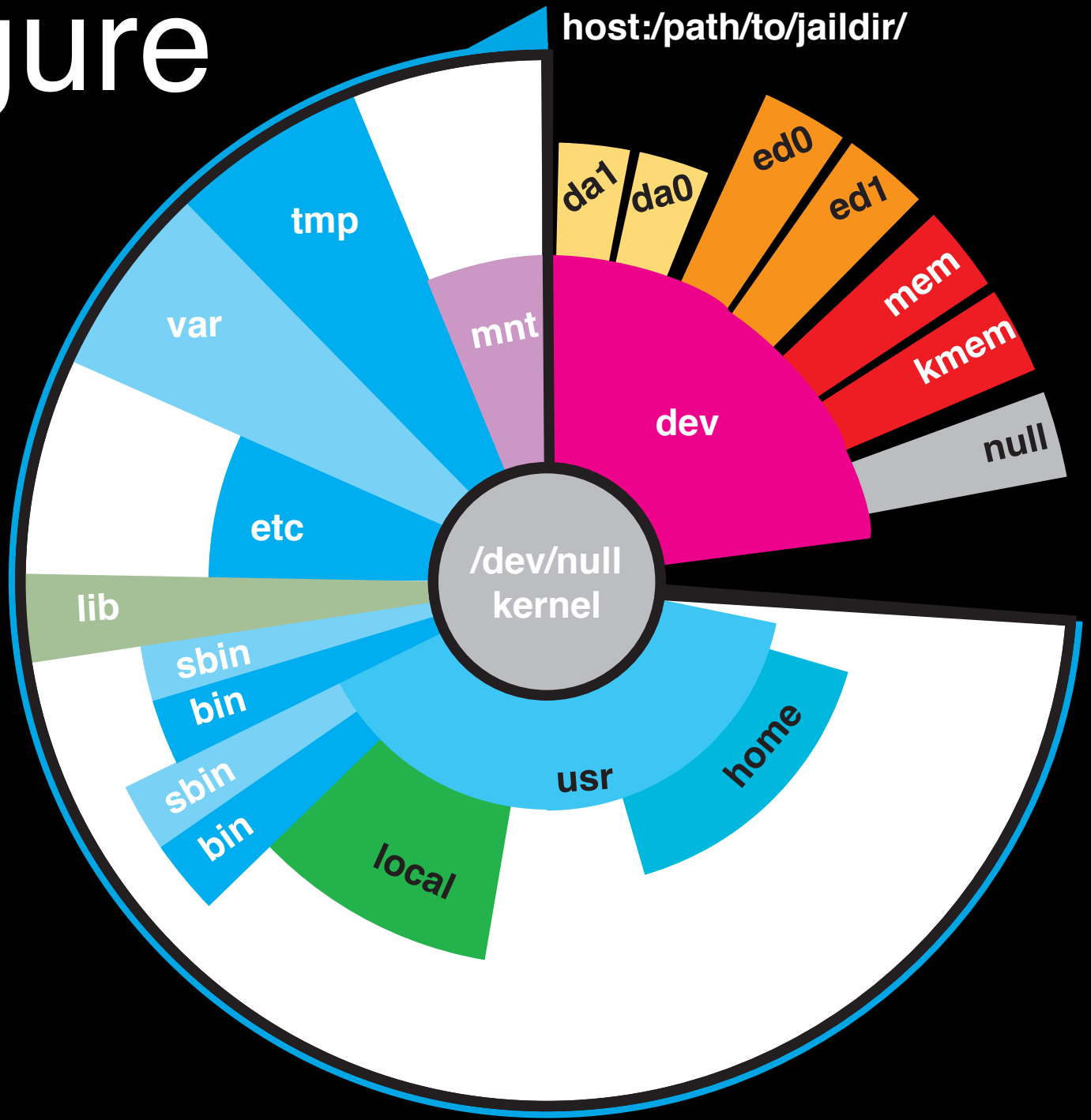


configure - call jailed sh

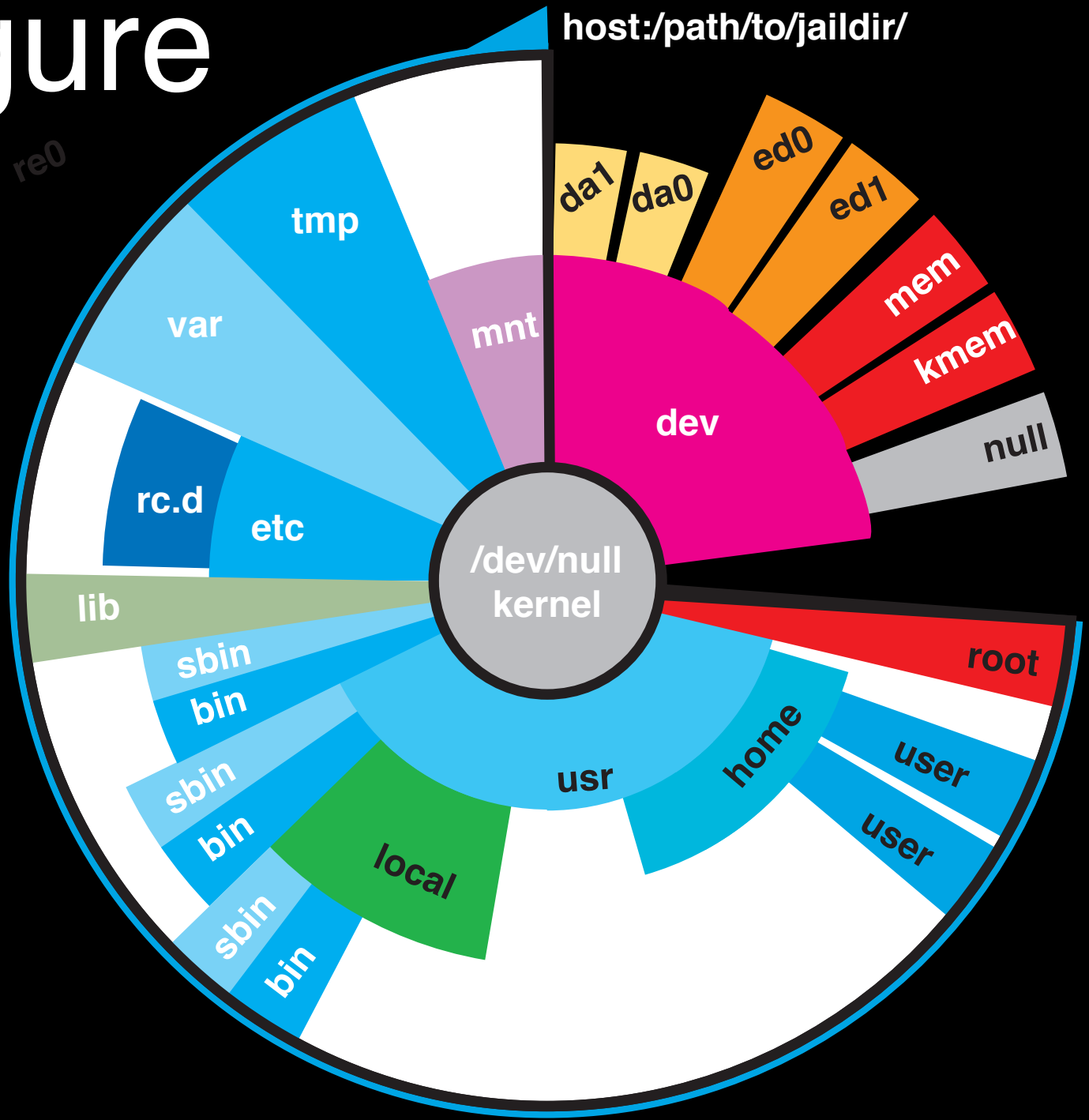
jail-specific stuff (just use common sense)



configure

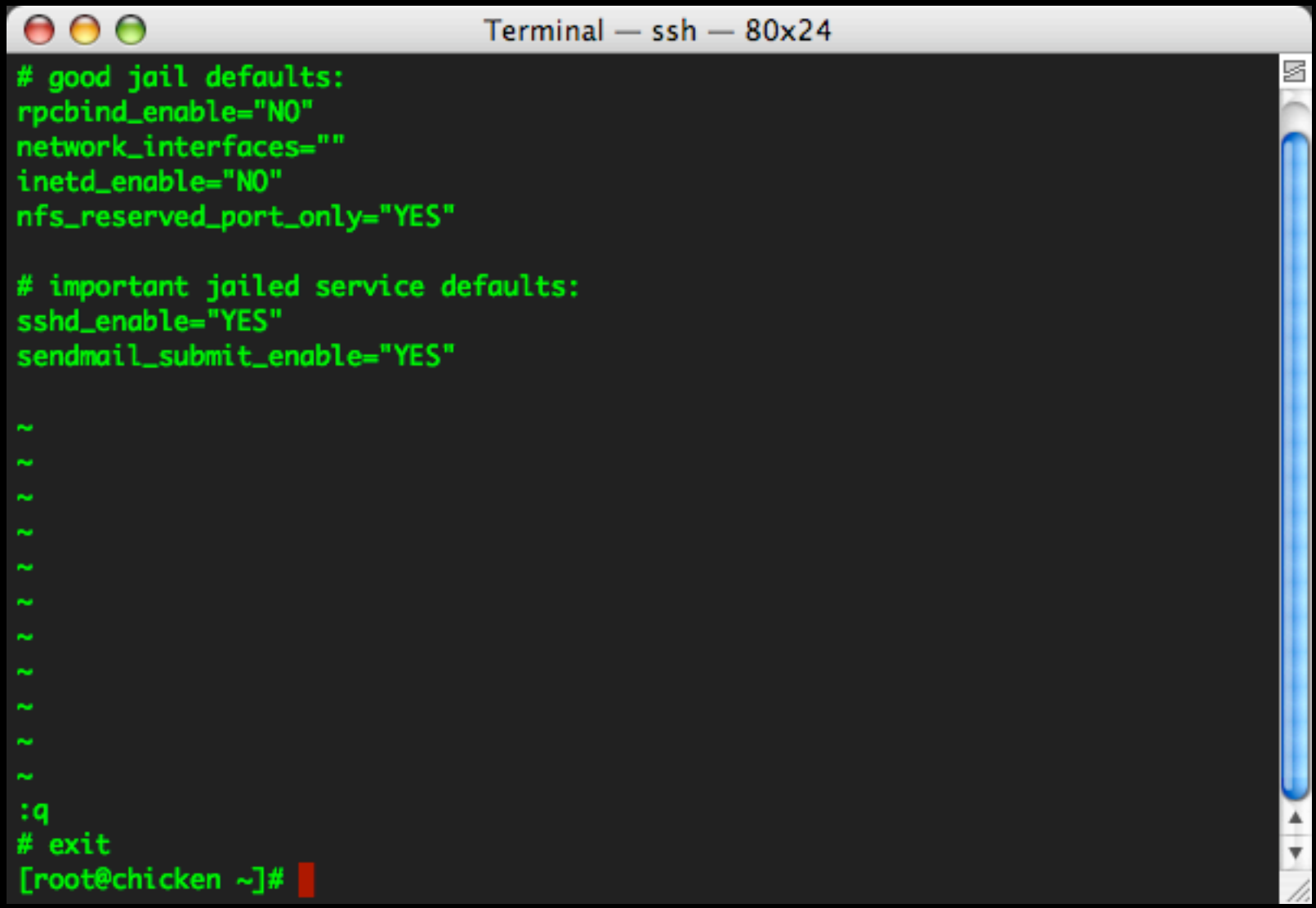

host:/path/to/jaildir/
da1
da0
ed0
ed1
mem
kmem
null
dev
mnt
tmp
var
etc
lib
sbin
bin
sbin
bin
local
usr
home
/dev/null
kernel

configure


host:/path/to/jaildir/
re0
tmp
var
mnt
da1
da0
ed0
ed1
mem
kmem
null
dev
rc.d
etc
/dev/null
kernel
lib
sbin
bin
sbin
bin
sbin
bin
local
usr
home
root
user
user

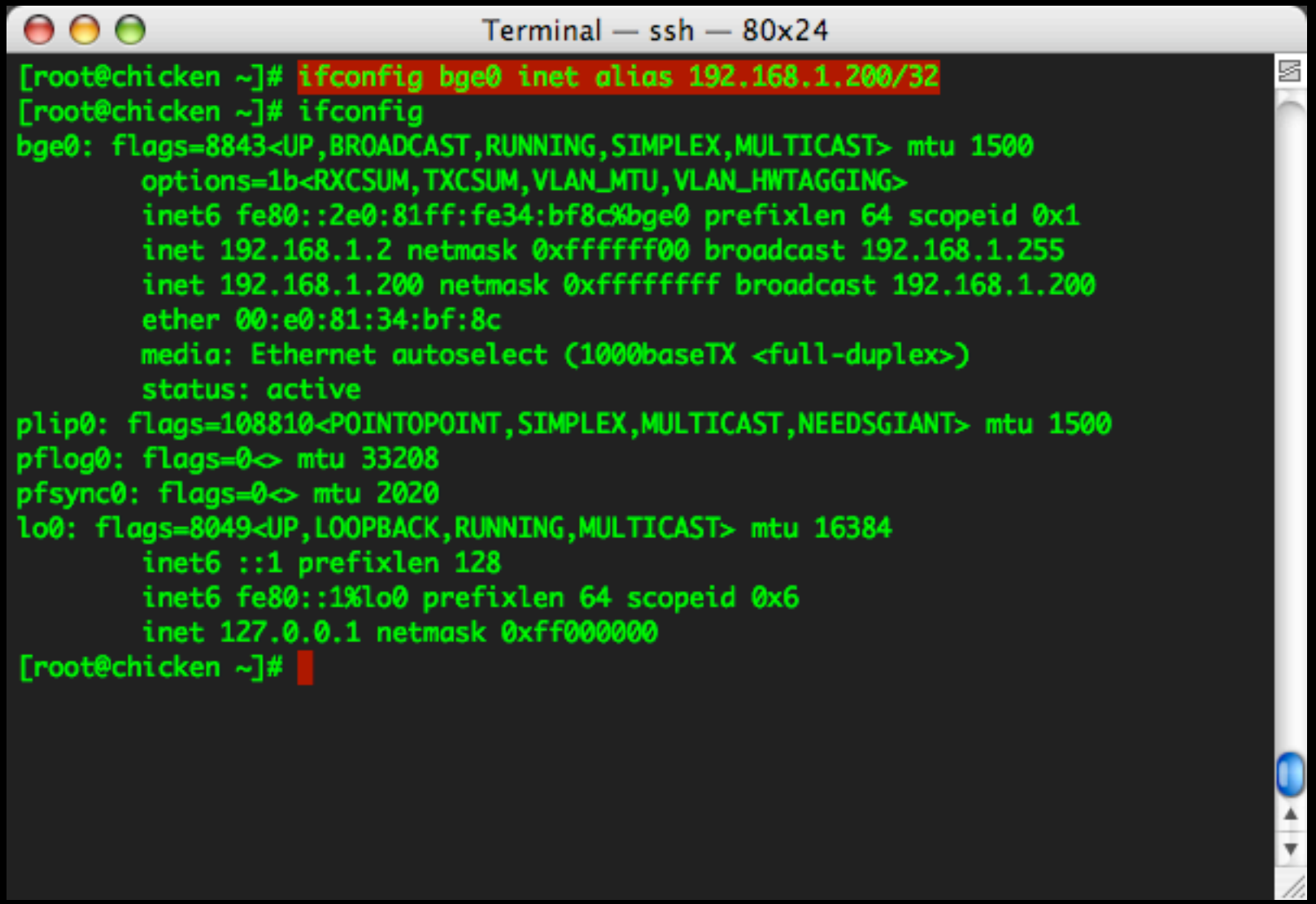
configure - call jailed sh

we’re finished configuring jailed system!



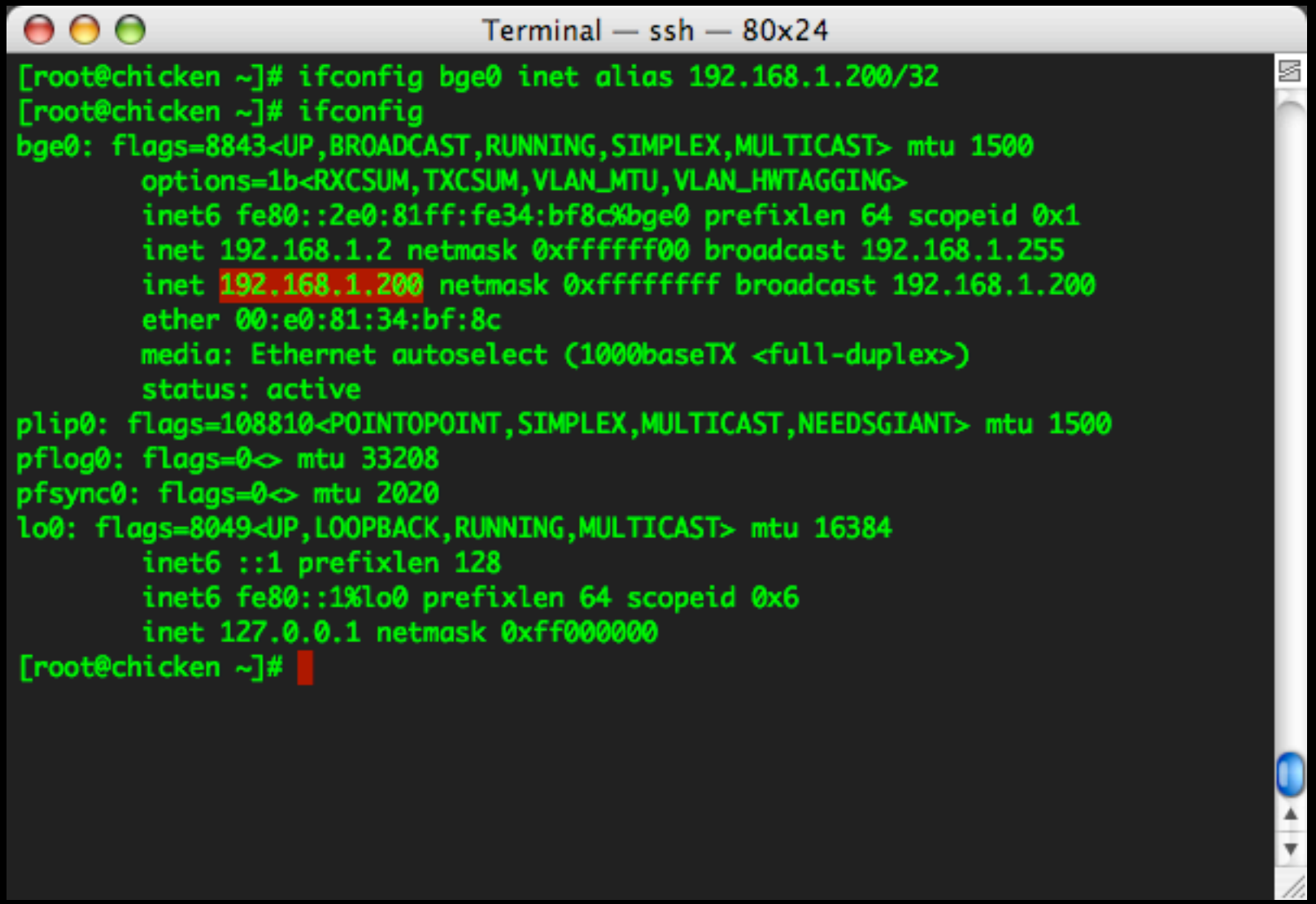
configure - assign ip alias

(use ifconfig)



