Improving System Management With ZFS

Brooks Davis
brooks@aero.org

Enterprise Information Services
Technical Computing Services
19 March 2011

Monday, September 26, 2011

What is ZFS?

...
Most of you already have some idea.
But a quick overview is in order.

Zetabyte File System

AEROSPACE

Monday, September 26, 2011

expanding the acronym we get the Zetabyte File System.
which doesn't tell us much except that it's big!

More than a filesystem

In fact it is more than a file system

Data Protection and Integrity

AEROSPACE

combines RAID like data protection and check sum based integrity verification

Volume Management

with volume management

File Systems

supporting the creation of file systems

Block Devices

and block devices aka volumes.

Snapshots

AEROSPACE

Modern filesystem features like snapshots.

Single interface for storage management

AEROSPACE

...

ZFS History

- September 2004: Announced
- November 2005: Released in OpenSolaris
- June 2006: Released in Solaris
- April 2007: Committed to FreeBSD

AEROSPACE

How Does ZFS Work?

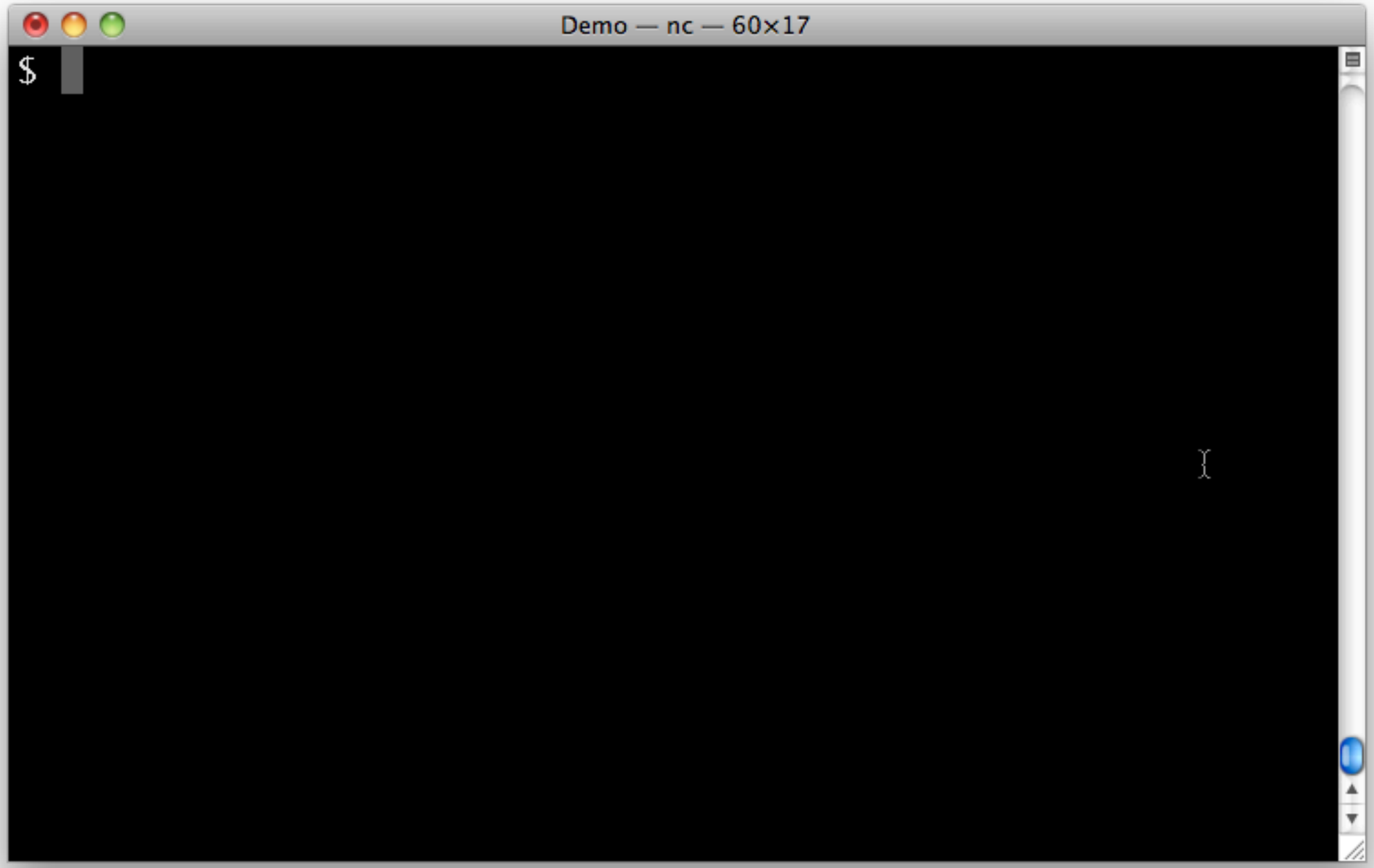
AEROSPACE

Quick tutorial on general ZFS functionality

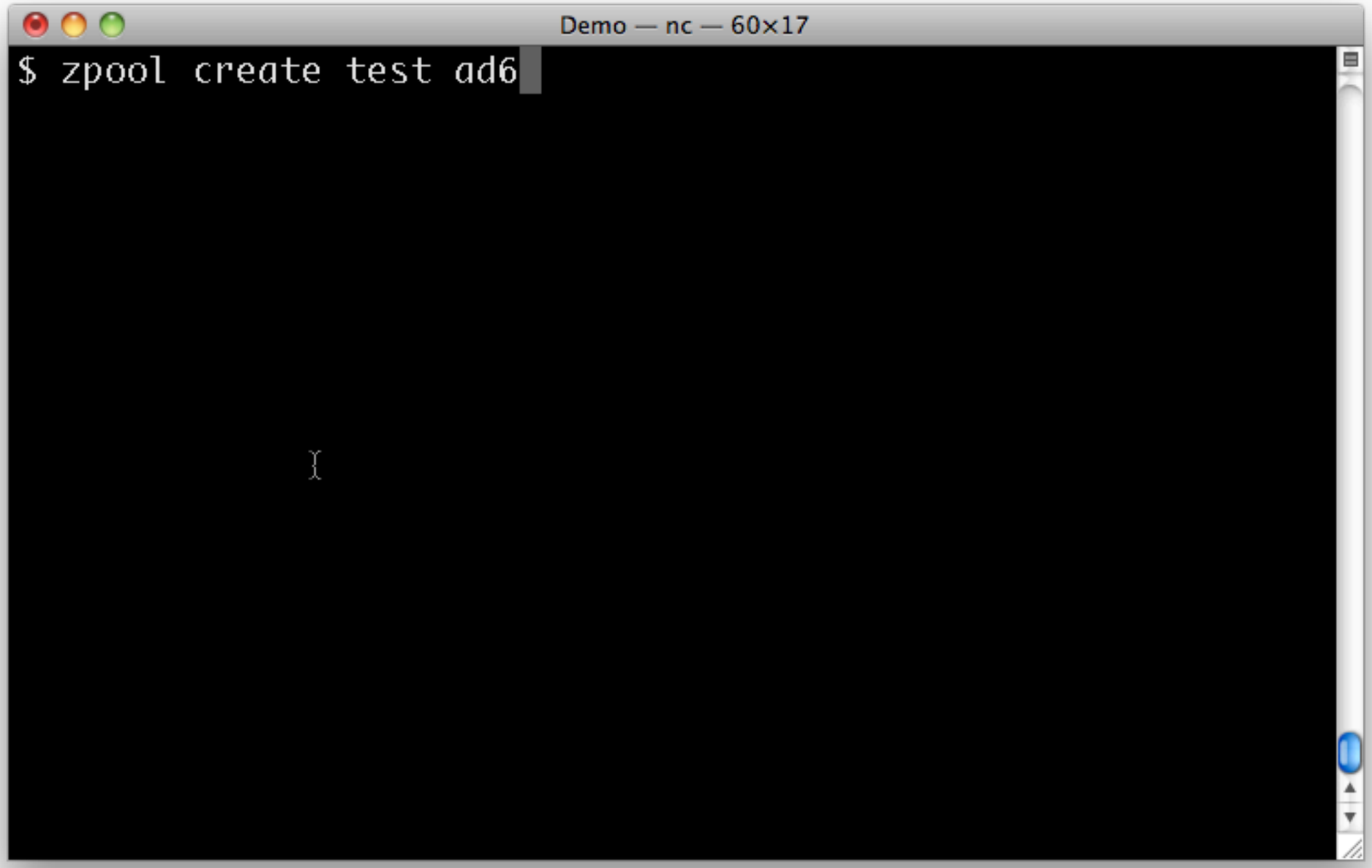
zpool(1)