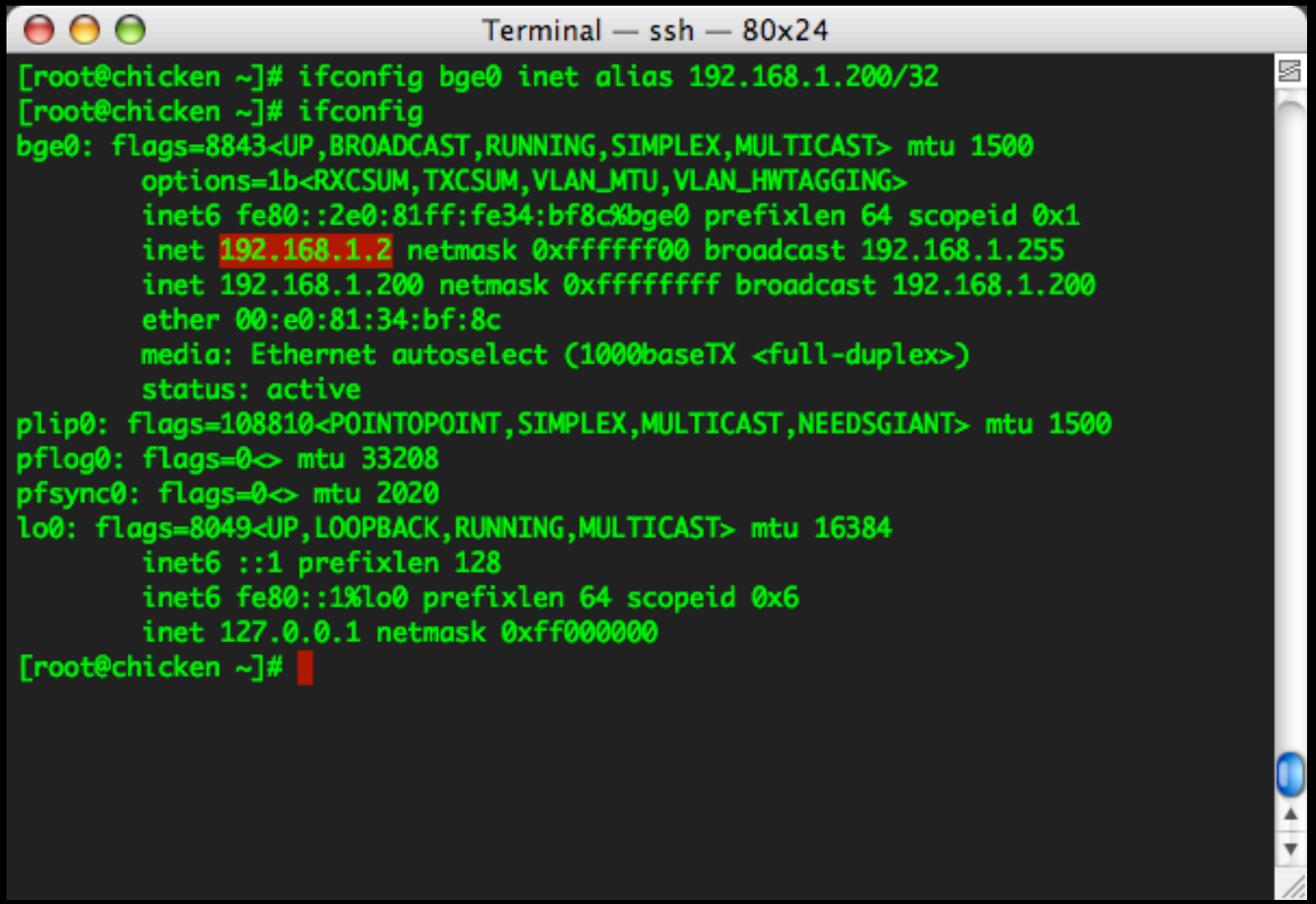
configure - assign ip alias

(ip for the jail)



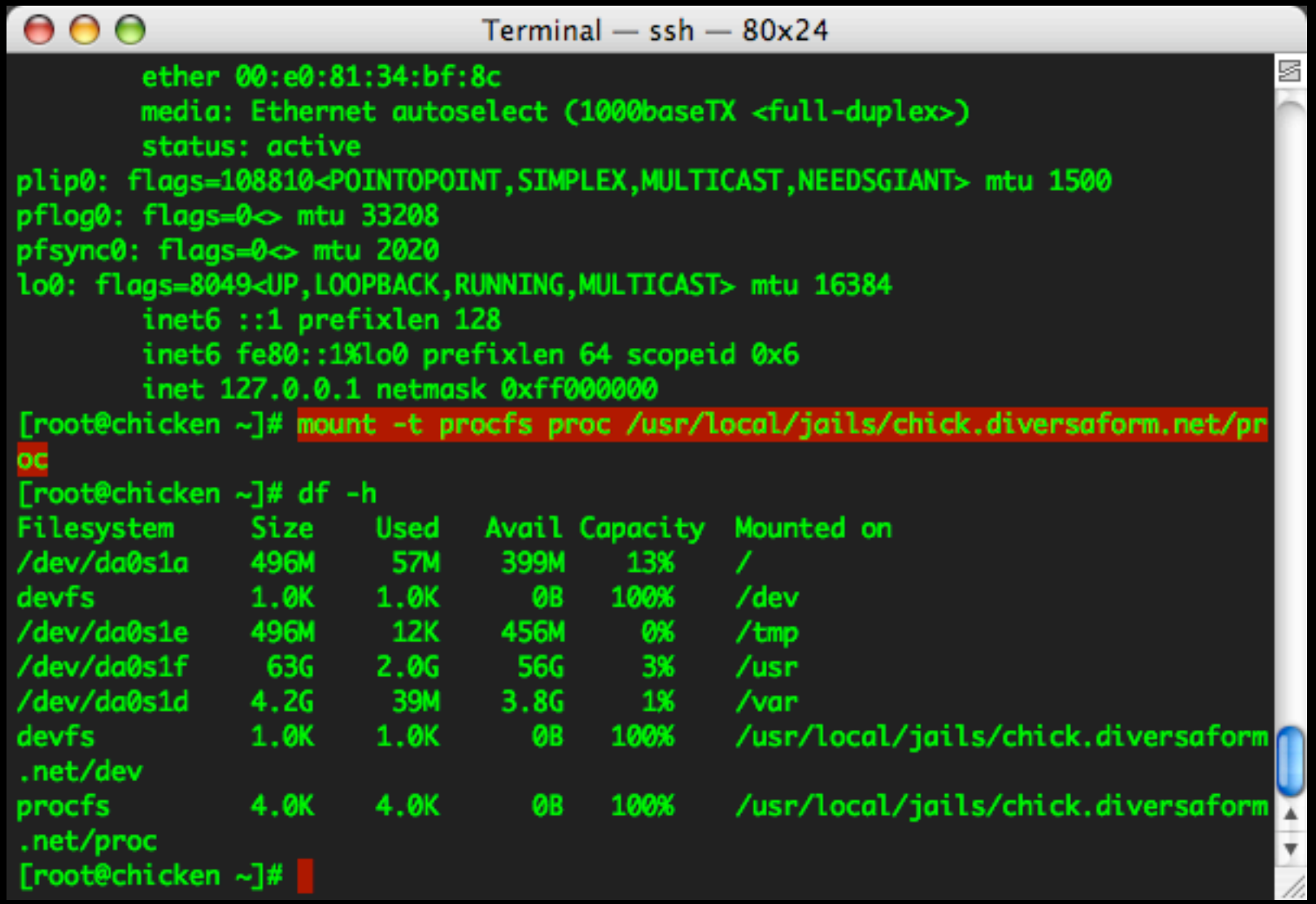
configure - assign ip alias

(original ip for the host machine)



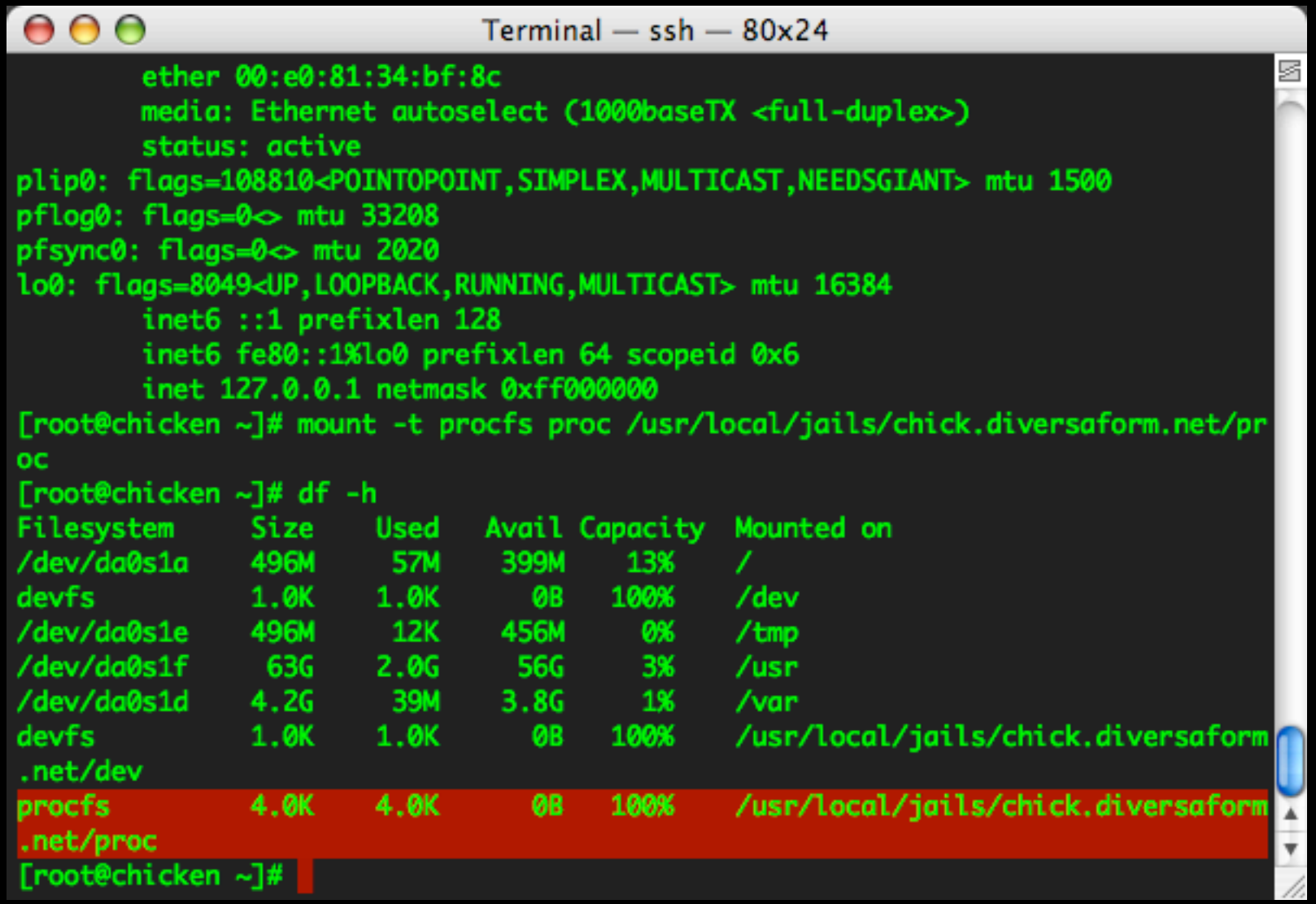
configure - call jailed sh

(analagous to booting a machine in su mode)

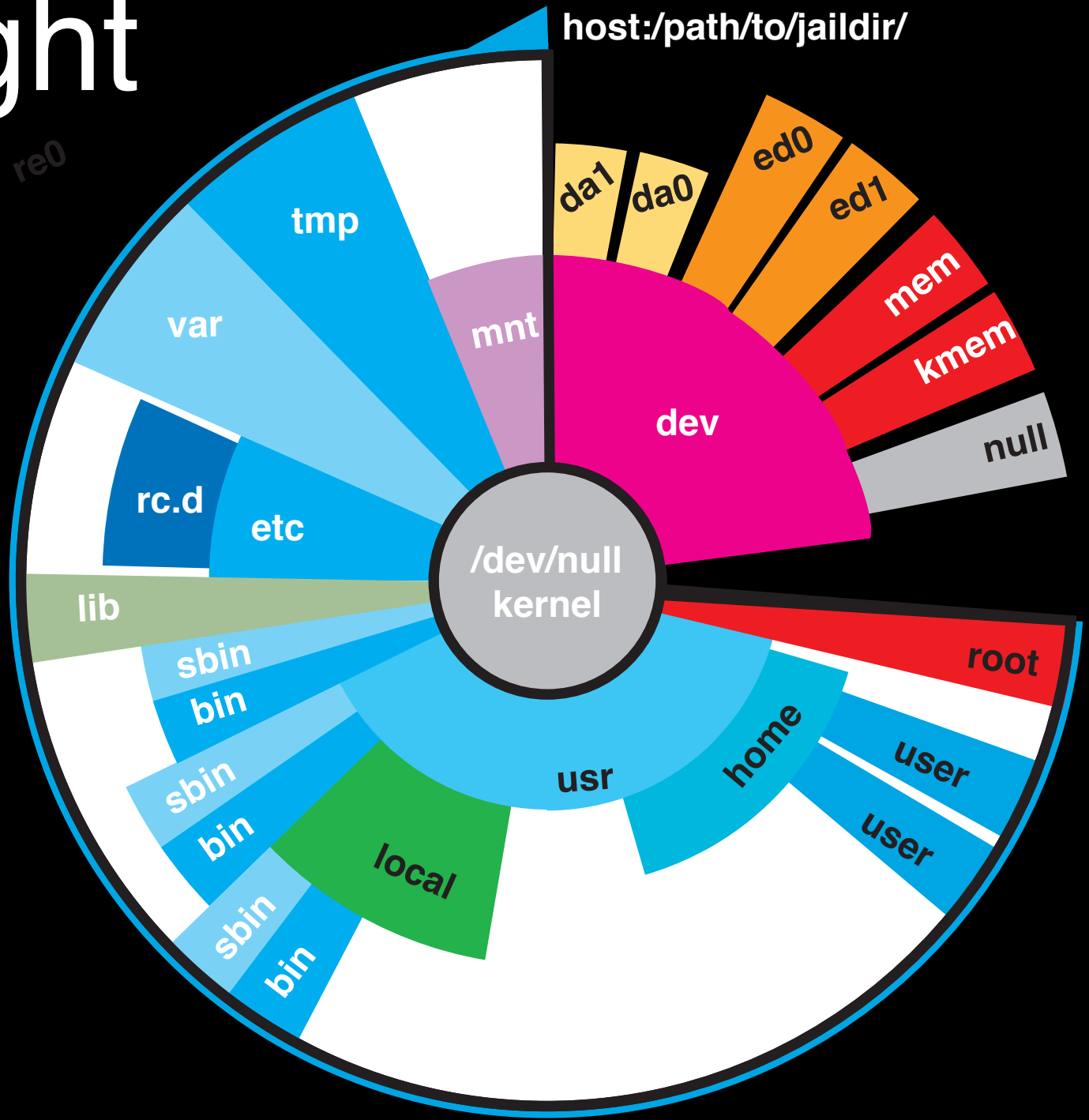


configure - call jailed sh

(analagous to booting a machine in su mode)



preflight

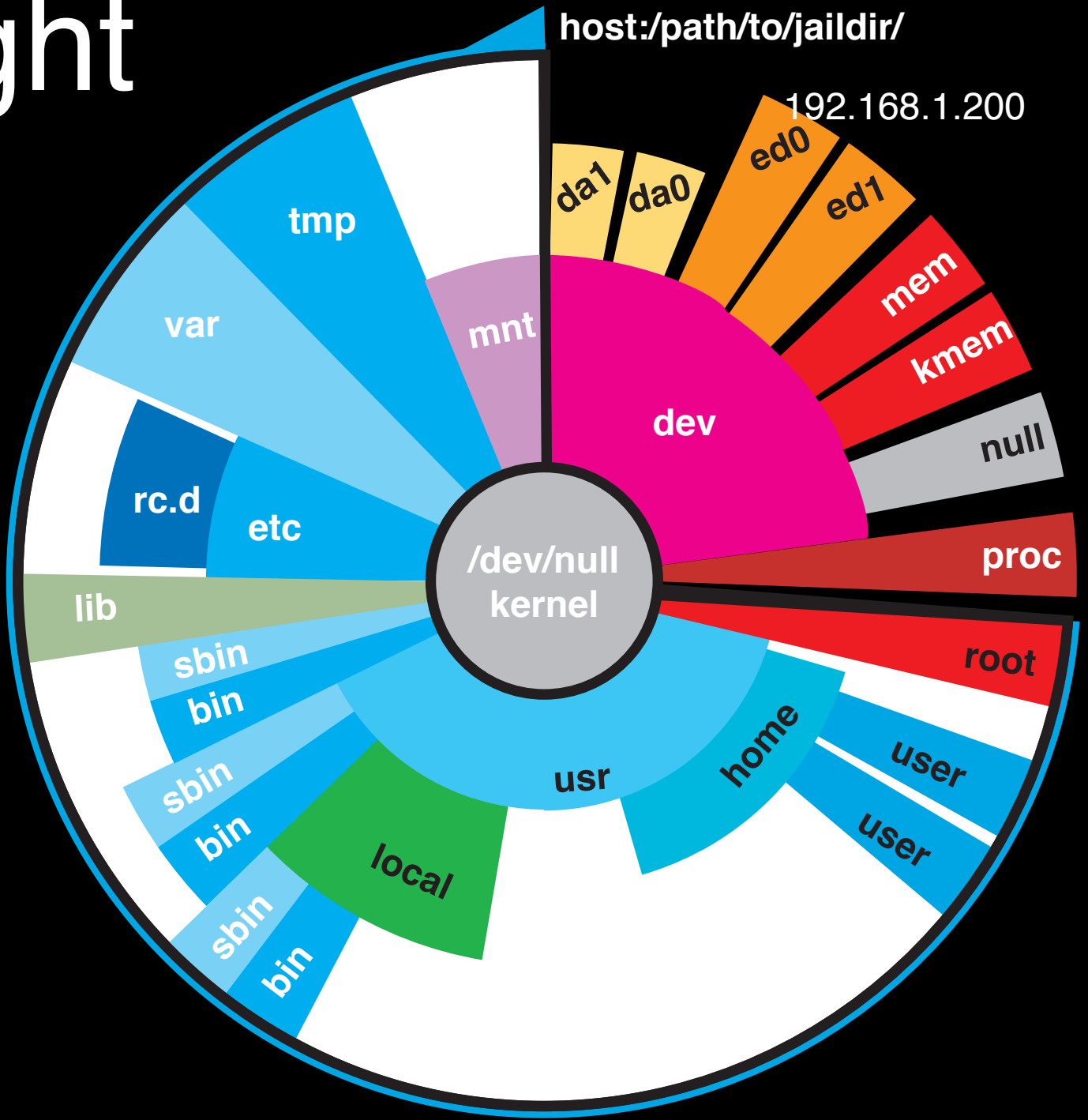

host:/path/to/jaildir/
re0
da1
da0
ed0
ed1
mem
kmem
null
dev
mnt
tmp
var
rc.d
etc
lib
/dev/null
kernel
root
user
user
home
usr
sbin
bin
sbin
bin
sbin
bin
local