AEROSPACE

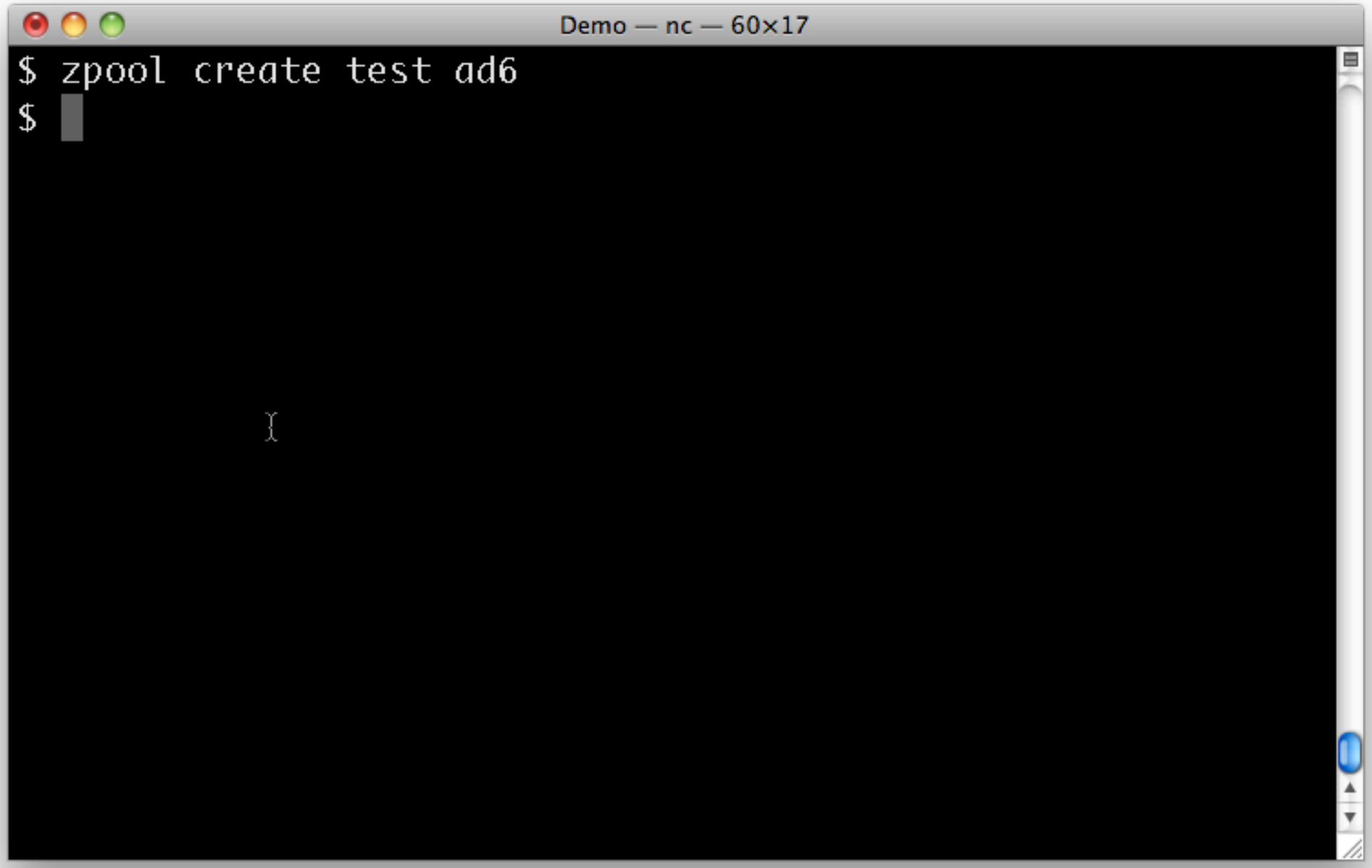
We'll start at the bottom with storage pools.
Storage pools are created and managed by the zpool(1) command



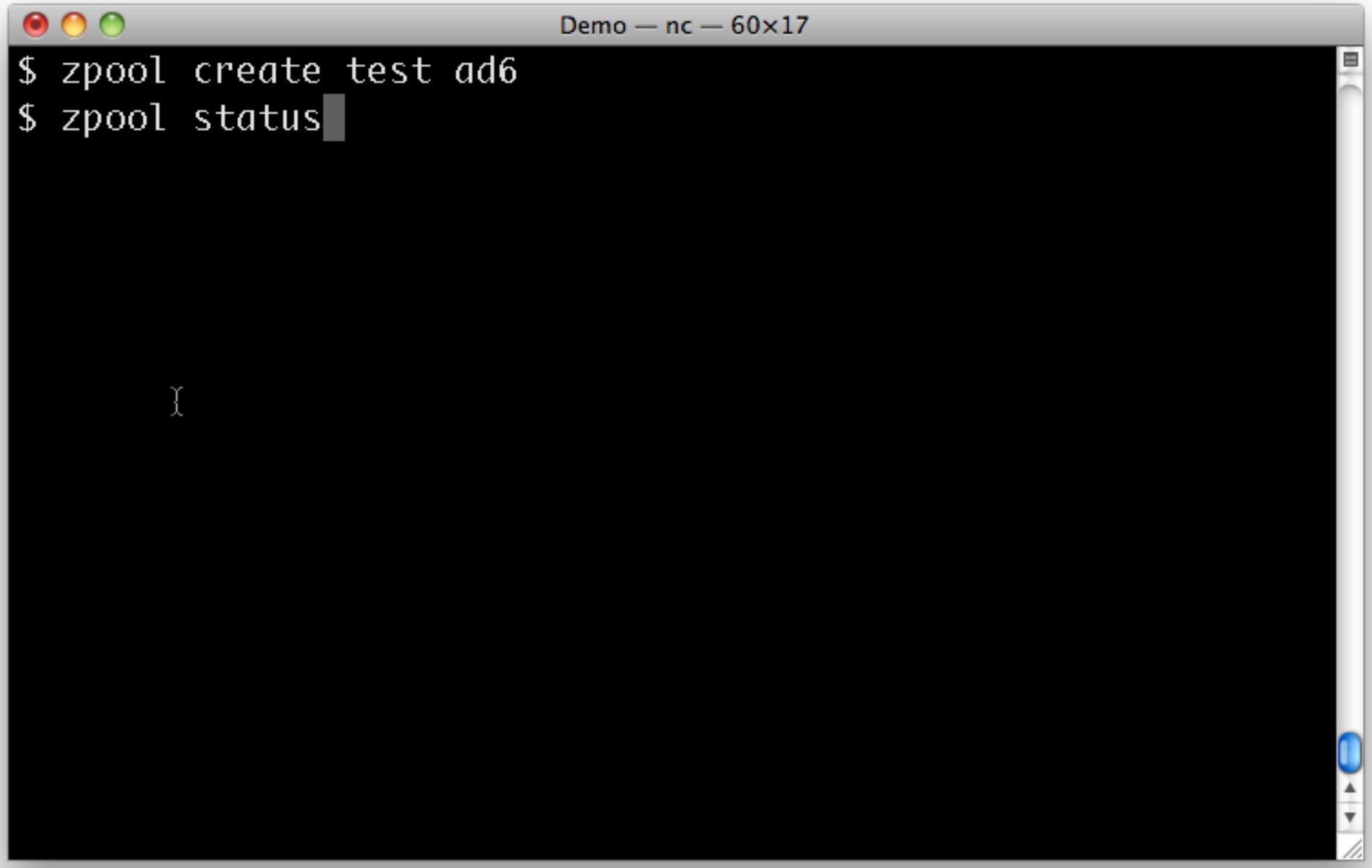
AEROSPACE



AEROSPACE



AEROSPACE

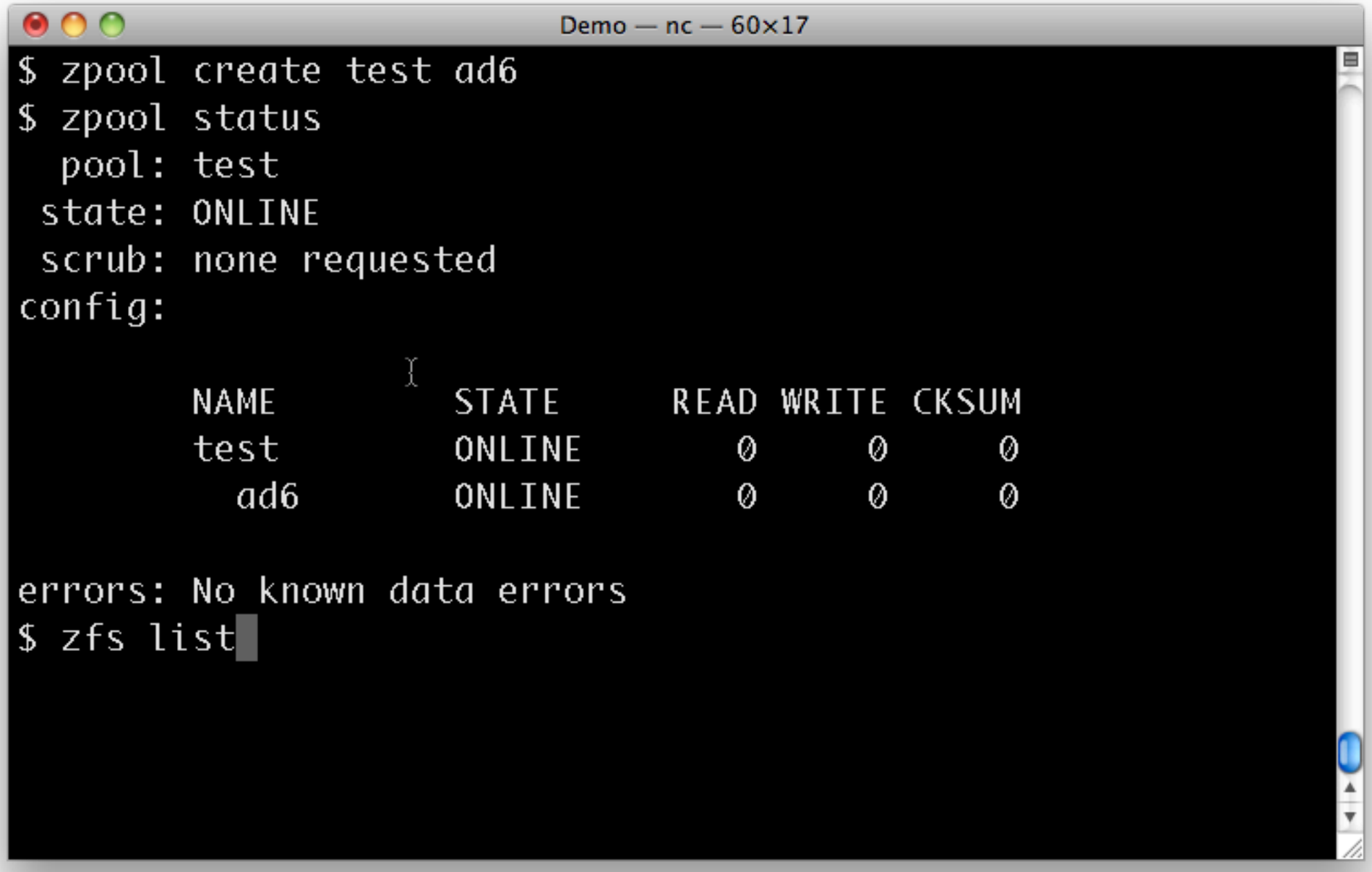


AEROSPACE

Demo — nc — 60×17

```
$ zpool create test ad6
$ zpool status
  pool: test
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        test        ONLINE       0     0     0
          ad6       ONLINE       0     0     0

errors: No known data errors
$
```

AEROSPACE


Demo — nc — 60×17
$ zpool create test ad6
$ zpool status
pool: test
state: ONLINE
scrub: none requested
config:
NAME STATE READ WRITE CKSUM
test ONLINE 0 0 0
ad6 ONLINE 0 0 0
errors: No known data errors
$ zfs list

AEROSPACE

Demo — nc — 60×17

```
$ zpool create test ad6
$ zpool status
  pool: test
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        test        ONLINE       0     0     0
          ad6       ONLINE       0     0     0

errors: No known data errors
$ zfs list
NAME   USED  AVAIL  REFER  MOUNTPOINT
test  70.5K   457G    21K  /test
$
```

AEROSPACE

Demo — nc — 60×17

```
$ zpool create test ad6
$ zpool status
  pool: test
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        test        ONLINE       0     0     0
          ad6       ONLINE       0     0     0

errors: No known data errors
$ zfs list
NAME   USED  AVAIL  REFER  MOUNTPOINT
test  70.5K   457G    21K  /test
$ mount -t zfs
```

AEROSPACE

Demo — nc — 60×17

```
$ zpool status
  pool: test
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        test        ONLINE       0     0     0
          ad6       ONLINE       0     0     0

errors: No known data errors
$ zfs list
NAME   USED  AVAIL  REFER  MOUNTPOINT
test  70.5K   457G    21K  /test
$ mount -t zfs
test on /test (zfs, local)
$
```

AEROSPACE

pools are collections of vdevs

AEROSPACE

In the previous example ad6 is a real and virtual device

vdev types

- device: disk in Solaris, GEOM in FreeBSD
- mirror: RAID 1-like group of devices
- raidz1: single parity protection
- raidz2: double parity protection
- raidz3: triple parity protection

AEROSPACE

Monday, September 26, 2011

raidz1 -> RAID5
raidz2 -> RAID6
Allowing 1 and 2 drive failures to be recovered
ZFS is good at recovering data from partially damaged disks due to it's use of check summing

special vdev types

- spare: device waiting to replace a failing device
- log: intent log device
- cache: read cache

AEROSPACE

Monday, September 26, 2011

Spares are used by management infrastructure to replace disks when they fail.
log devices speed up writes and can (should!) be mirrored
cache devices speed up reads and can only be normal devices

What if we need more storage?