preflight

re0

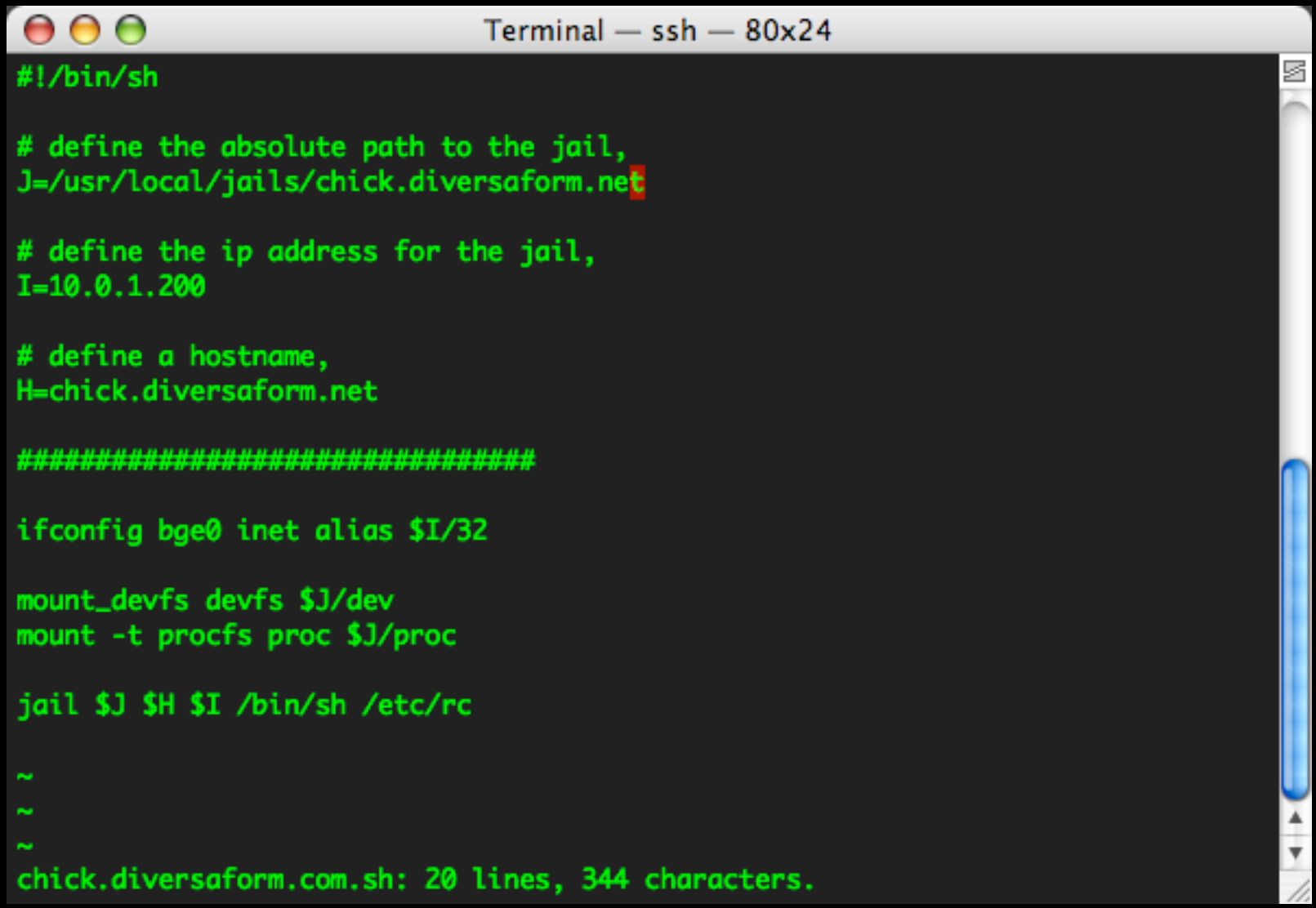
192.168.1.2

192.168.1.200
192.168.1.x



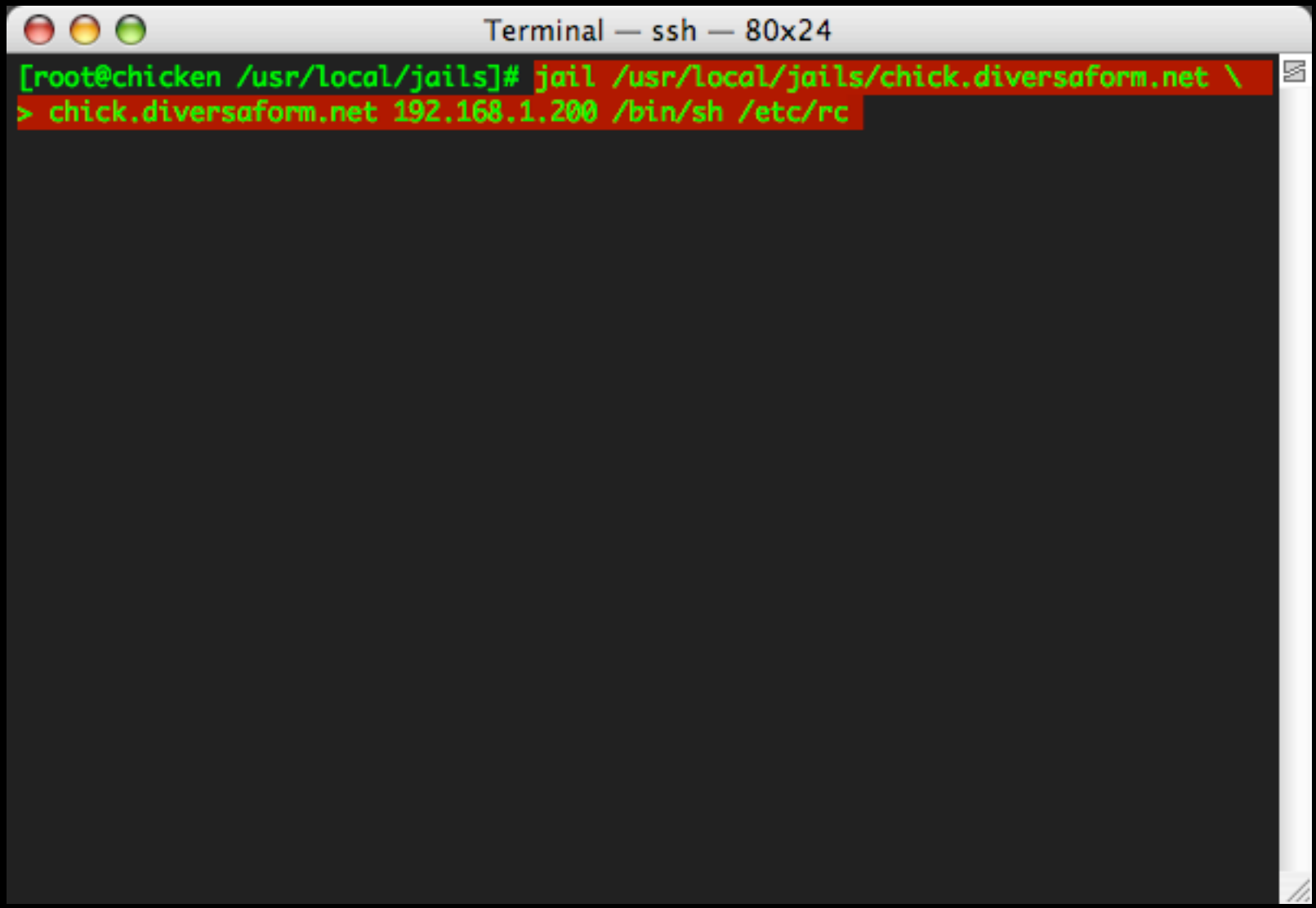
start tangent! (script),

remember how I said rc.d is usually a bad idea?



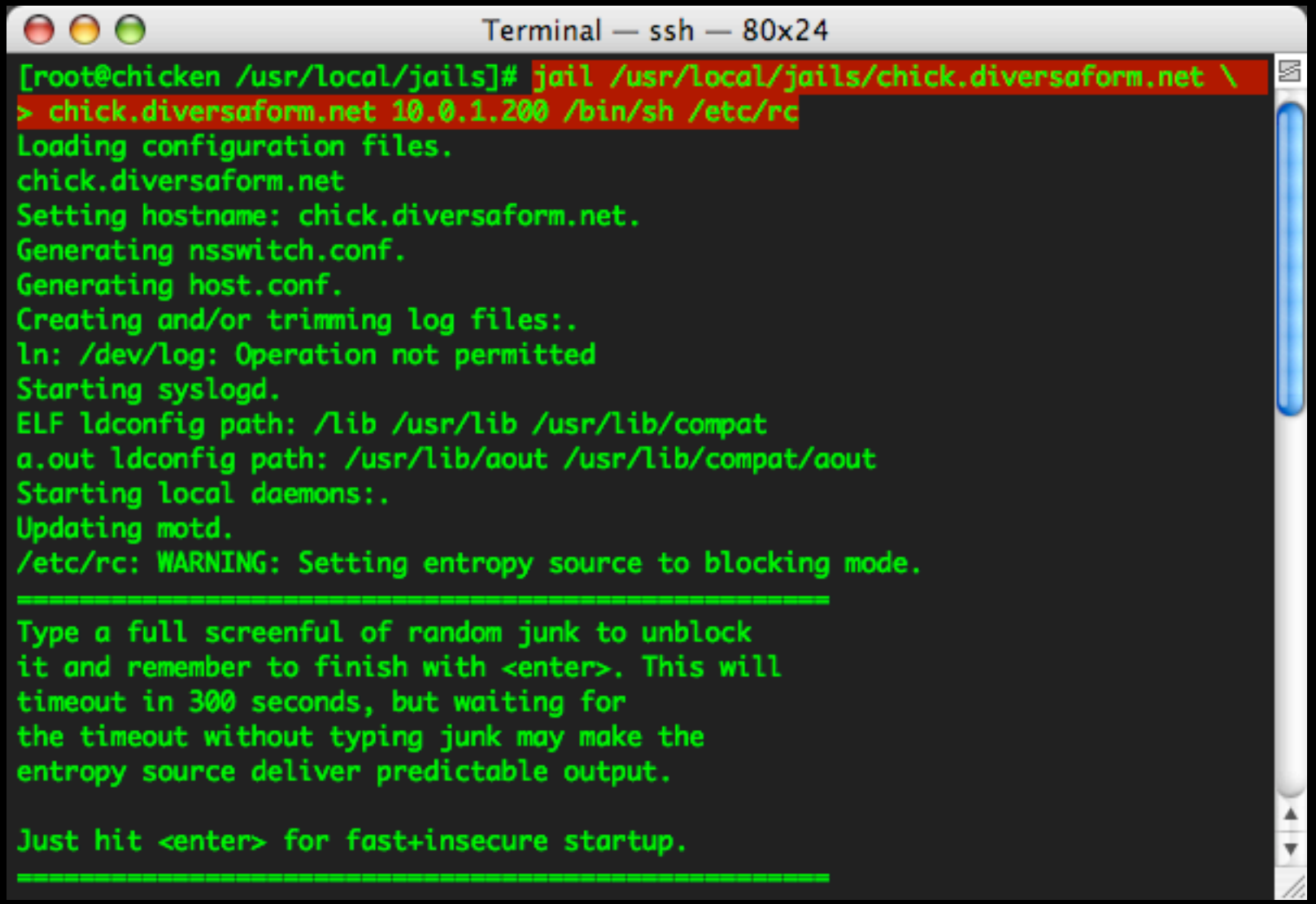
start!

we’re gonna start the jail manually here....



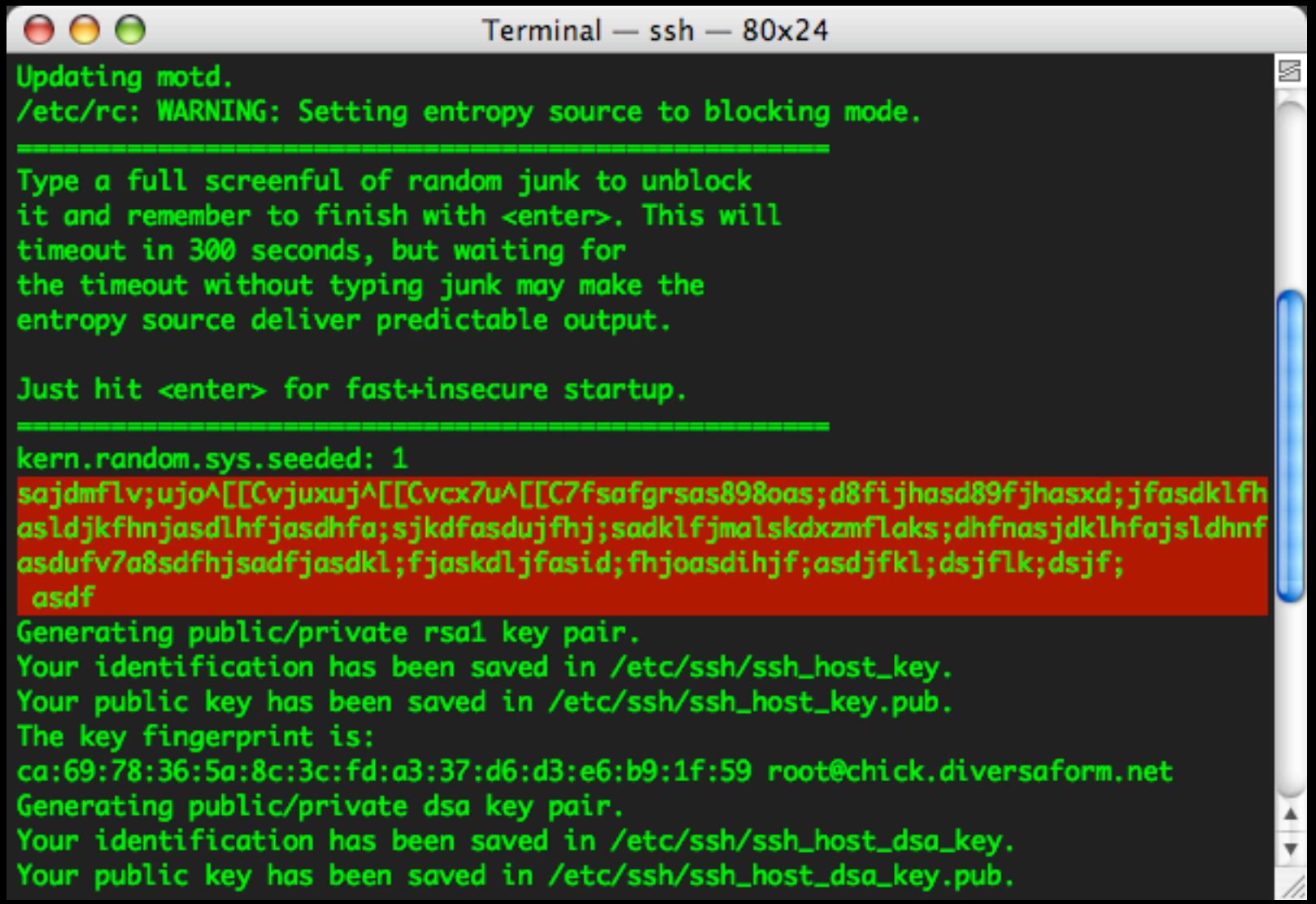
start!

we’re gonna start the jail manually here....



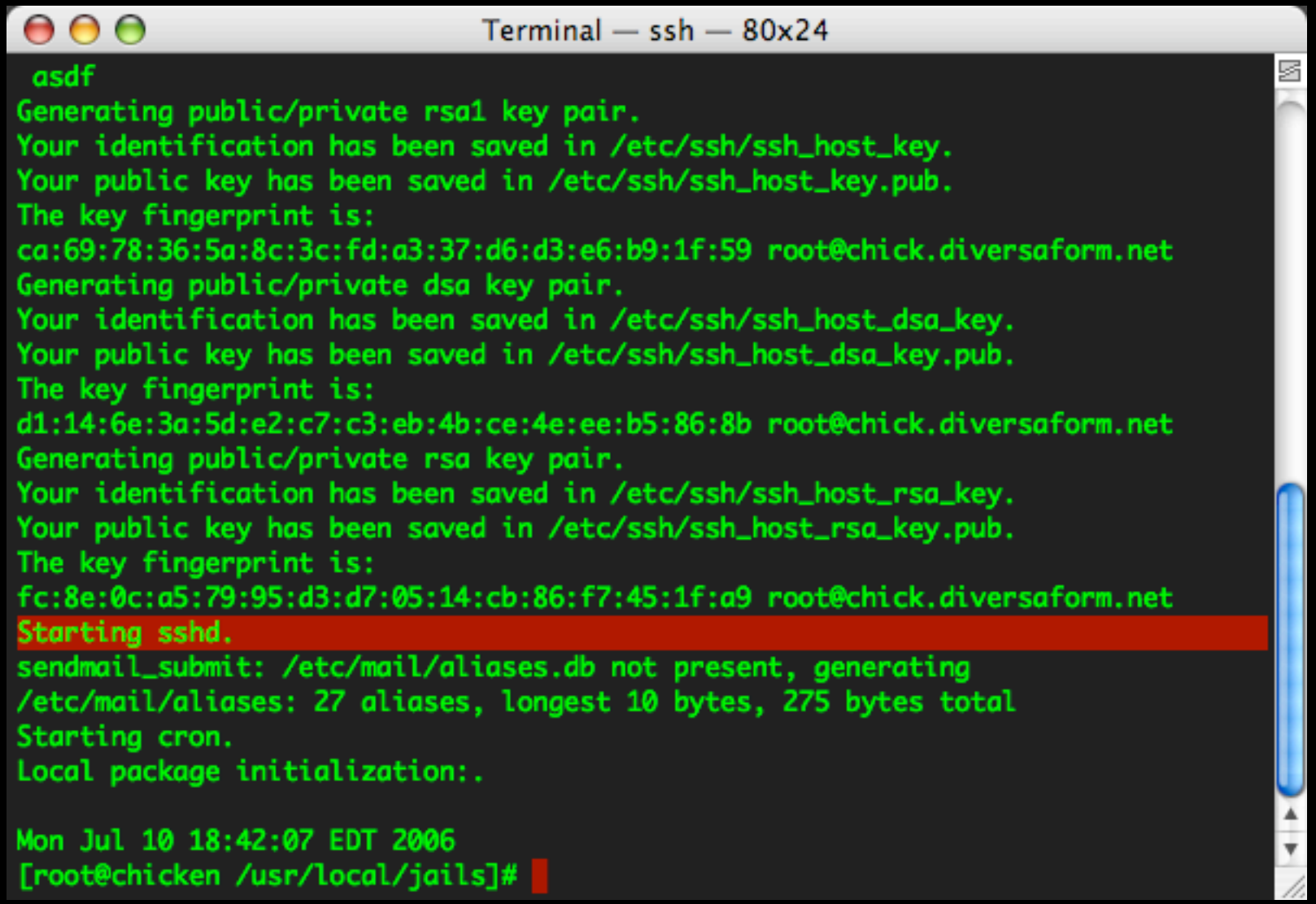
start!

type some random junk to seed entropy,



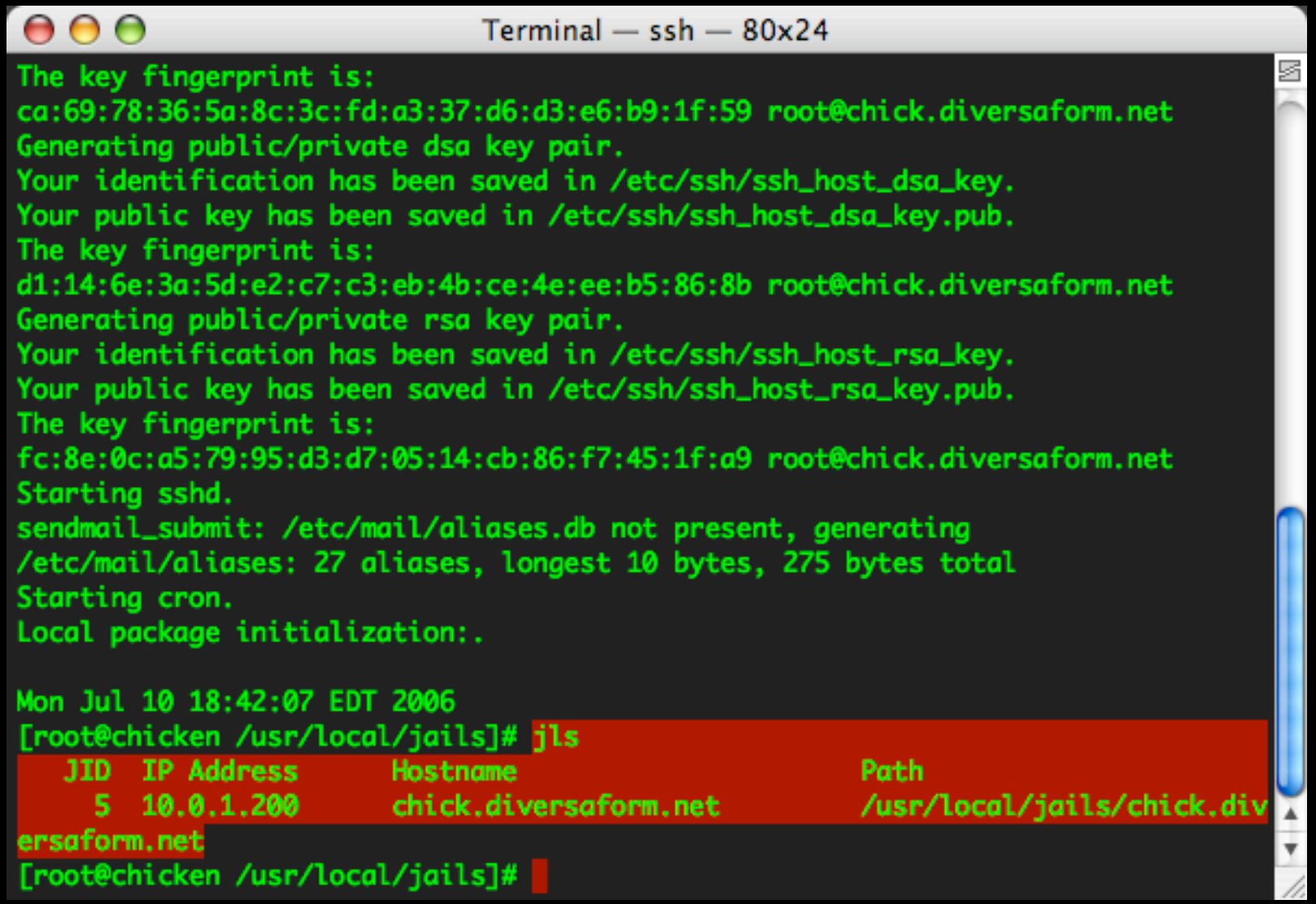
start!

jail finished starting



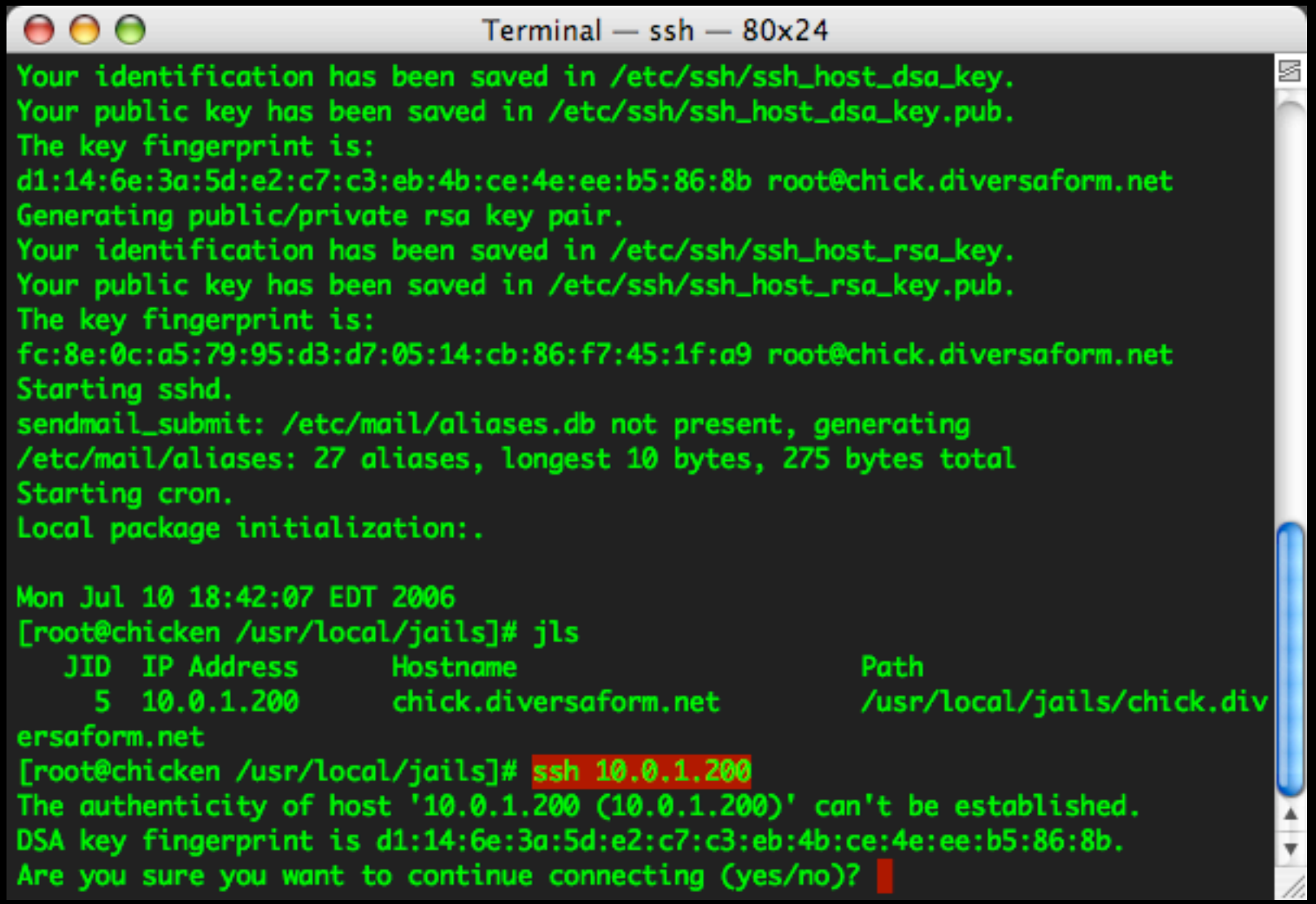
running

jls(8) lists running jails, gives a jail ID



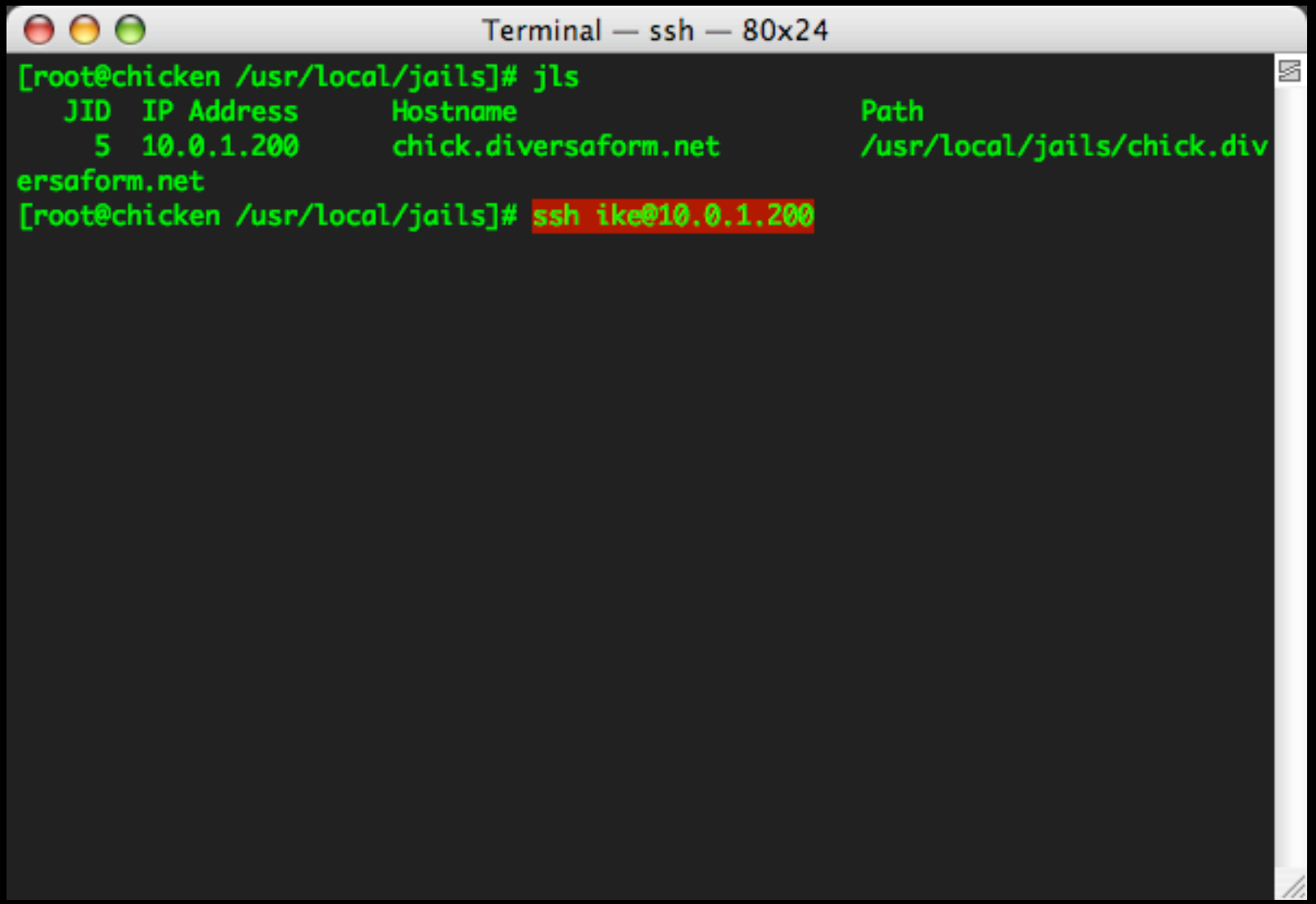
using the jail

ssh into the jail, treat it like a server.



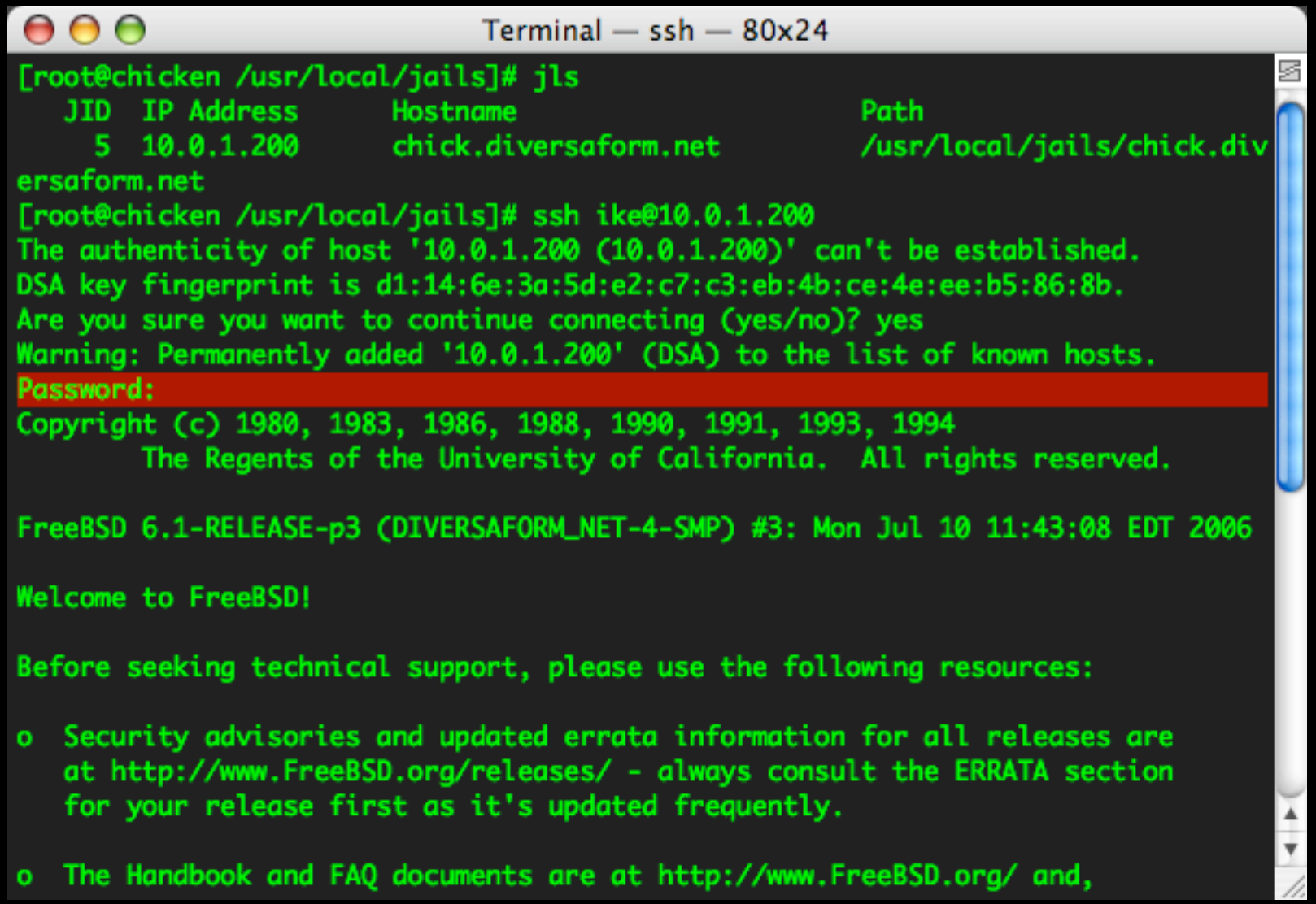
using the jail

ssh into the jail, treat it like a server.



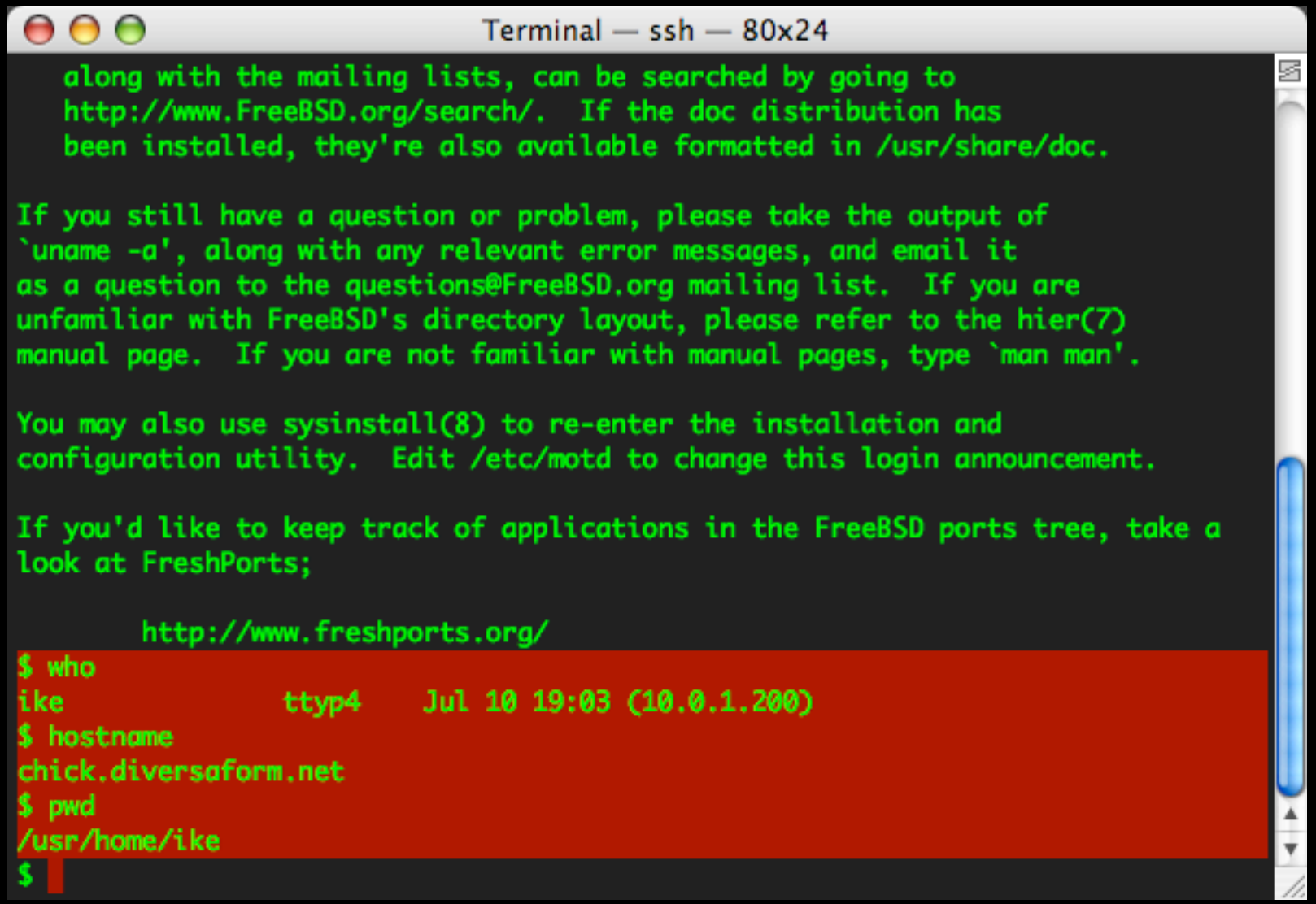
using the jail

ssh into the jail, treat it like a server.



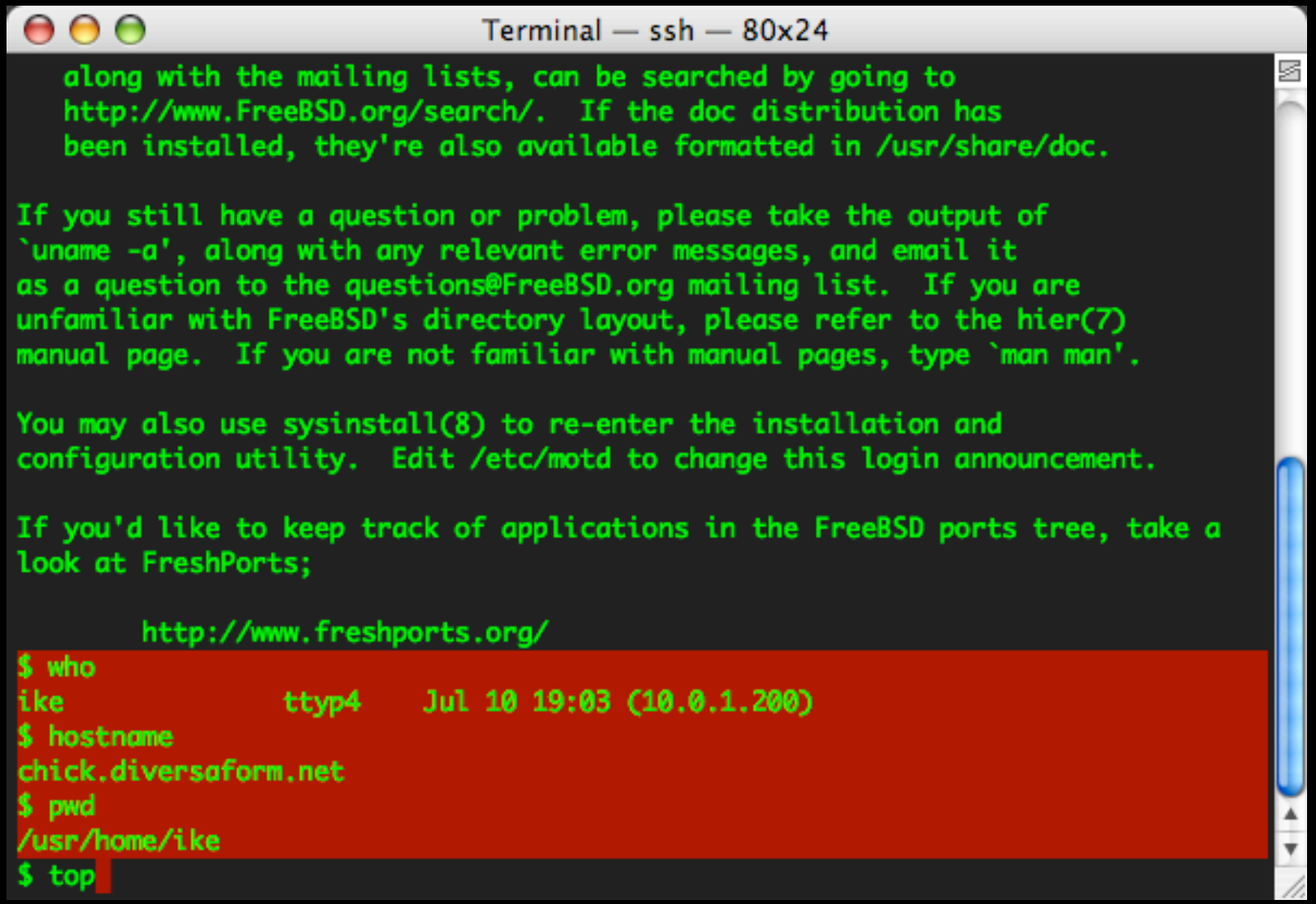
inside the jail

just like any new server



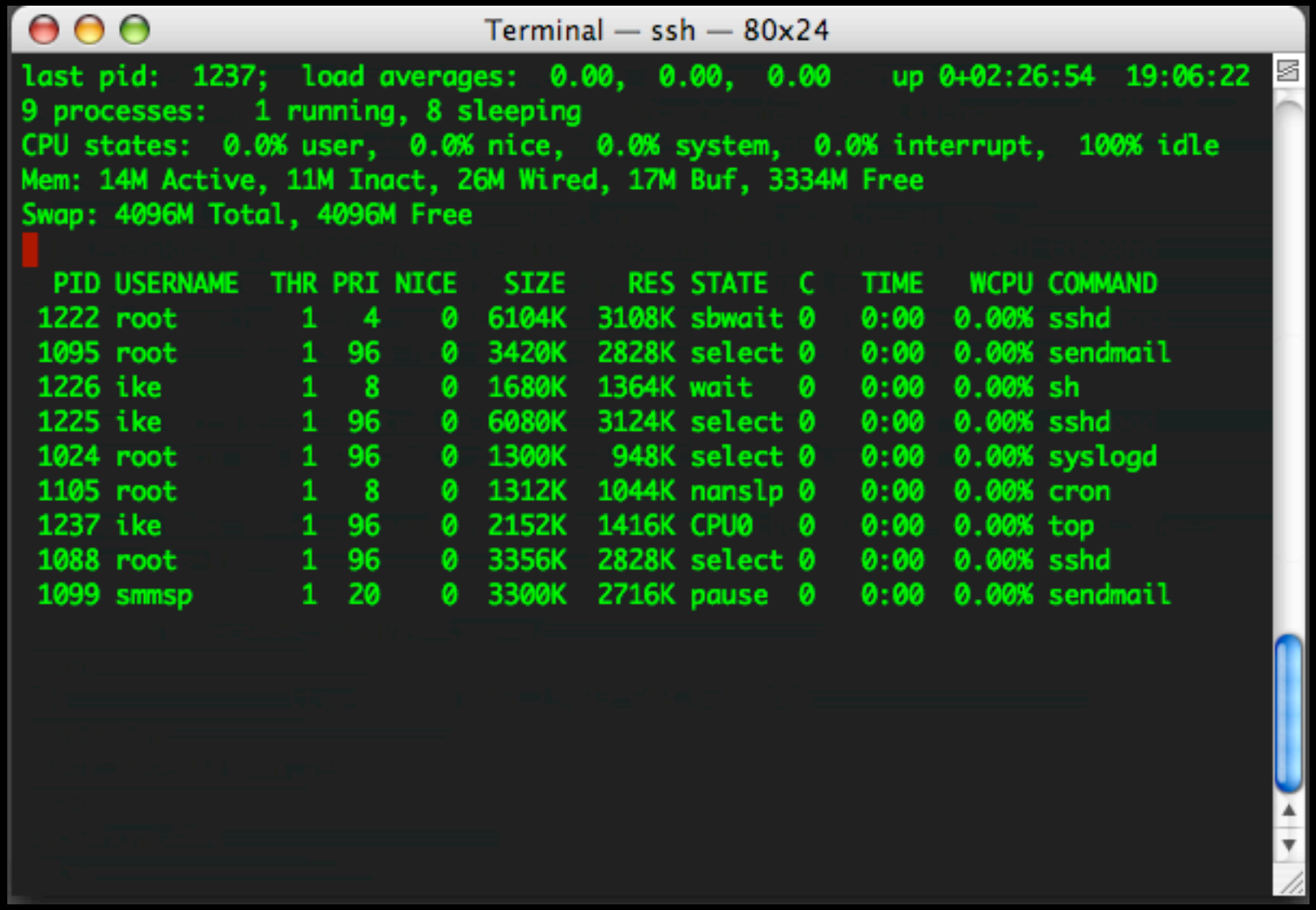
inside the jail

just like any new server



inside the jail

just like any new server

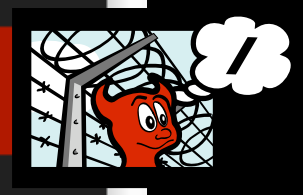


inside the jail

you have root!