AEROSPACE

back to the examples

Demo — nc — 60×17

```
$ zpool status
  pool: test
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        test        ONLINE       0     0     0
          ad6       ONLINE       0     0     0

errors: No known data errors
$ zfs list
NAME   USED  AVAIL  REFER  MOUNTPOINT
test  70.5K   457G    21K  /test
$ mount -t zfs
test on /test (zfs, local)
$
```

AEROSPACE

Monday, September 26, 2011

picking up where we left off
adding space is trivial, we just add a vdev

Demo — nc — 60×17

```
$ zpool status
  pool: test
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        test        ONLINE       0     0     0
          ad6       ONLINE       0     0     0

errors: No known data errors
$ zfs list
NAME   USED  AVAIL  REFER  MOUNTPOINT
test  70.5K   457G    21K  /test
$ mount -t zfs
test on /test (zfs, local)
$ zpool add test ad7
```

AEROSPACE

Demo — nc — 60×17

```
  pool: test
 state: ONLINE
 scrub: none requested
config:

        NAME         STATE     READ WRITE CKSUM
        test         ONLINE       0     0     0
          ad6        ONLINE       0     0     0

errors: No known data errors
$ zfs list
NAME   USED  AVAIL  REFER  MOUNTPOINT
test  70.5K   457G    21K  /test
$ mount -t zfs
test on /test (zfs, local)
$ zpool add test ad7
$
```

AEROSPACE

Demo — nc — 60×17

```
  pool: test
 state: ONLINE
 scrub: none requested
config:

        NAME         STATE     READ WRITE CKSUM
        test         ONLINE       0     0     0
          ad6        ONLINE       0     0     0

errors: No known data errors
$ zfs list
NAME   USED  AVAIL  REFER  MOUNTPOINT
test  70.5K   457G    21K  /test
$ mount -t zfs
test on /test (zfs, local)
$ zpool add test ad7
$ zpool status
```

Demo — nc — 60×17

```
test  70.5K   457G    21K  /test
$ mount -t zfs
test on /test (zfs, local)
$ zpool add test ad7
$ zpool status
  pool: test
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        test        ONLINE       0     0     0
          ad6       ONLINE       0     0     0
          ad7       ONLINE       0     0     0

errors: No known data errors
$
```

AEROSPACE

Demo — nc — 60×17

```
test  70.5K   457G    21K  /test
$ mount -t zfs
test on /test (zfs, local)
$ zpool add test ad7
$ zpool status
  pool: test
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        test        ONLINE       0     0     0
          ad6       ONLINE       0     0     0
          ad7       ONLINE       0     0     0

errors: No known data errors
$ zfs list
```

AEROSPACE

Demo — nc — 60×17

```
$ zpool add test ad7
$ zpool status
  pool: test
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        test        ONLINE       0     0     0
          ad6       ONLINE       0     0     0
          ad7       ONLINE       0     0     0

errors: No known data errors
$ zfs list
NAME   USED  AVAIL  REFER  MOUNTPOINT
test    75K   913G    21K  /test
$
```

AEROSPACE

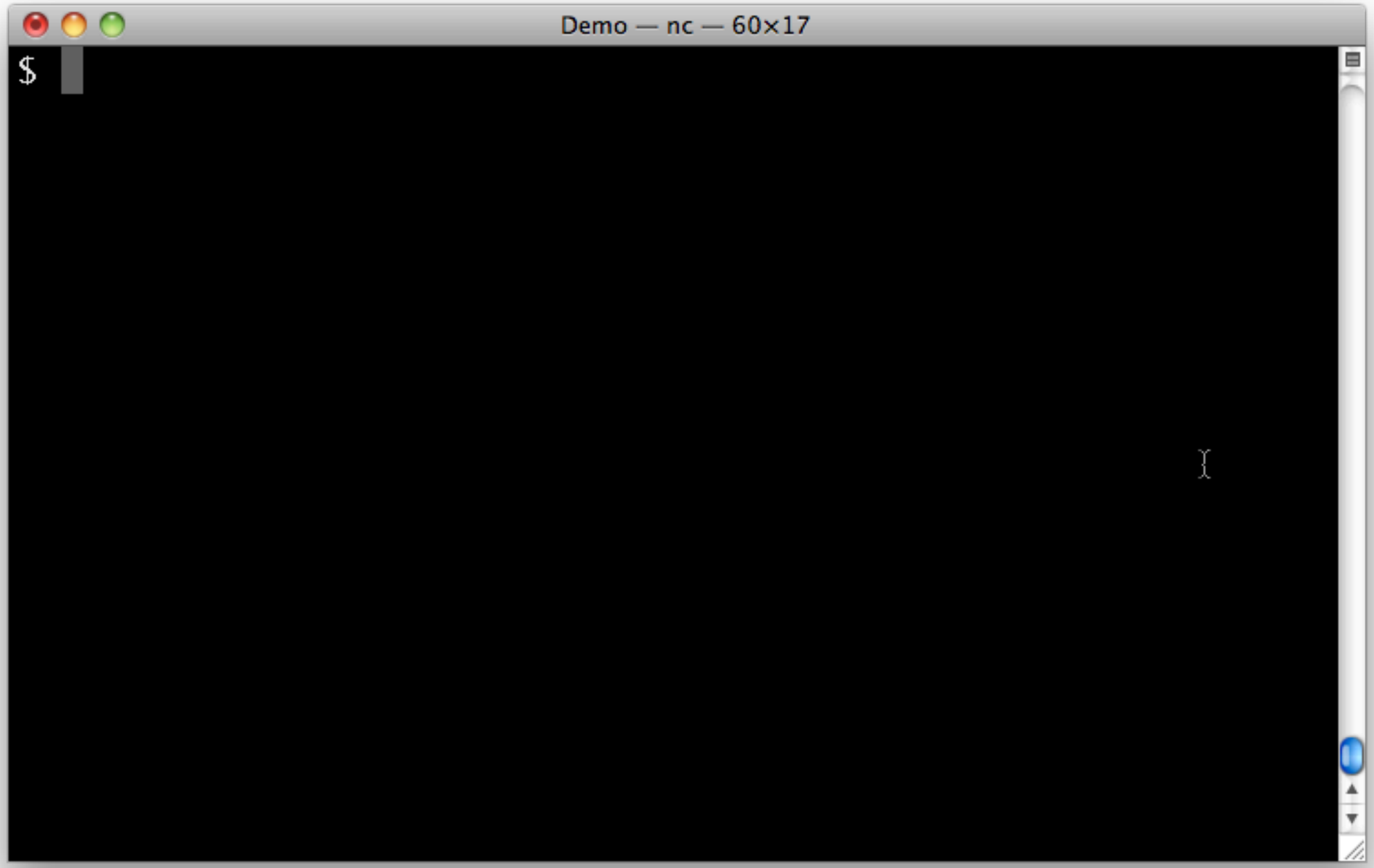
Monday, September 26, 2011

That's really all there is do it. Of course, we've just create a RAID0-like system which probably isn't what we want.