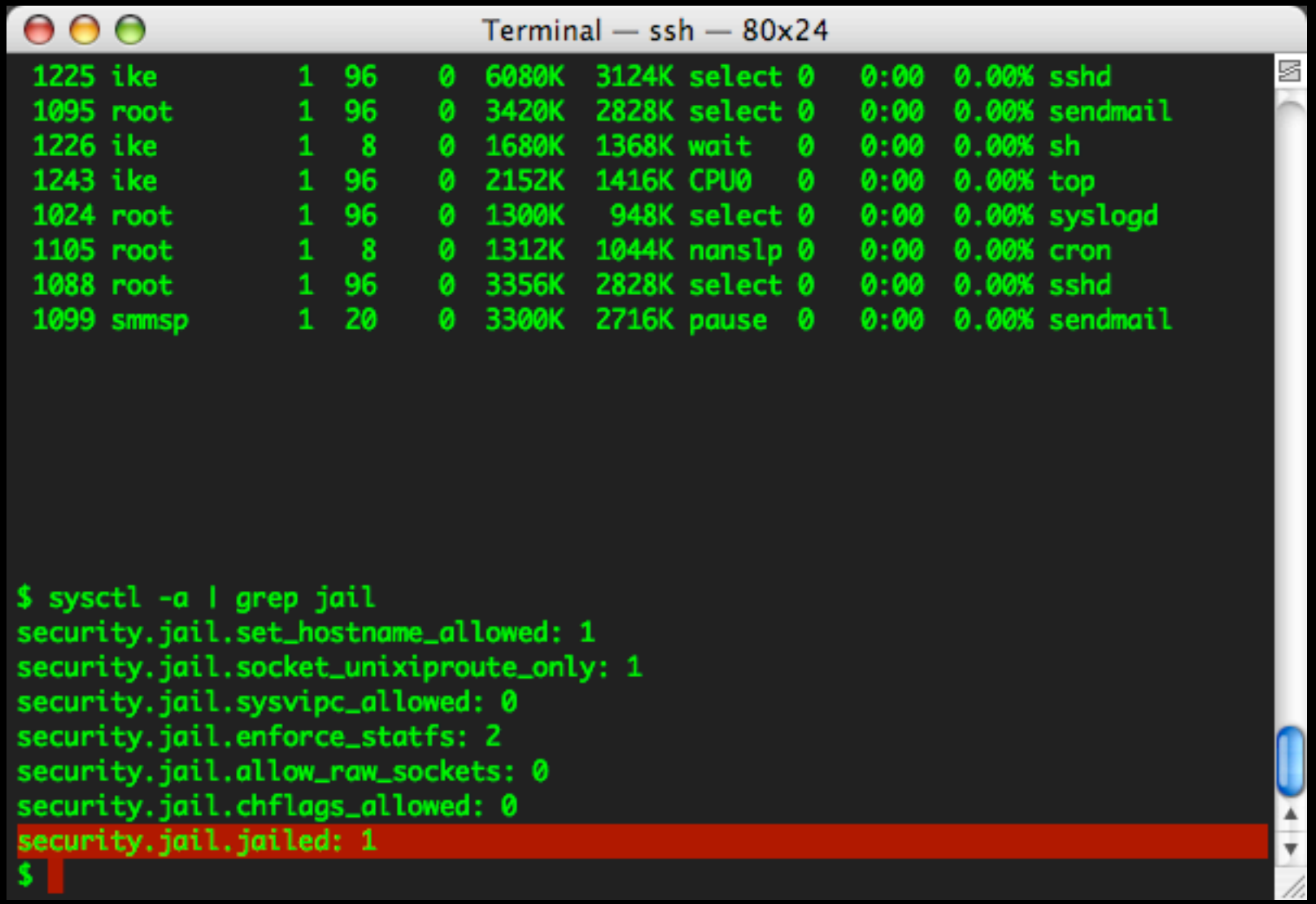
Terminal — ssh — 80x24

```
$ hostname
chick.diversaform.net
$ ifconfig
bge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=1b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING>
        inet 10.0.1.200 netmask 0xffffffff broadcast 10.0.1.200
        ether 00:e0:81:34:bf:8c
        media: Ethernet autoselect (1000baseTX <full-duplex>)
        status: active
bge1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=1b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING>
        ether 00:e0:81:34:bf:8d
        media: Ethernet autoselect (1000baseTX <full-duplex>)
        status: active
plip0: flags=108810<POINTOPOINT,SIMPLEX,MULTICAST,NEEDSGIANT> mtu 1500
pflog0: flags=0<> mtu 33208
pfsync0: flags=0<> mtu 2020
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
$ su
Password:
chick# whoami
root
chick#
```



inside the jail

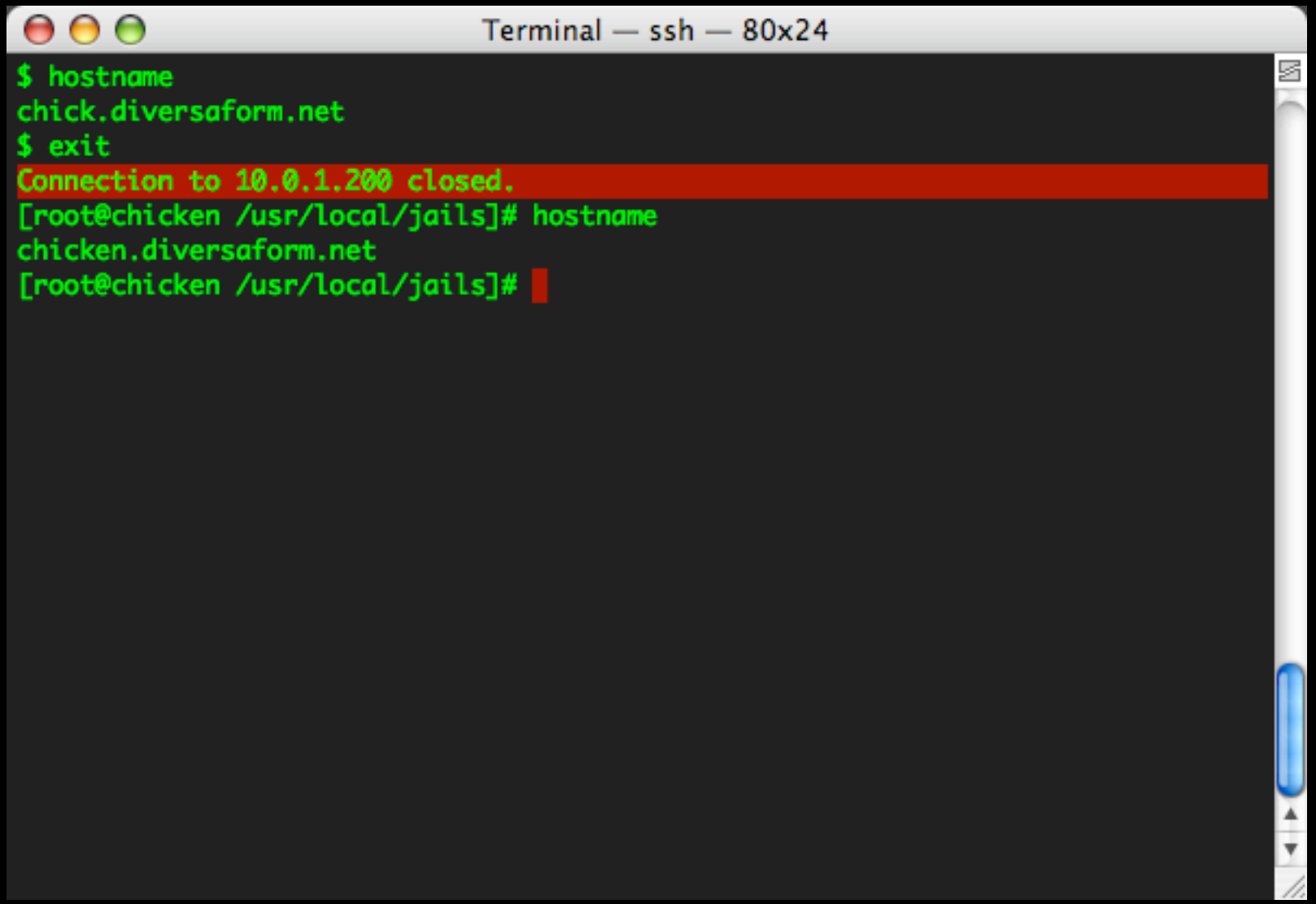
how do you know you are inside a jail?



http://www.freebsd.org/cgi/query-pr.cgi?pr=95977 - will explain this url later...

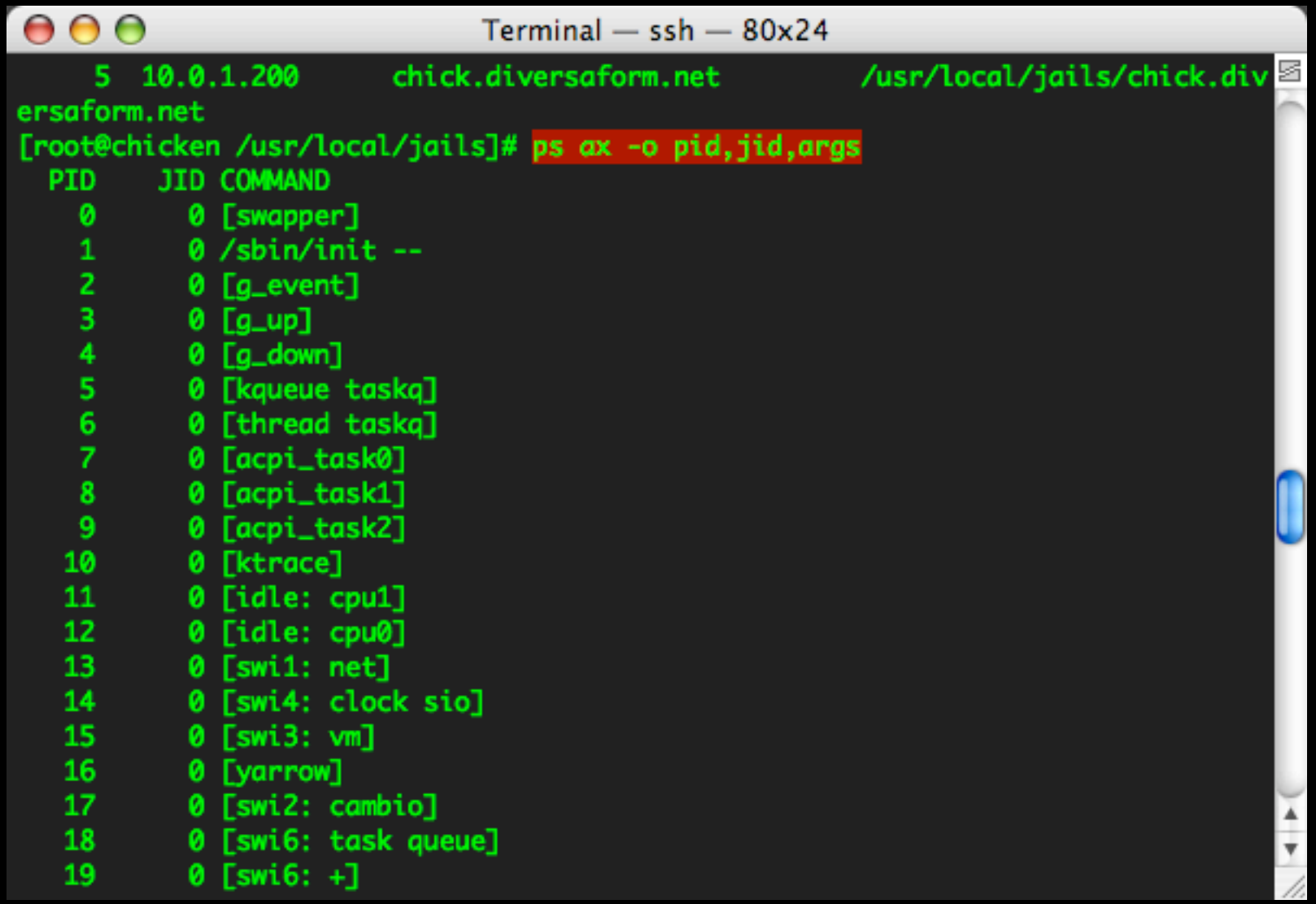
stop and start jail

exit the jail, (ssh)



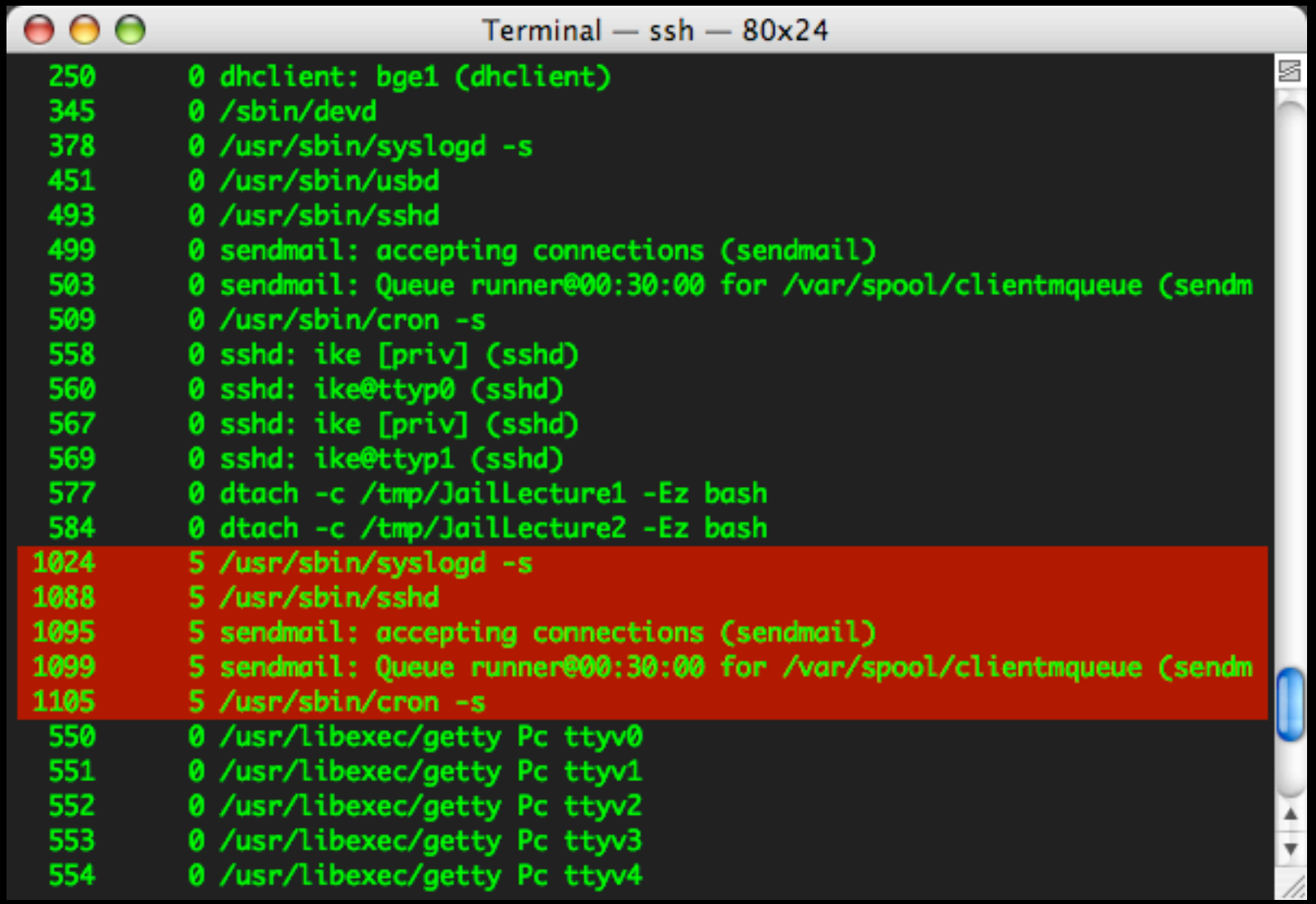
stop and start jail

look at jailed processes (man page goodies)



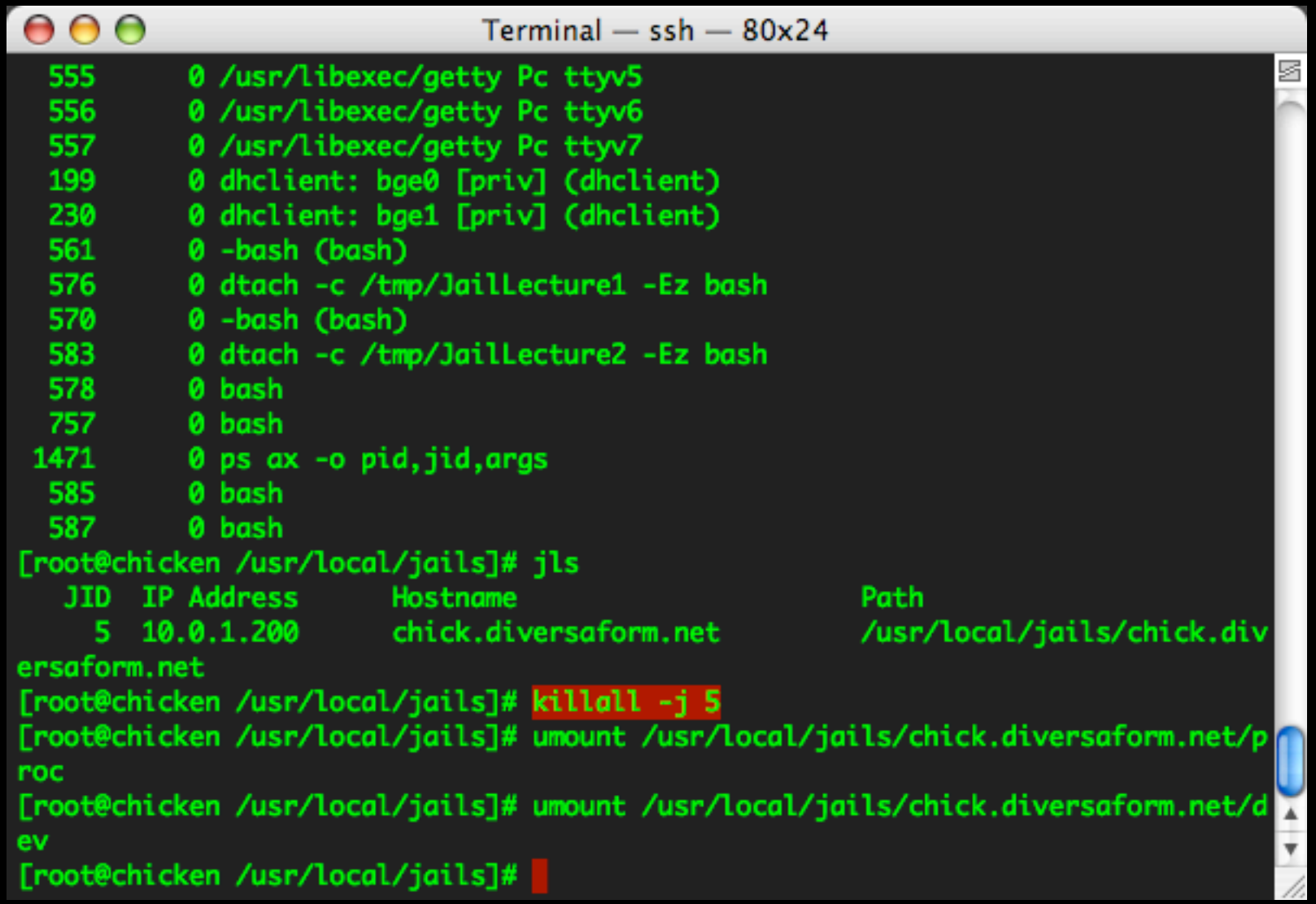
stop and start jail

look at jailed processes (man page goodies)