What if you actually like your data?

AEROSPACE

So ...
and want to keep it around?
I won't go in to much more detail
here's a quick example of mirroring for more realism



AEROSPACE

AEROSPACE

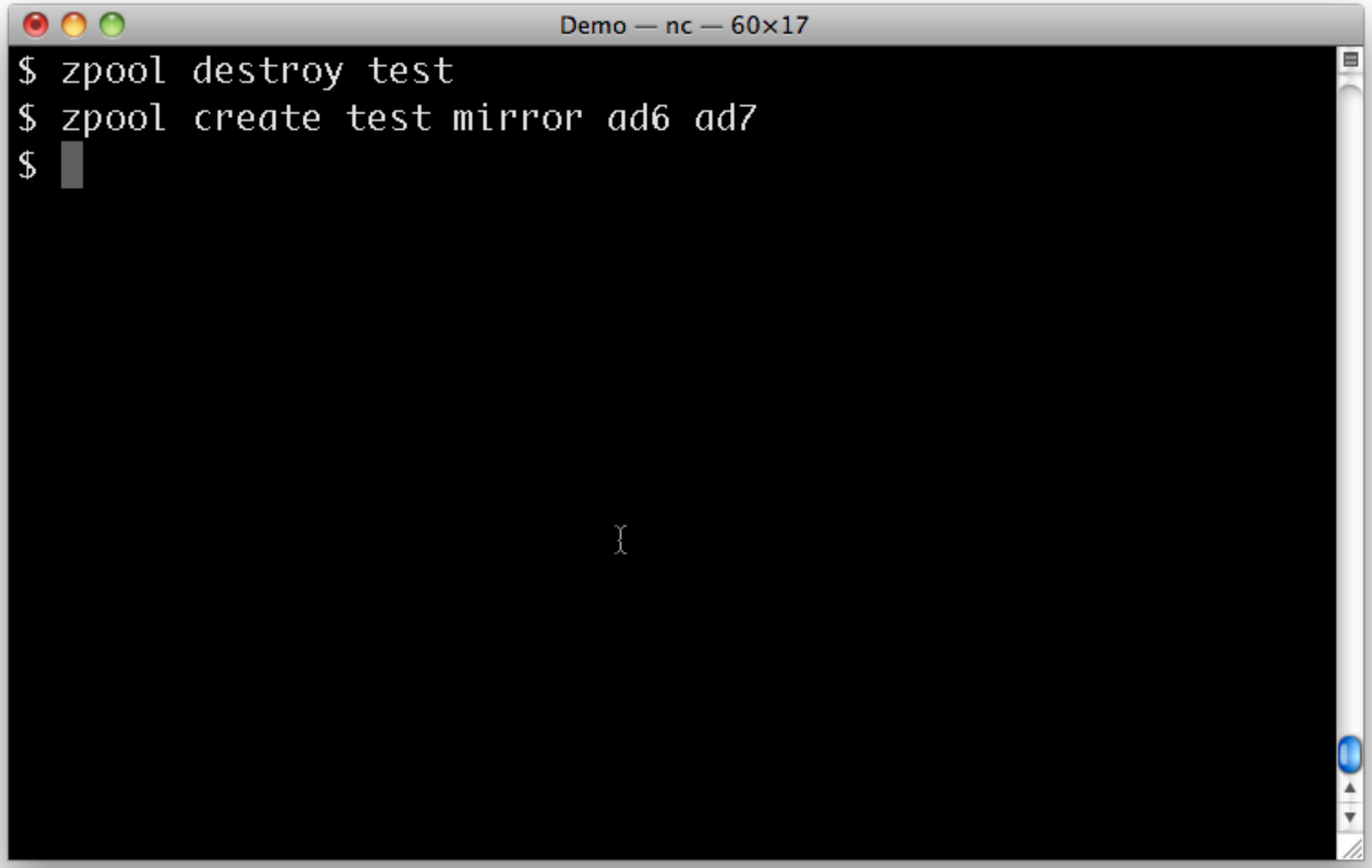
AEROSPACE

Monday, September 26, 2011

I always find this a little unnerving.
You can wipe our terabytes of data as easily as this empty test volume.


Demo — nc — 60×17
$ zpool destroy test
$ zpool create test mirror ad6 ad7

AEROSPACE


Demo — nc — 60×17
$ zpool destroy test
$ zpool create test mirror ad6 ad7
$

AEROSPACE


Demo — nc — 60×17
$ zpool destroy test
$ zpool create test mirror ad6 ad7
$ zpool status

AEROSPACE

Demo — nc — 60×17

```
$ zpool destroy test
$ zpool create test mirror ad6 ad7
$ zpool status
  pool: test
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        test        ONLINE       0     0     0
          mirror    ONLINE       0     0     0
            ad6     ONLINE       0     0     0
            ad7     ONLINE       0     0     0

errors: No known data errors
$
```

AEROSPACE

Demo — nc — 60×17

```
$ zpool destroy test
$ zpool create test mirror ad6 ad7
$ zpool status
  pool: test
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        test        ONLINE       0     0     0
          mirror    ONLINE       0     0     0
            ad6     ONLINE       0     0     0
            ad7     ONLINE       0     0     0

errors: No known data errors
$ zfs list
```

AEROSPACE

Demo — nc — 60×17

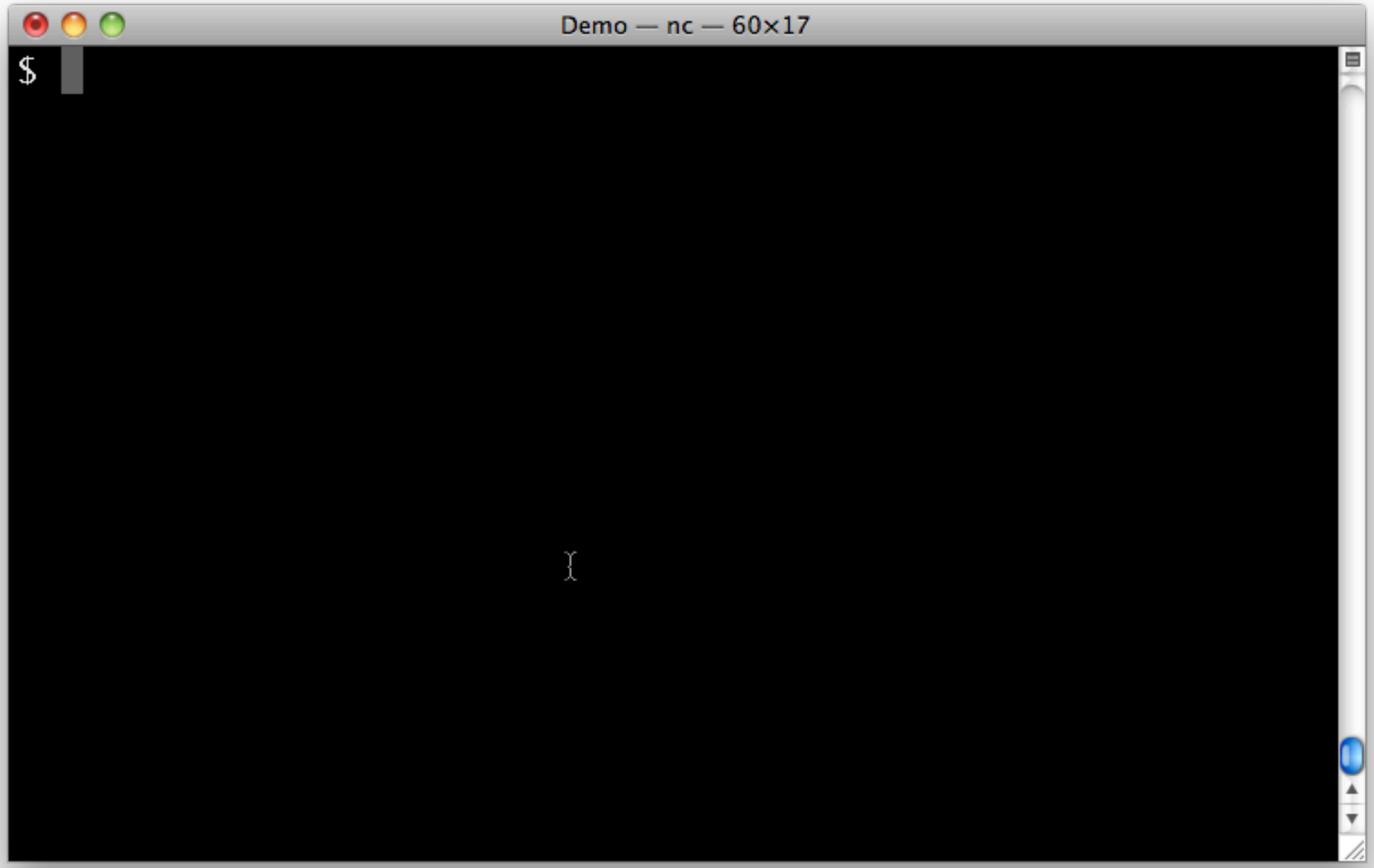
```
$ zpool status
  pool: test
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        test        ONLINE       0     0     0
          mirror    ONLINE       0     0     0
            ad6     ONLINE       0     0     0
            ad7     ONLINE       0     0     0

errors: No known data errors
$ zfs list
NAME   USED  AVAIL  REFER  MOUNTPOINT
test    72K   457G    21K  /test
$
```