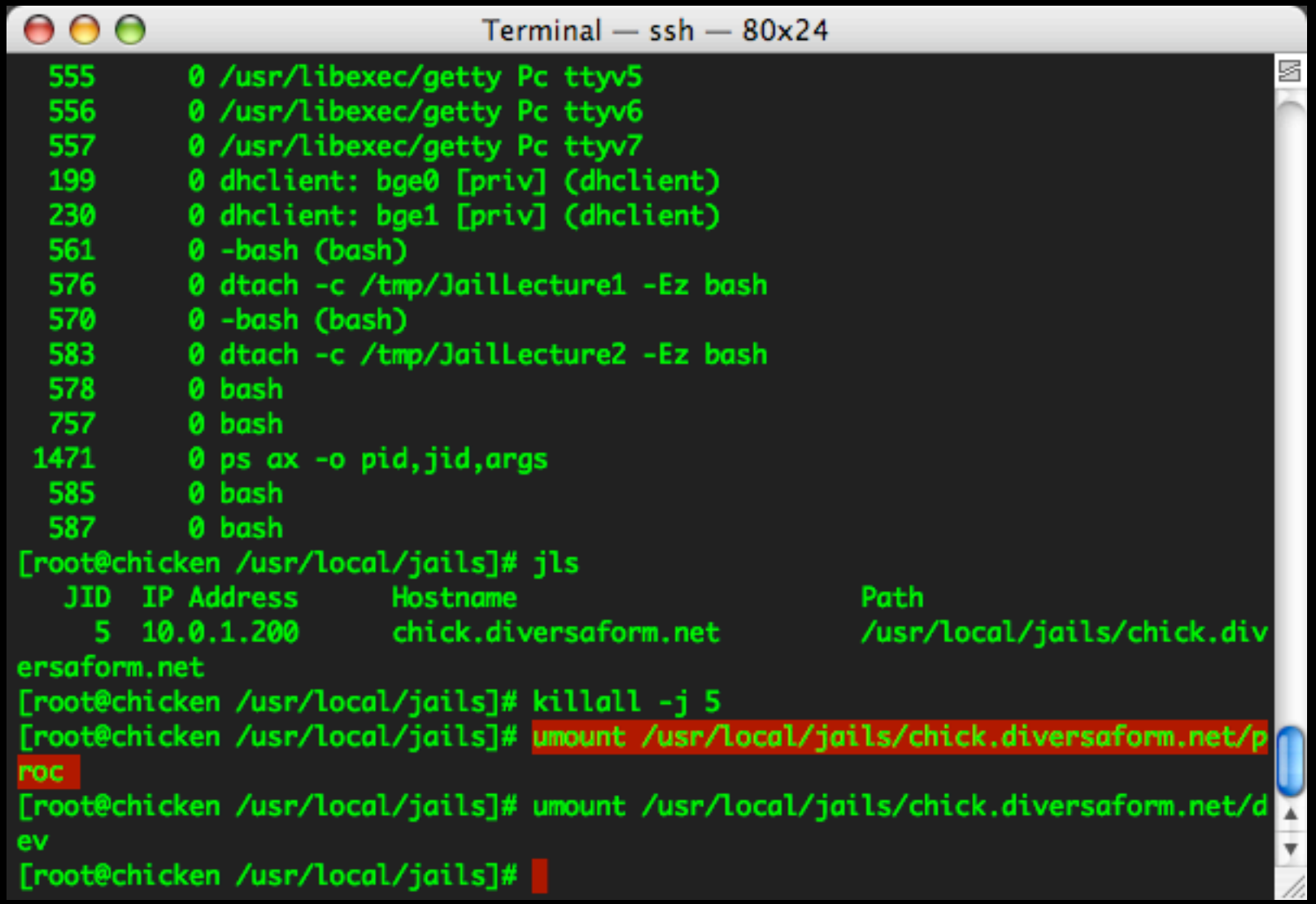
stop and start jail

use killall with -j flag



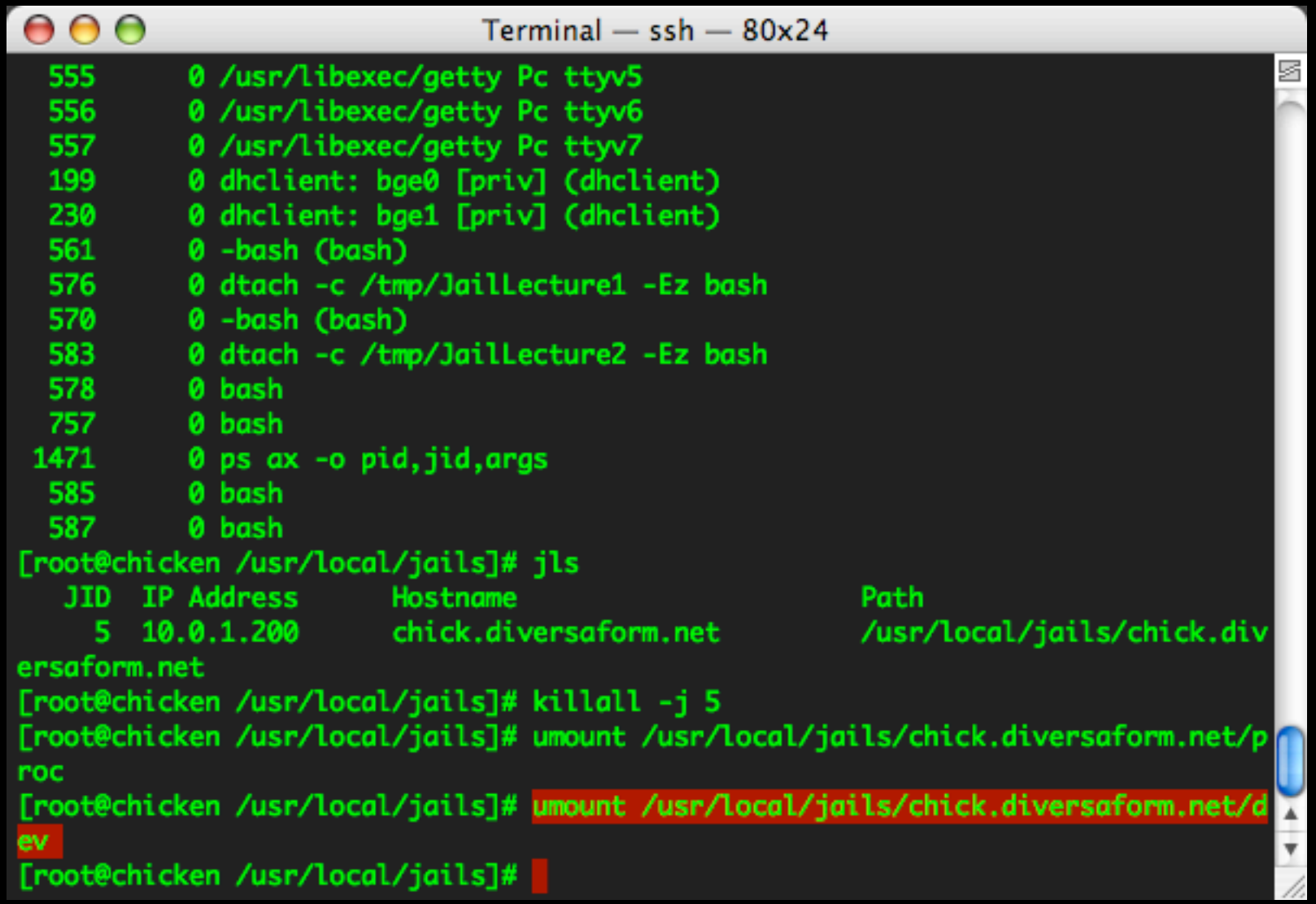
stop and start jail

watch out for stacking mount points!



stop and start jail

watch out for stacking mount points!

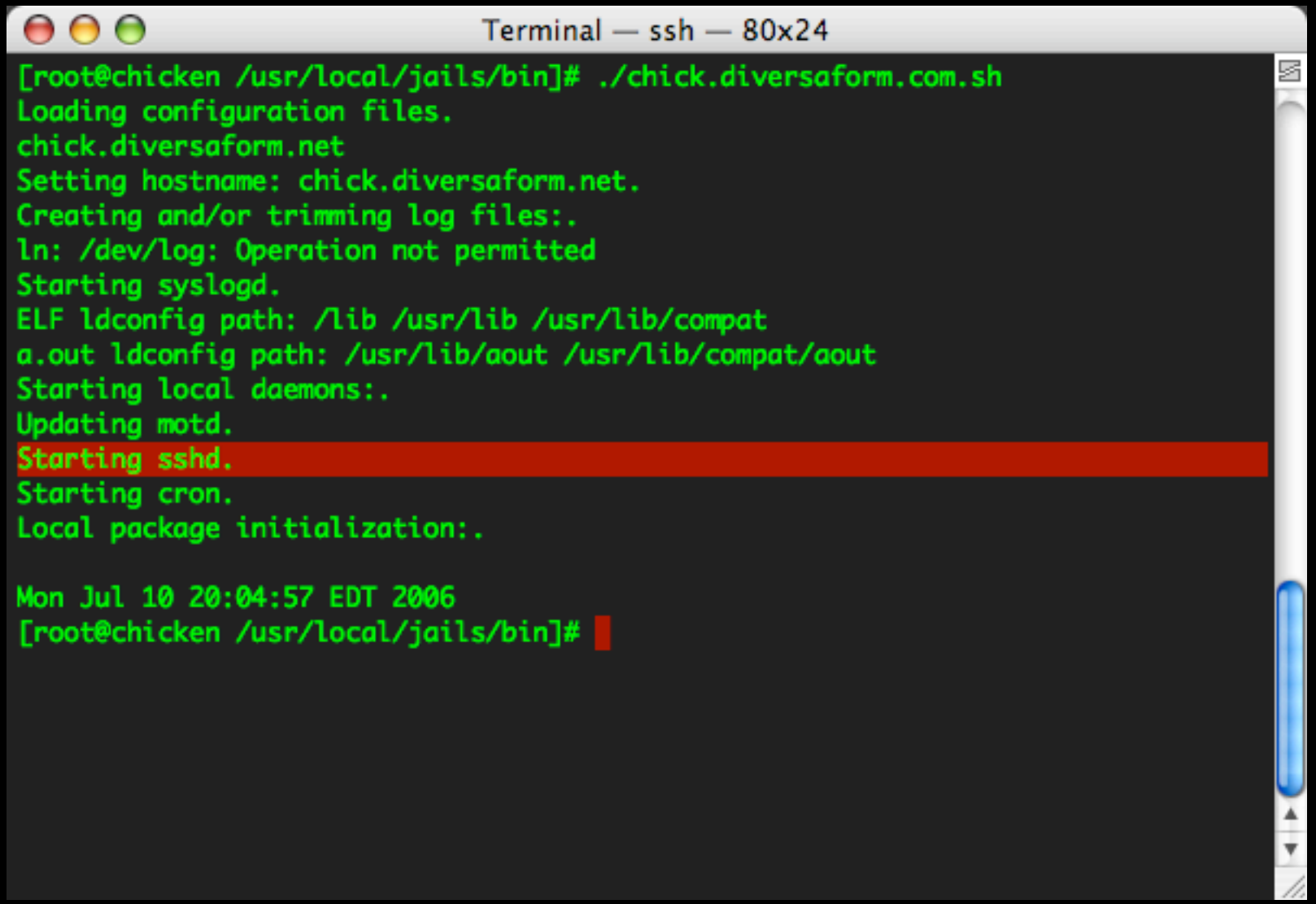


stop and start jail

restarting with the script this time,

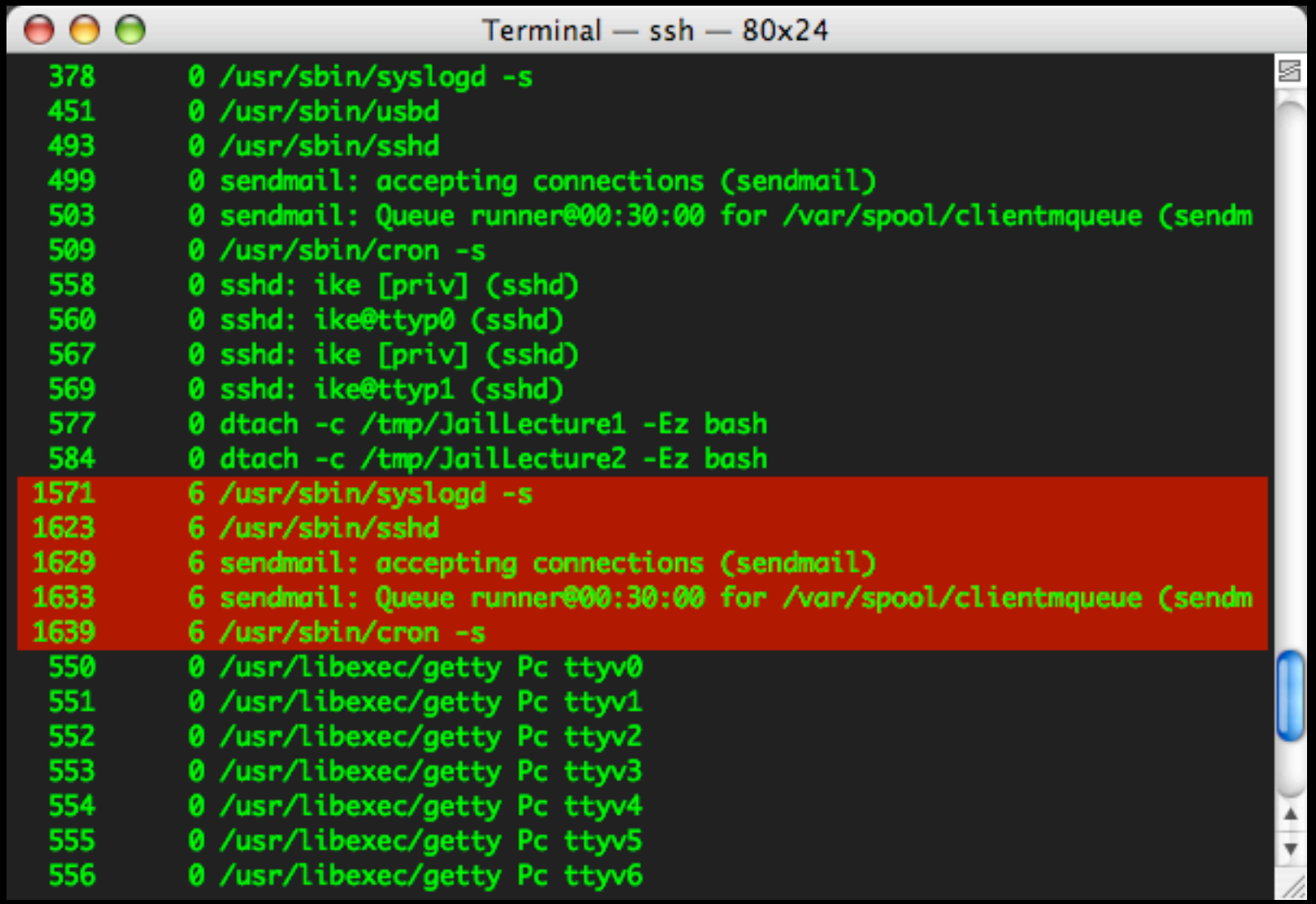
stop and start jail

restarting with the script this time,



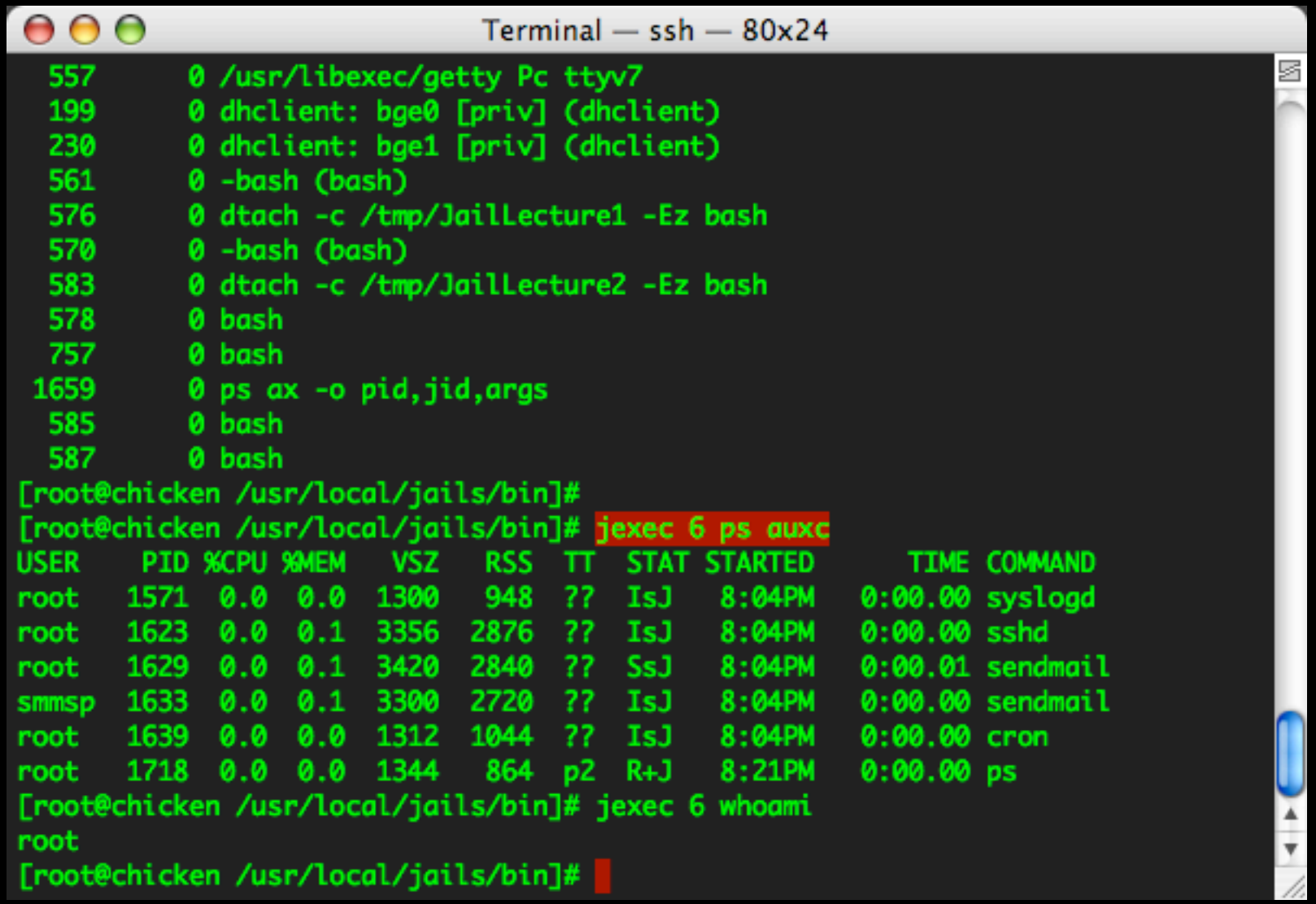
stop and start jail

now the jid has incrimented once, to 6

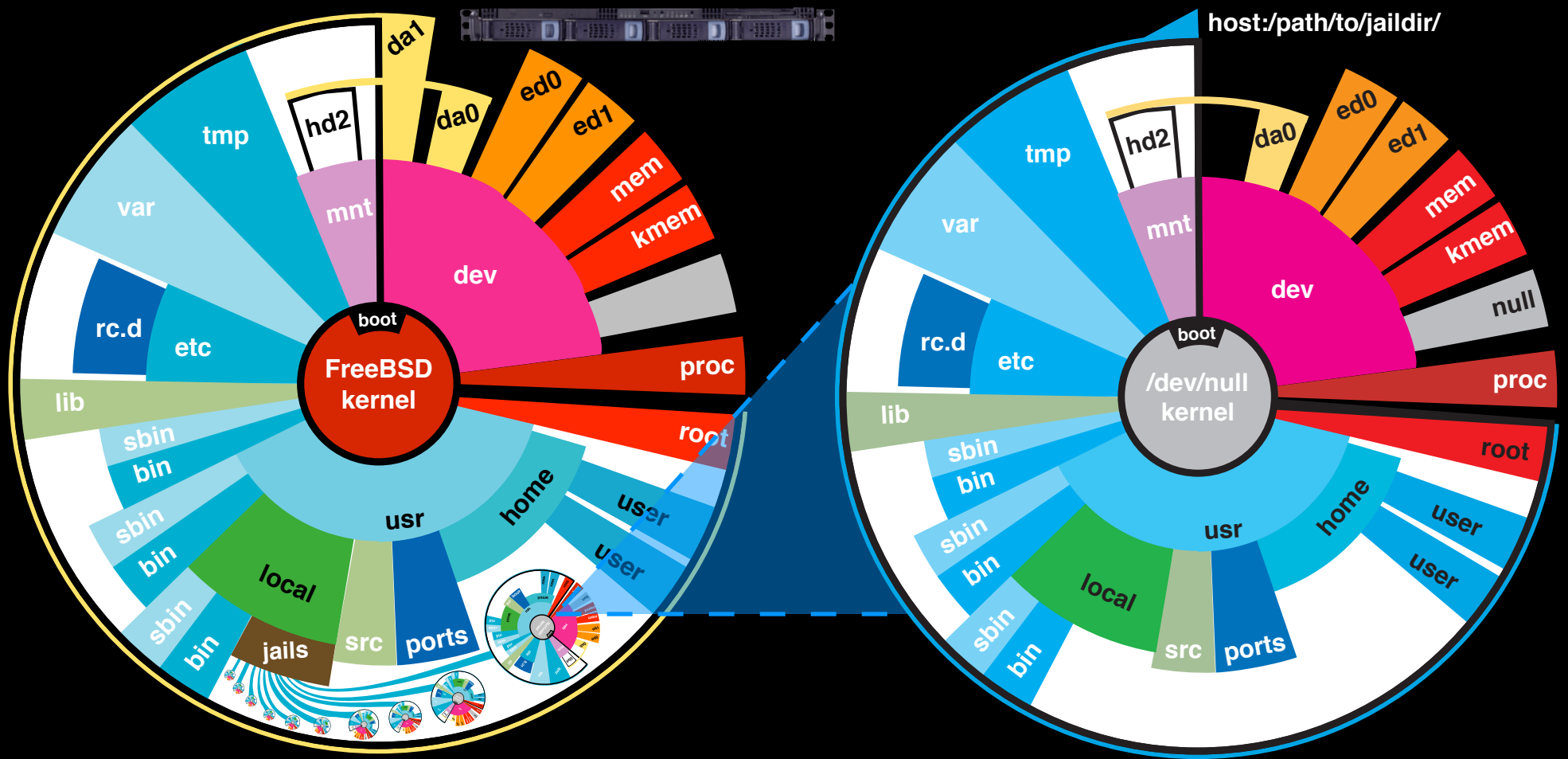


running processes

jexec to check processes (bad idea, in practice)