AEROSPACE

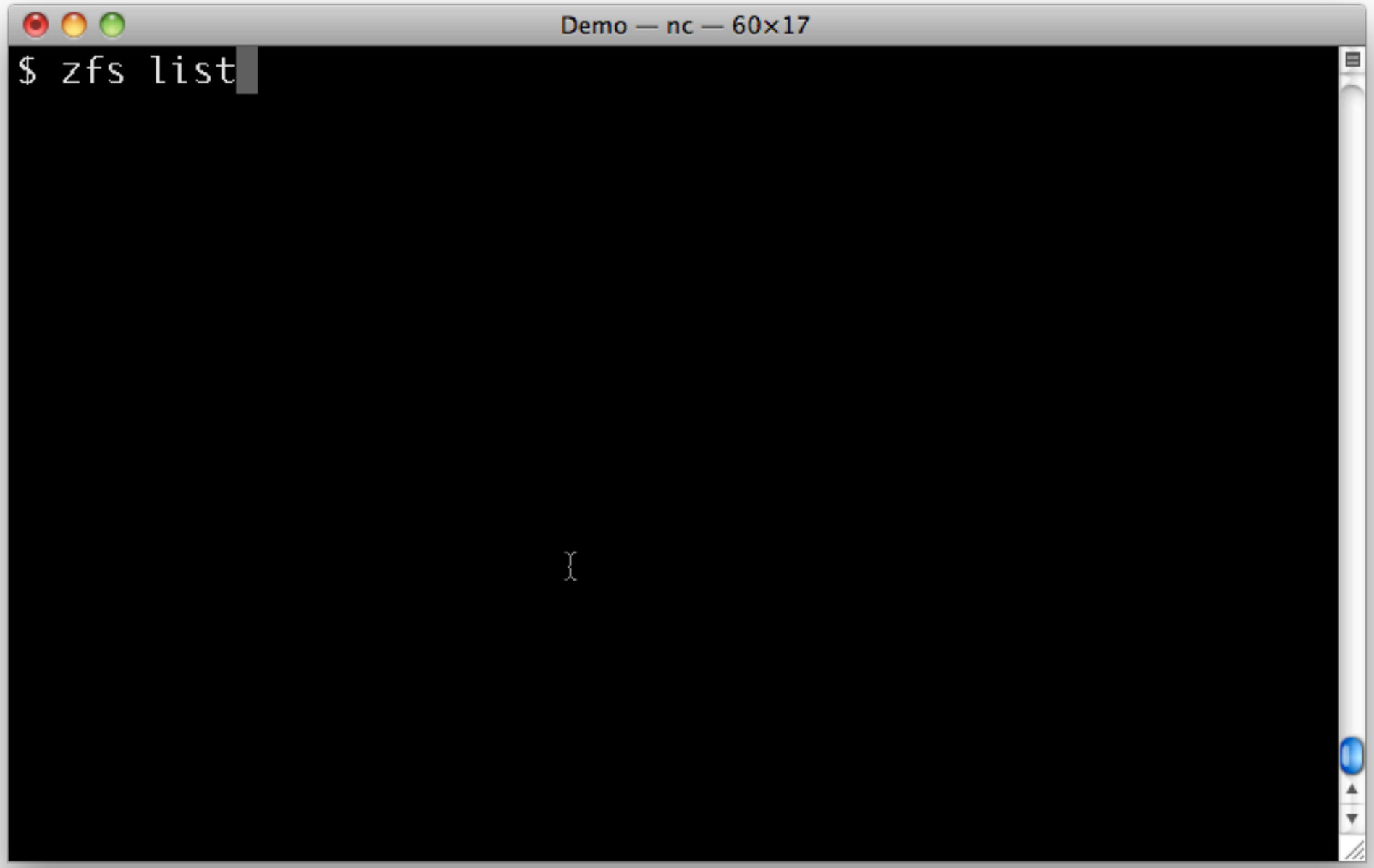
Verifying data integrity

AEROSPACE

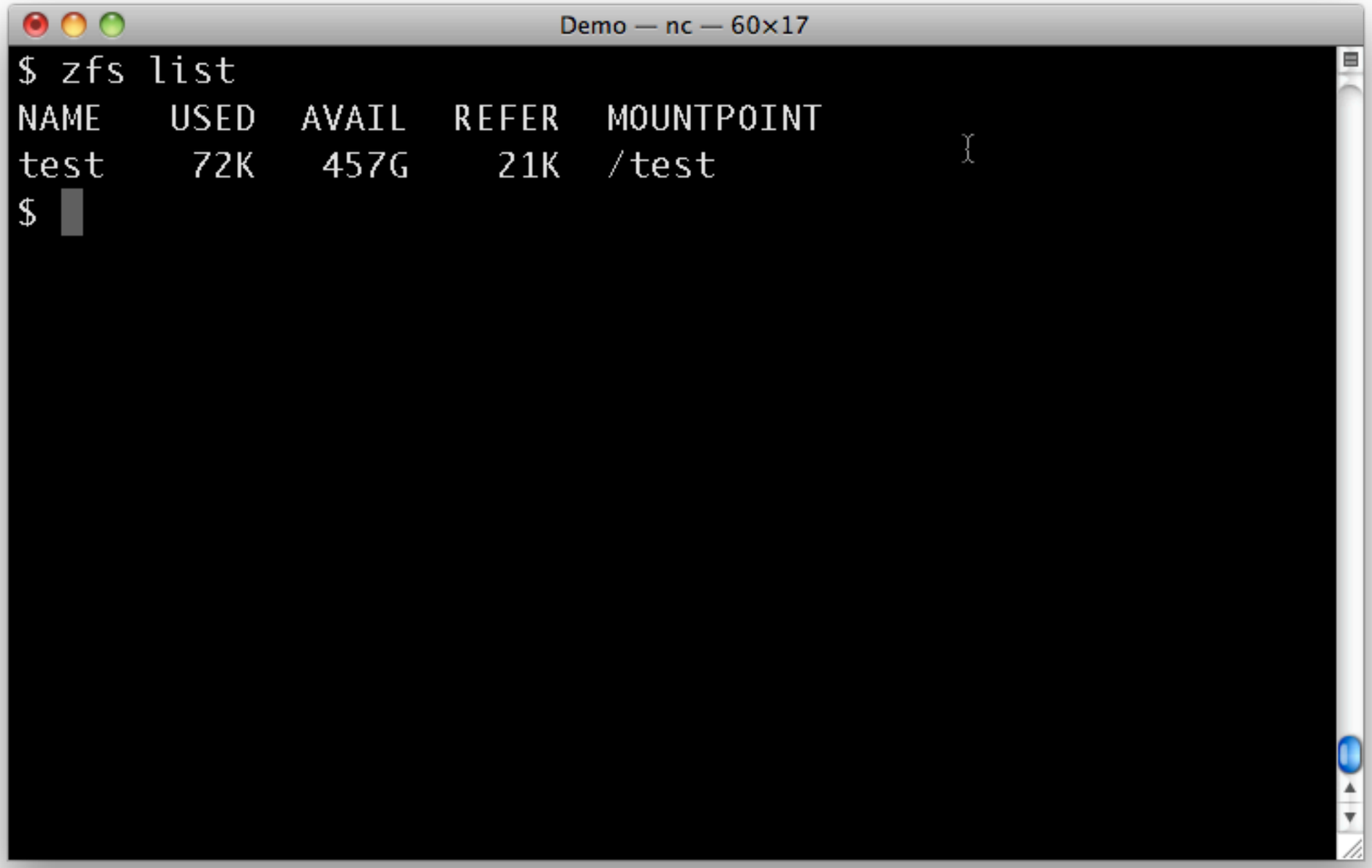
One of the more useful features of ZFS is that all data is protected by checksums. Those are verified as data is read but can also be verified using the “zpool scrub” command.