Practical Comparison



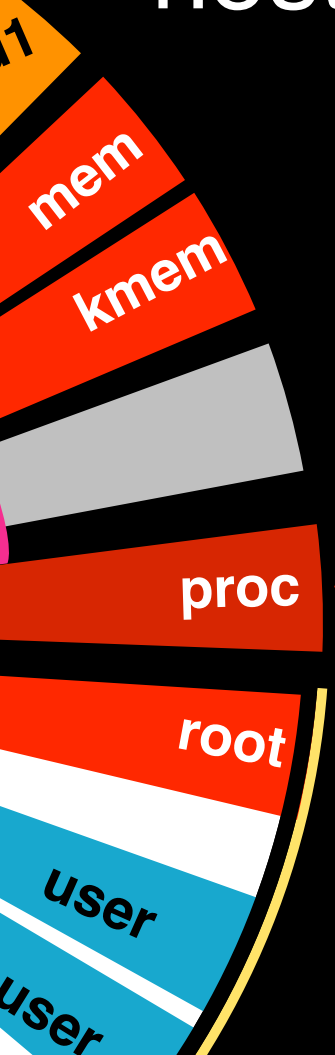
host

jail

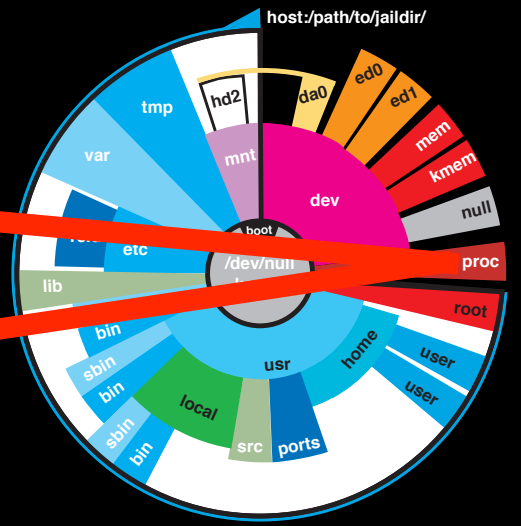
Process Tree:

JailingServer
_init
_daemon/process etc...
_daemon/process etc...
_daemon/process etc...
_daemon/process etc...
_jail (Jail 1)
_daemon/process etc...
_daemon/process etc...
_daemon/process etc...
_jail (Jail 2)
_daemon/process etc...
_daemon/process etc...
_daemon/process etc...
_jail (Jail 3)
_daemon/process etc...
_daemon/process etc...
_daemon/process etc...
_jail (Jail 4)
_daemon/process etc...
_daemon/process etc...
_daemon/process etc...

host



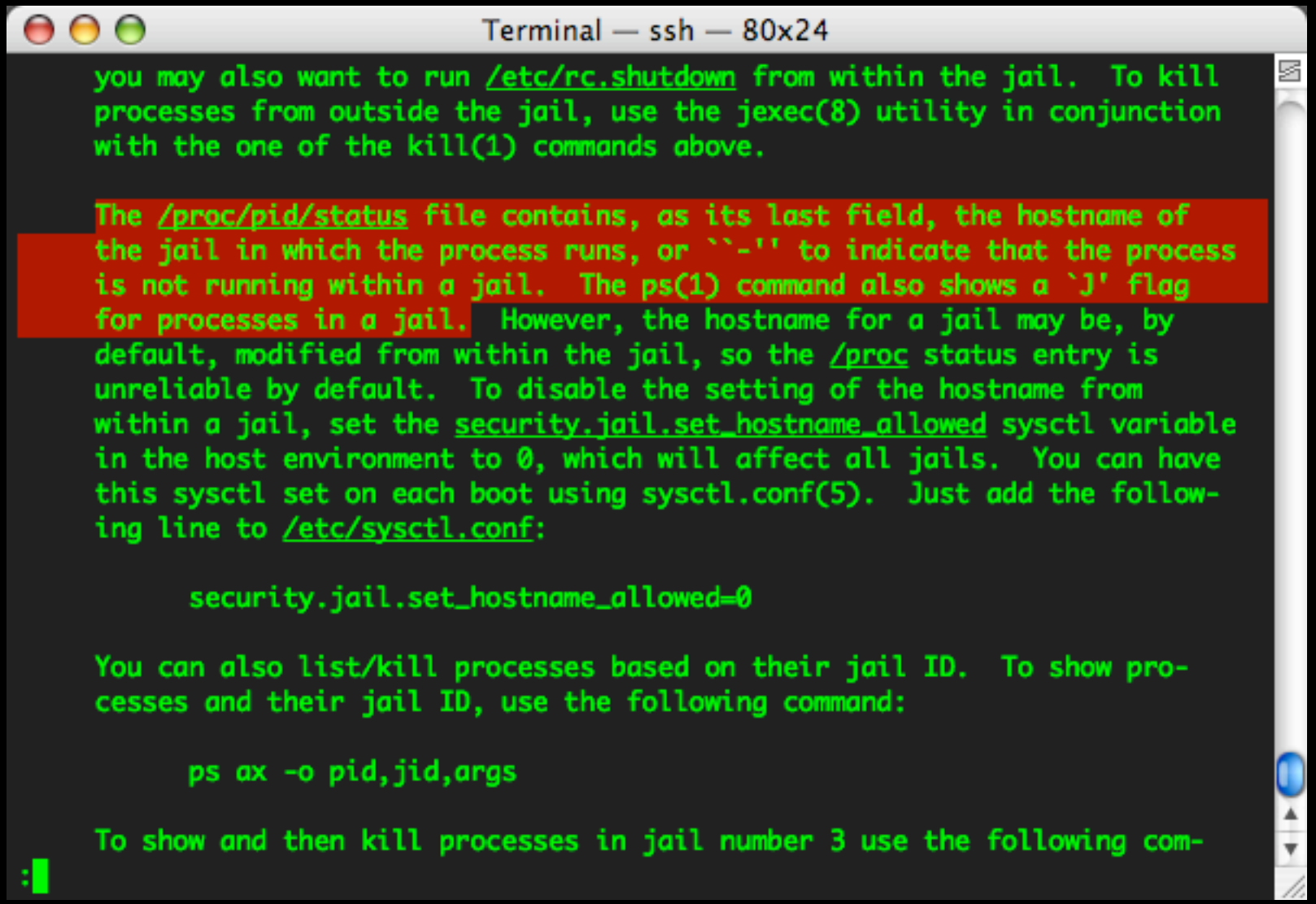
jail

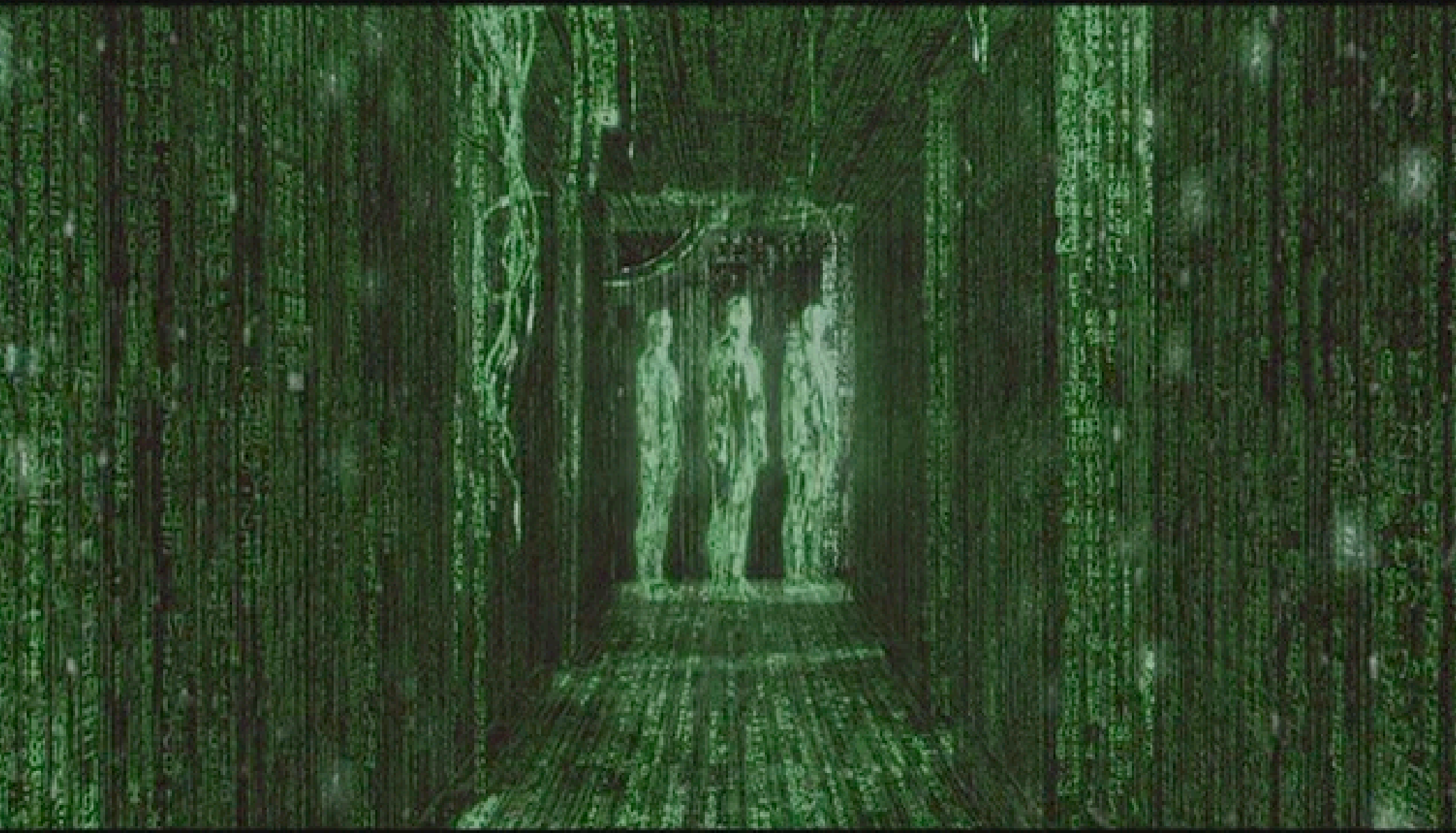



Terminal — ssh — 80x24
chickenhawk:/home/ike ike$ sudo cat /proc/35721/status
sshd 35721 1 35721 35721 - sldr 1138578391,270482 4,956102 27,766581 select 0 0
0,0 chick.diversaform.net
chickenhawk:/home/ike ike$ sudo cat /proc/93030/status
sshd 93030 93025 93025 93025 noflags 1152553319,908960 0,61426 0,40950 select
1001 1001 1001,1001,1001,0 -
chickenhawk:/home/ike ike$
mem
kmem
proc
root
user
user
host:/path/to/jaildir/
hd2
da0
ed0
ed1
mem
kmem
null
proc
root
user
user
home
usr
local
src
ports
sbin
bin
sbin
bin
sbin
bin
lib
rc.d
etc
var
tmp
mnt
dev
boot
/dev/null
kernel
host
jail

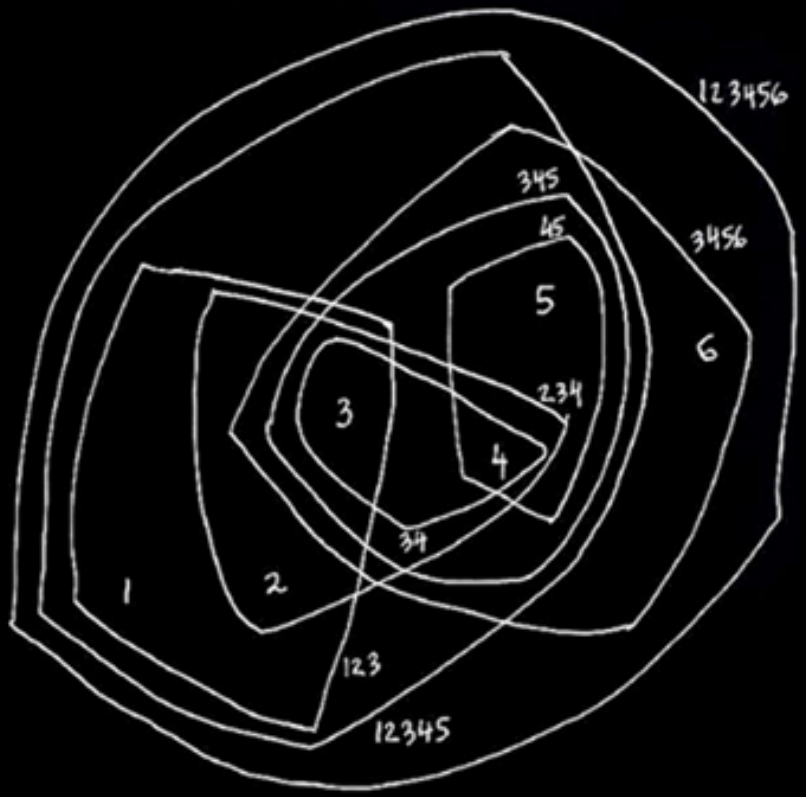

Terminal — ssh — 80x24
chickenhawk:/home/ike ike$ sudo cat /proc/35721/status
sshd 35721 1 35721 35721 - sldr 1138578391,270482 4,956102 27,766581 select 0 0
0,0 chick.diversaform.net
chickenhawk:/home/ike ike$ sudo cat /proc/93030/status
sshd 93030 93025 93025 93025 - noflags 1152553319,908960 0,61426 0,40950 select
1001 1001 1001,1001,1001,0 -
chickenhawk:/home/ike ike$
mem
kmem
proc
root
user
user
host
host:/path/to/jaildir/
boot
/dev/null
kernel
tmp
var
rc.d
etc
lib
sbin
bin
sbin
bin
sbin
bin
hd2
mnt
da0
ed0
ed1
mem
kmem
dev
null
proc
root
usr
home
user
user
local
src
ports
jail

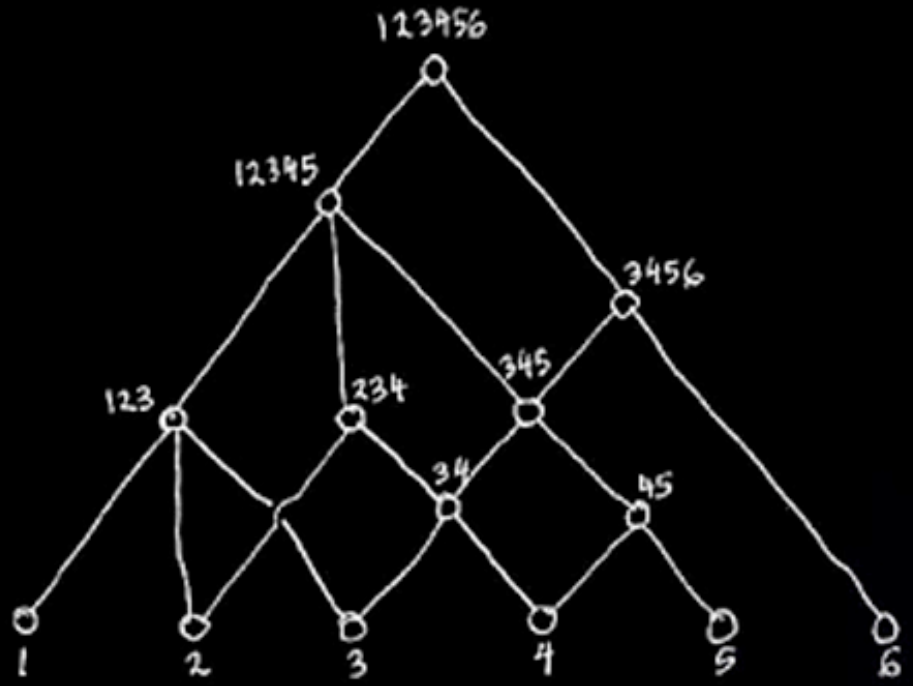

Terminal — ssh — 80x24
chickenhawk:/home/ike ike$ sudo cat /proc/35721/status
sshd 35721 1 35721 35721 - sldr 1138578391,270482 4,956102 27,766581 select 0 0
0,0 chick.diversaform.net
chickenhawk:/home/ike ike$ sudo cat /proc/93030/status
sshd 93030 93025 93025 93025 - noflags 1152553319,908960 0,73969 0,52749 select
1001 1001 1001,1001,1001,0 -
chickenhawk:/home/ike ike$ ps auxc | grep VSZ; ps auxc | grep 35721; ps auxc | g
rep 93030
USER PID %CPU %MEM VSZ RSS TT STAT STARTED TIME COMMAND
root 35721 0.0 0.1 3352 1152 ?? IsJ 29Jan06 0:32.72 sshd
ike 93030 0.0 0.1 6092 2492 ?? S 1:41PM 0:00.13 sshd
chickenhawk:/home/ike ike$
mem
kmem
proc
root
user
host
host:/path/to/jaildir/
hd2
da0
ed0
ed1
mem
kmem
null
proc
root
home
user
user
usr
local
src
ports
dev
mnt
tmp
var
rc.d
etc
lib
sbin
bin
boot
/dev/null
kernel
jail


Terminal — ssh — 80x24
you may also want to run /etc/rc.shutdown from within the jail. To kill processes from outside the jail, use the jexec(8) utility in conjunction with the one of the kill(1) commands above.
The /proc/pid/status file contains, as its last field, the hostname of the jail in which the process runs, or ``-'' to indicate that the process is not running within a jail. The ps(1) command also shows a `J' flag for processes in a jail. However, the hostname for a jail may be, by default, modified from within the jail, so the /proc status entry is unreliable by default. To disable the setting of the hostname from within a jail, set the security.jail.set_hostname_allowed sysctl variable in the host environment to 0, which will affect all jails. You can have this sysctl set on each boot using sysctl.conf(5). Just add the following line to /etc/sysctl.conf:
security.jail.set_hostname_allowed=0
You can also list/kill processes based on their jail ID. To show processes and their jail ID, use the following command:
ps ax -o pid,jid,args
To show and then kill processes in jail number 3 use the following com-
:



jail(8) best practices





diagrams from "A City is Not A Tree" , essay by urban designer Christopher Alexander

and opportunities...

break out of jail?

Poul-Henning Kamp (PHK) wrote the jail feature for R&D Associates http://www.rndassociates.com/ who contributed it to FreeBSD around 1998.

To my knowledge, nobody has broken out of a jail directly, ever. It is however assumed that nobody has tried that hard yet, as it is still considered ‘esoteric’.

If someone breaks jail, PHK wrote that he would love to know about it.

best practices

- ssh into jails to manage their processes!!!!
- You always can see the jailed filesystem/userland from host server, be careful.
- Design your jailing system carefully, be creative with core UNIX utilities.
- Use your highest secure practices for host server...

great utilities

- 4.x, jps, jkill, jtop
- 5.x, 6.x, onward builtin ps, kill
  - !plus jls(8), jexec(8) jattach (2), sysctl features for jailing
- Design your jailing system carefully, be creative (note about nullfs, devfs)
- additionally, handy: pstree, xtail, disk images via mdconfig

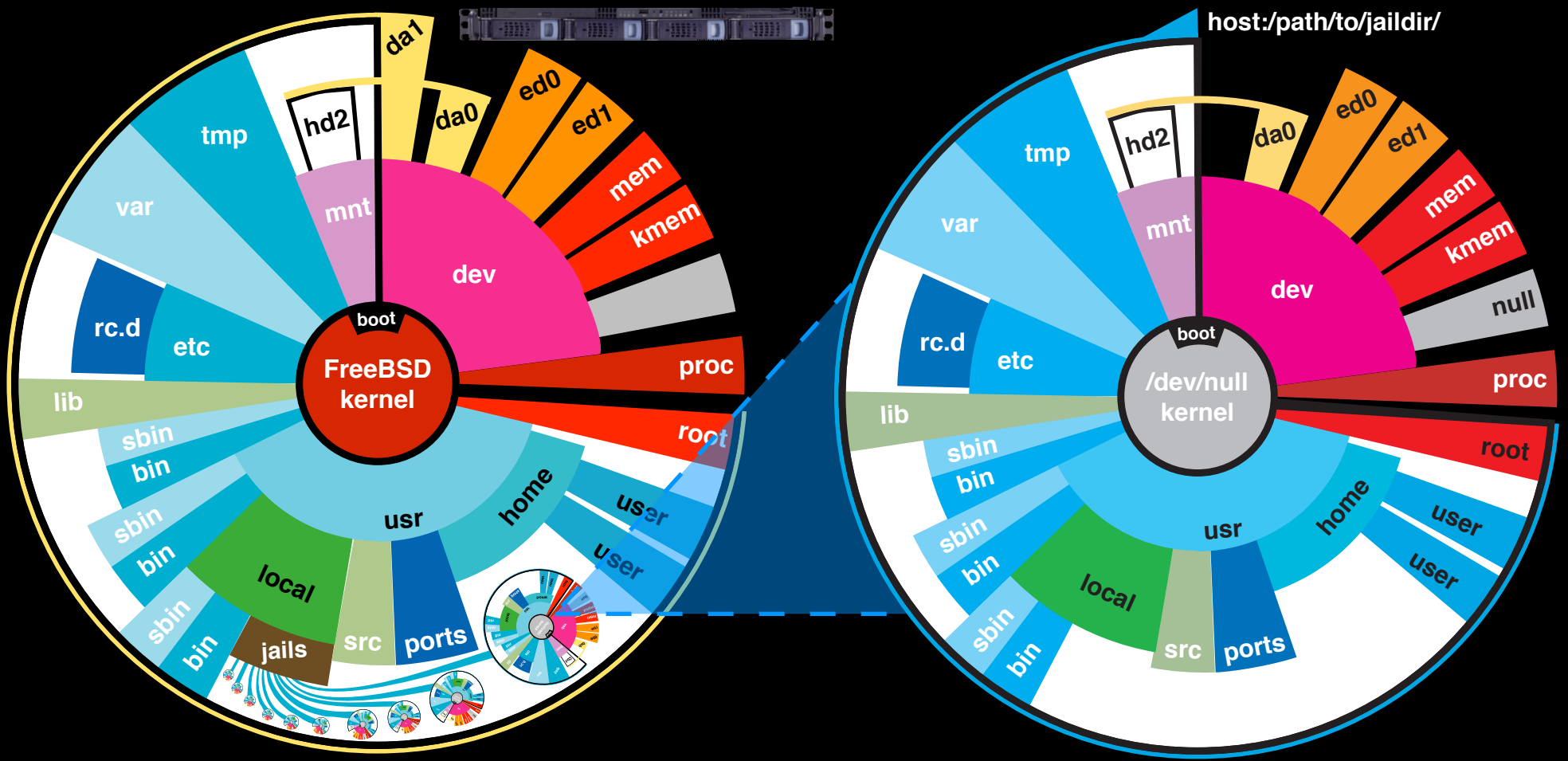
common weak points

- lost jail?
  - [hostname lockdown]
- resource attacks
  - disks full
    - [partitions, disk images]
  - fork bombs, memory hogs
    - [securelevels, login.conf]
  - process control
- direct driver access
  - [flags to mount devfs, procfs]

common weak points

- lost jail?
  - [hostname lockdown]
- resource attacks
  - disks full
    - [partitions, disk images]
  - fork bombs, memory hogs
    - [securelevels, login.conf]
  - process control
- direct driver access
  - [flags to mount devfs, procfs]

Comments on Isolation



memory/process attacks

http://www.samag.com/documents/s=1151/sam0105d/0105d.htm
http://www.freebsd.org/doc/en_US.ISO8859-1/books/arch-handbook/jail.html

OpenRoot Project, fork-bombs, FreeBSD SecureLevels/maxproc, reality, and process control

memory/process attacks

(check the Defcon 14 CD)

```
# hog.c, a small utility to hog system memory
# written by Brian Redman (BER) sometime around 1986
# Basic Instructions, Compile this code to a binary:

cc hog.c -o hog

# then run something like:

hog 10

# and the hog will do just that- sit and hog 10mb of ram.
# To run a hog stampede, (a fork bomb):

while (1)
hog 99m&
end
```

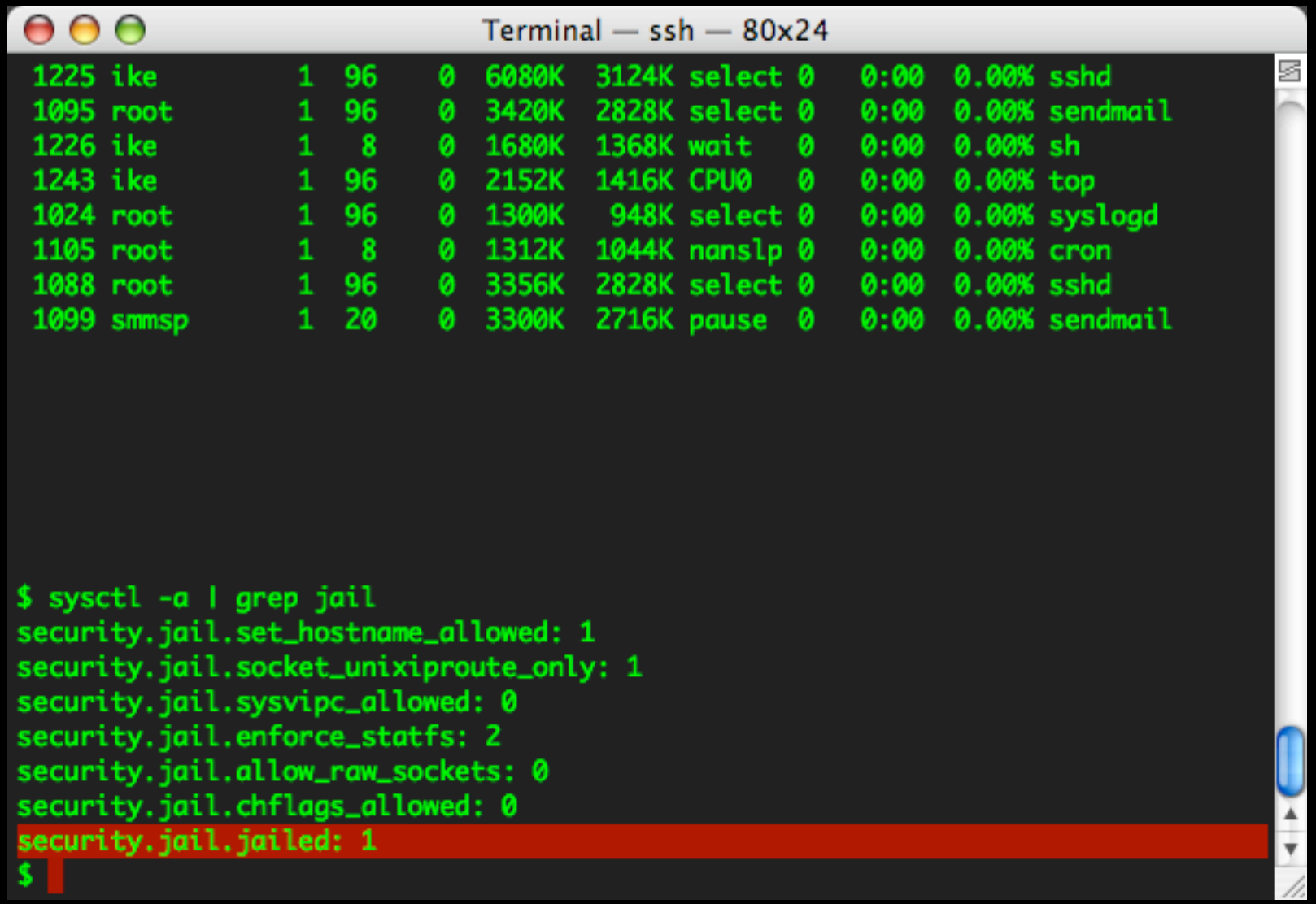
memory/process attacks

(check the Defcon 14 CD)

```
# STEP 1)
# jailed /etc/login.conf file, example of restricted values:
:maxproc=30:\
:memoryuse=25M:\

# STEP 2)
# Set immutable flags on jailed /etc/login.conf, example:
chflags schg $D/etc/login.conf

# STEP 3)
# Set a higher securelevel on a per-jail basis
# (5.x onward, 4.x jailing only securlevels for entire host)
# add the following line to the jailed /etc/sysctl.conf:

kern.securelevel=2
# securelevel 1 is minimum, read the man page for securelevel
```

honeypot?

compile and give the jail a kernel, fix sysctl:



http://www.freebsd.org/cgi/query-pr.cgi?pr=95977

disk resource control

- Put at least your jailed systems on a seperate partition, or perhaps each jail (rigid in practice)
- File-Backed Disk Images (mdconfig, in handbook)- insanely flexible, but take extra memory (usually negligible)

file-backed disks (.dmg)

- WOW, they’re convienent.
- watch out for device numbering (or things get lost), here’s where Jailing strategies from 4.x come in handy... unless someone has a better way of manging device nodes
- speed is getting excellent for file-backed memorry disks, but will always introduce some overhead in file I/O

file-backed disks (.dmg)

FreeBSD handbook has tons more information!

```
# writing 1gb blank file, (analagous to creating an
unformatted harddrive)
dd if=/dev/zero of=1gb.img bs=1k count=1024k

# attaching the file (analagous to attaching a harddrive)...
mdconfig -a -t vnode -f 1gb.img -u 1101

# formating the disk...
disklabel -r -w md1101 auto

# detaching the disk (analagous to ejecting a harddrive)...
mdconfig -d -u 1101
```

file-backed disks (.dmg)

mount disks when starting jails,

```
<snip - jail start script>


mdconfig -a -t vnode -f /path/to/jaildisk_file.dmg -u 200
mount /dev/md200c /path/to/jail_userland_mount_dir

# regarding '-u 200' above, it can be handy to use some
# variant of a jail's respective IP address for it's disk
# image devide node id, so it's easy to track down on host
# system with many jailed servers.

# later in script,
jail /path/to/jail_userland_mount_dir \
hostname.fqdn.com \
10.0.1.200 \
/bin/sh /etc/rc

</snip>
```

automation

- Tarball packaging is your friend.
  clean, simple, reliable.
    be aware of dev/proc mounts
    be aware of symlinks
- use FreeBSD Ports Mechanism!
  (**not** for the ports collection, that's insanely presumptuous, [borderline irresponsible]
- CVS/SVN anyone?

upgrading jailed systems

- Simply use buildworld, (FROM HOST SYSTEM),
- toss buildworld DESTDIR flag, with a jail's userland path
- follow the handbook: http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/makeworld.html

/etc/sysctl.conf (host)

(check the Defcon 14 CD)

```
# $FreeBSD: src/etc/sysctl.conf,v 1.8 2003/03/13 18:43:50 mux Exp $
#
#  This file is read when going to multi-user and its contents piped thru
#  ``sysctl'' to adjust kernel values.  ``man 5 sysctl.conf'' for details.
#

# Uncomment this to prevent users from seeing information about processes that
# are being run under another UID.
#security.bsd.see_other_uids=0

# ikenote jailing additives
security.jail.set_hostname_allowed=0     # default = 1 # jailed resetting hostname.
security.jail.enforce_statfs=2           # default = 2 # mount point info.
security.jail.allow_raw_sockets=0        # default = 0 # for ping, etc...
security.jail.socket_unixiproute_only=1  # default = 1 # access to routing sockets.
security.jail.sysvipc_allowed=0          # default = 0 # SysV shareed mem? Ha!
security.jail.chflags_allowed=0          # default = 0 # root less than root...
```

sysctl (stock values)

(check the Defcon 14 CD)

```
$ sysctl -a | grep jail
security.jail.set_hostname_allowed: 0
security.jail.socket_unixiproute_only: 1
security.jail.sysvipc_allowed: 0
security.jail.enforce_statfs: 2
security.jail.allow_raw_sockets: 0
security.jail.chflags_allowed: 0
security.jail.jailed: 0
```

firewalls (quick comment)

- context:
  - *why jail in the first place again?*
  - threats affect an entire host server
  - firewall at a higher level (mental shift to treat the host like a network gateway!)
    - global system firewalling, throttling
    - different boxes? different rules?

Start Script w/ Disk Image

(check the Defcon 14 CD)

```
#!/bin/sh

# simple, complete script to start a jail.

# define the absolute path to the jail,
J=/usr/local/jails/jailed.userland.directory

# define the ip address for the jail,
I=10.0.1.192

# define a hostname,
H=fqdn.com

ifconfig en0 inet alias $I/32

mount -t procfs proc $J/proc
mount_devfs devfs $J/dev
## add additonal flags to mount_devfs, to hide unnecessary devices!!!
## check the man page for mount_devfs

jail $J $H $I /bin/sh /etc/rc
```

jail crontab misc...

(check the Defcon 14 CD)

```
# comment out the following, just to keep syslog quiet for irrelevant items.

# Save some entropy so that /dev/random can re-seed on boot.
# */11  *       *       *       *       operator /usr/libexec/save-entropy

# Adjust the time zone if the CMOS clock keeps local time, as opposed to
# UTC time.  See adjkerntz(8) for details.
# 1,31  0-5     *       *       *       root    adjkerntz -a
```

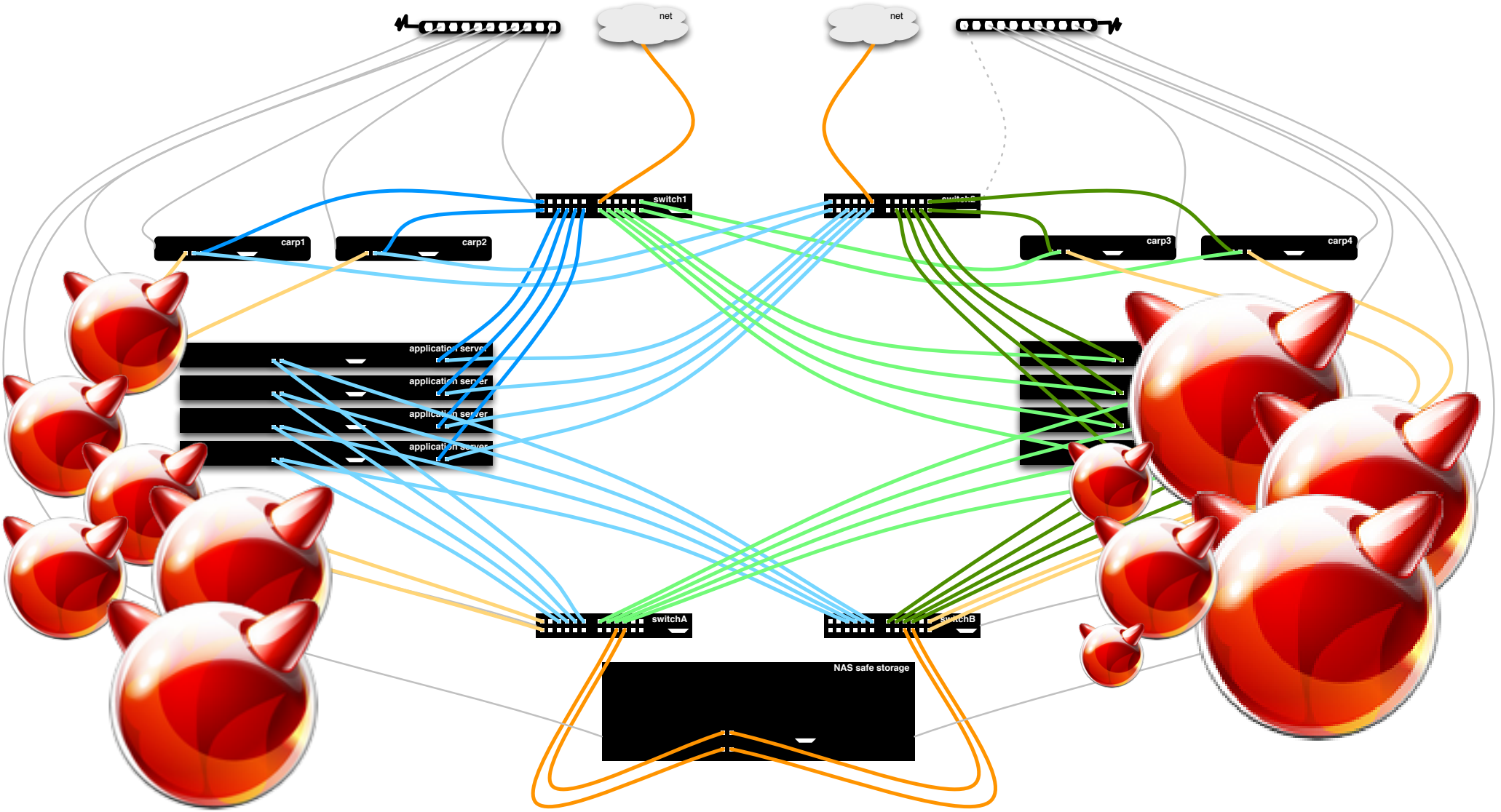
future directions...

important fun:

- CARP, from PF/OpenBSD
- GEOM
- NFS Improvements
- more NAS/SAN support (GEOM, ggated)
- FreeBSD 4.x, 5.x, 6.x, (7x!)

sick possibilities...

GEOM Gate, CARP, fun with failover jails...



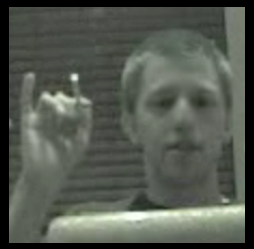
misc

- Compile md(4) into the kernel for File-Backed Disks, for better performance
- GOTCHA: rm a jail directory? chflags -R noschg jaildir


Go to Jail(8) Free Site — Jail(8) Community Website
http://jailing.net:8080/
Google
small text normal text large text
jailing.net
Search
news
members
you are not logged in log in join
navigation
Home
log in
Name
Password
Log in
Go to Jail(8) Free Site
jailing.net is a website dedicated to jail(8), a unix system to imprison a process and its descendants, used for virtualizing machine services by isolating an operating sytstem.
1) To centralize and keep track resources on the topic of jailing which exist on the internet
2) To provide a platform for new documentation and information about jailing
3) To provide a platform for individuals and orginizations involved with jailing, providing membership for content authorship
Last modified 2005-03-22 03:16
news
BSDCan in Canada!
2005-03-22
More...
upcoming events
BSDCan FreeBSD Developer Summit
Ottawa, Canada,
2005-05-11
BSDCan in Canada
Ottawa, Canada,
2005
2005
Su We Th Fr
1 2 5 6 7
8 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
BSD Daemon Copyright | Site Copyrights
Stillborn.
Suggestions?


NYCBUG
NEW YORK CITY *BSD USER GROUP
Jailing Party - jail(8)
Friday-Monday, Sept.3-6 2004
Next time we get more resiliant hardware...
Thanks everyone for all the fun!

Special Thanks:



wintermute (of iMeme), taught me to jail(8).
He's here somewhere- buy him a drink.

reality schooled me more BSD than he knows...

Poul-Henning Kamp wrote the jail feature for R&D Associates http://www.rndassociates.com/ who contributed it to FreeBSD around 1998.

Robert Watson wrote the extended documentation, found a few bugs, added a few new features, and cleaned up the userland jail environment.

ike is proud to be a part of the ***New York City *BSD Users Group***, and the ***Lower East Side Mac Unix Users Group***

NYCBUG
NEW YORK CITY *BSD USER GROUP

ike is proud to be a part of the ***New York City *BSD Users Group***, and the ***Lower East Side Mac Unix Users Group***

NYCBUG
NEW YORK CITY *BSD USER GROUP