AEROSPACE

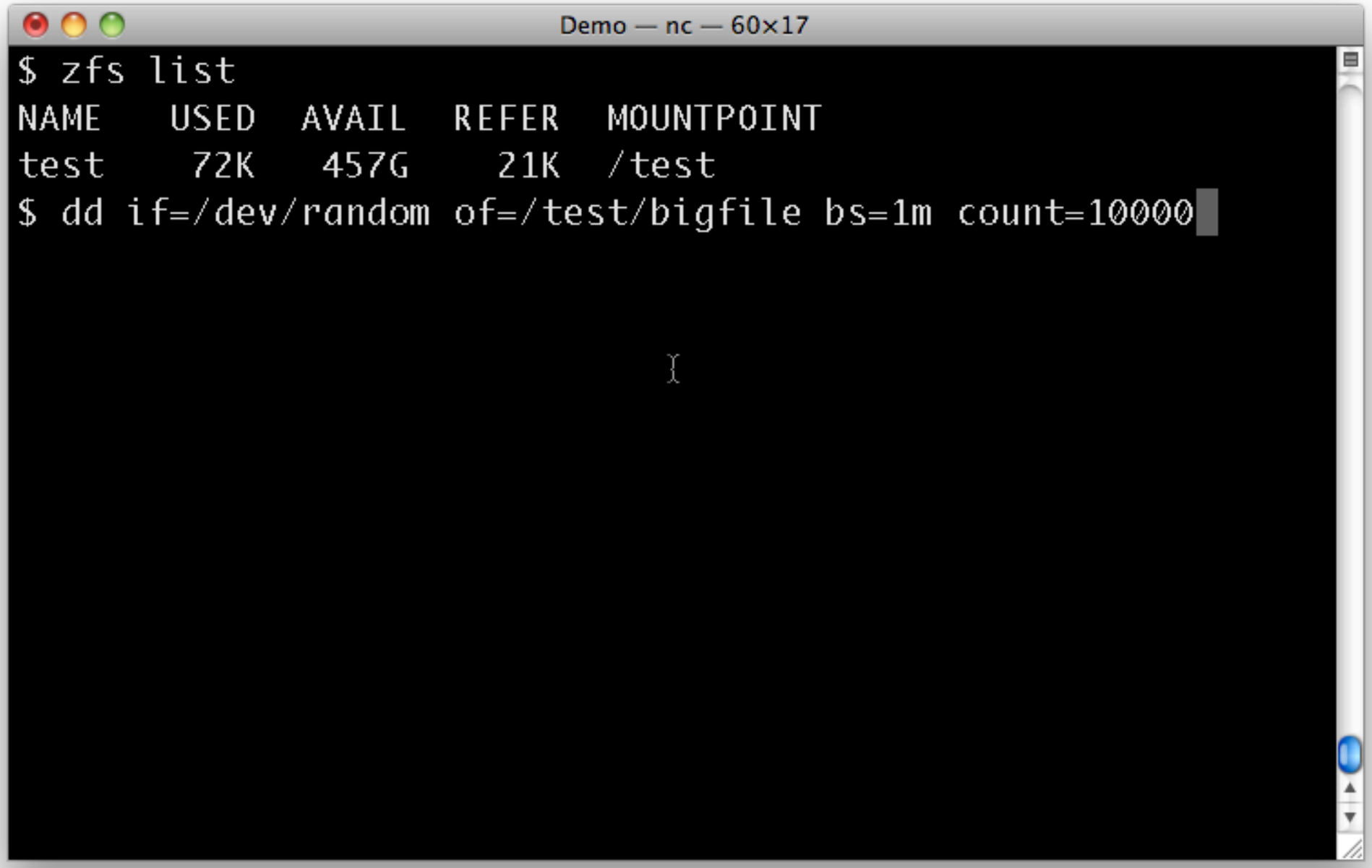


AEROSPACE



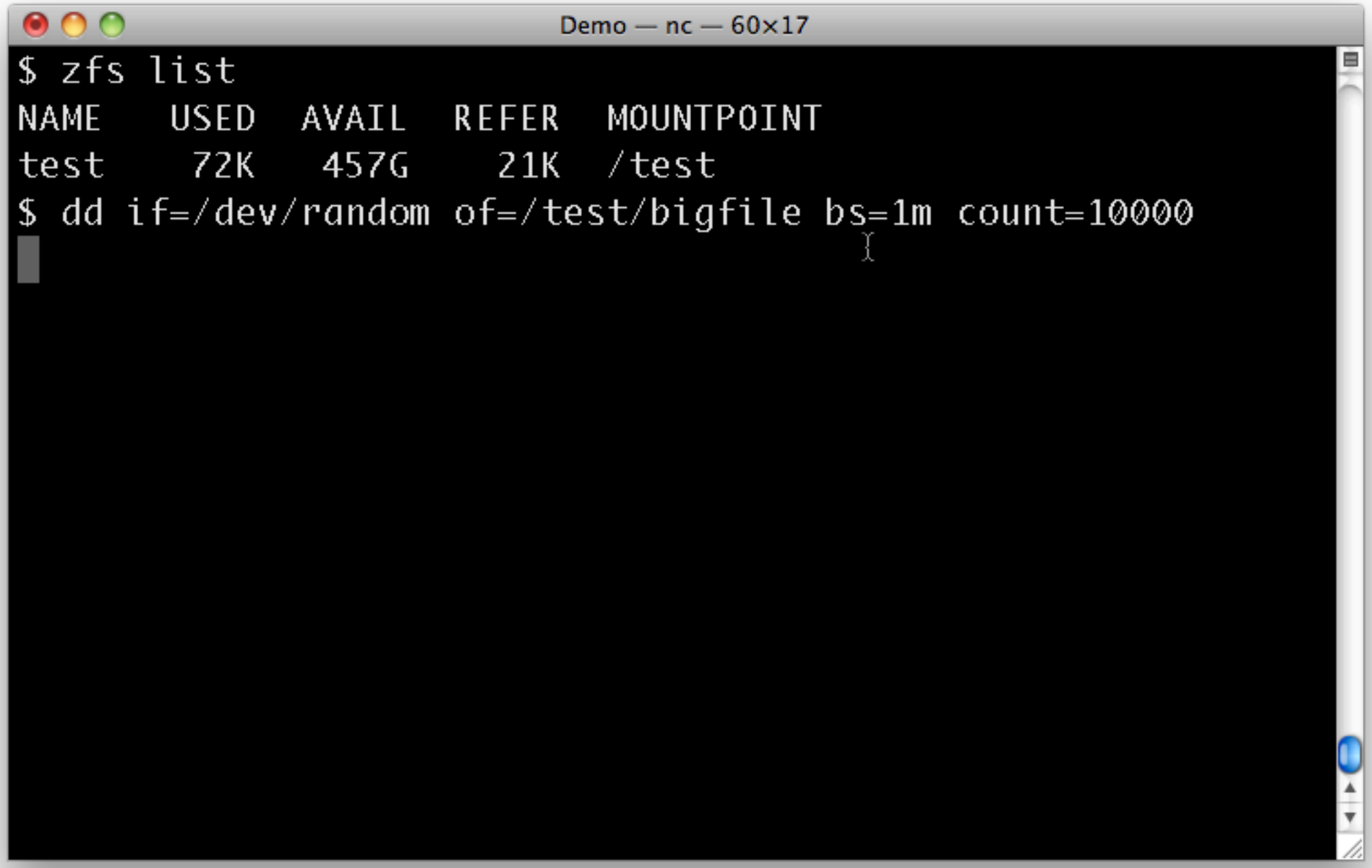
AEROSPACE

We've got out storage pool

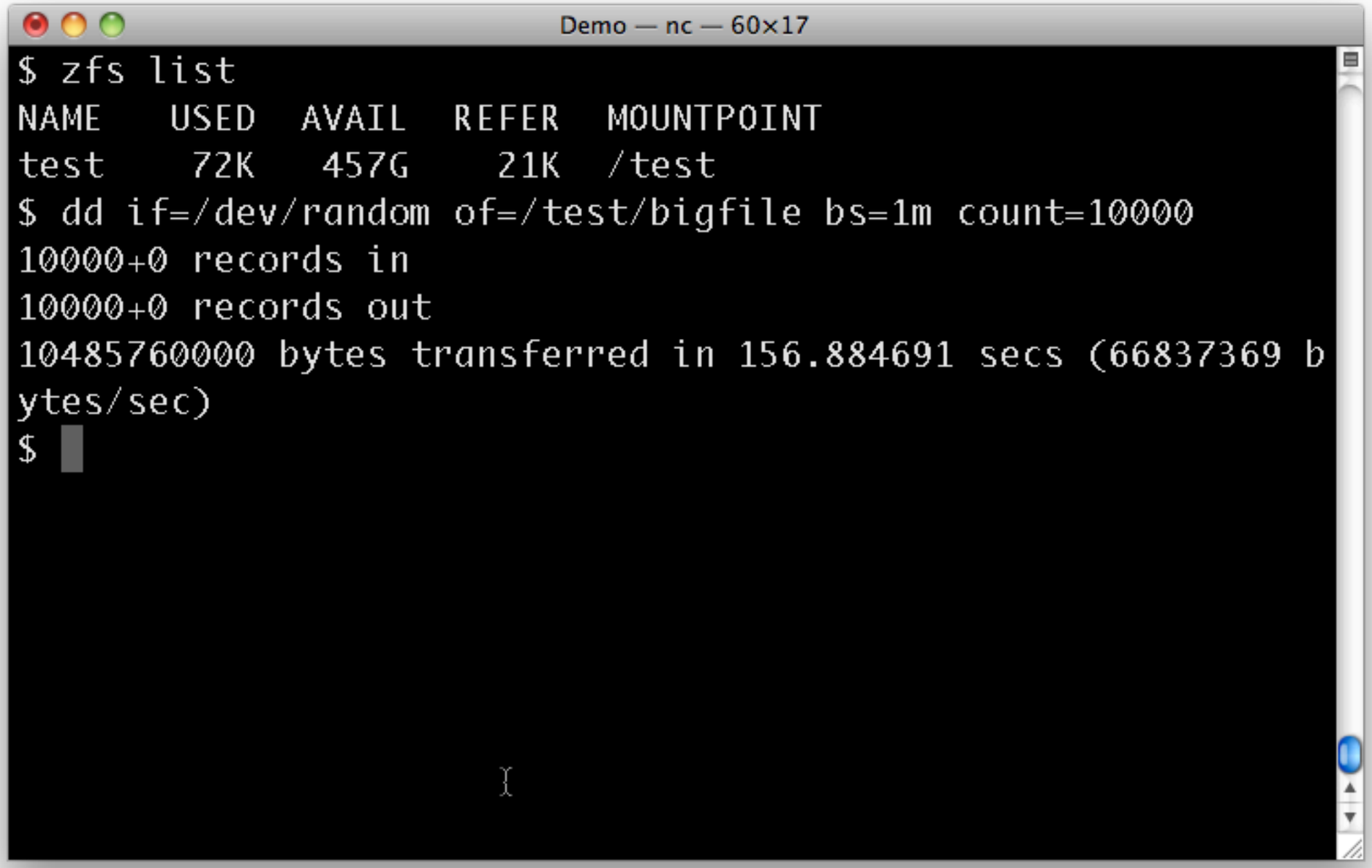


AEROSPACE

Now let’s put some data on it.

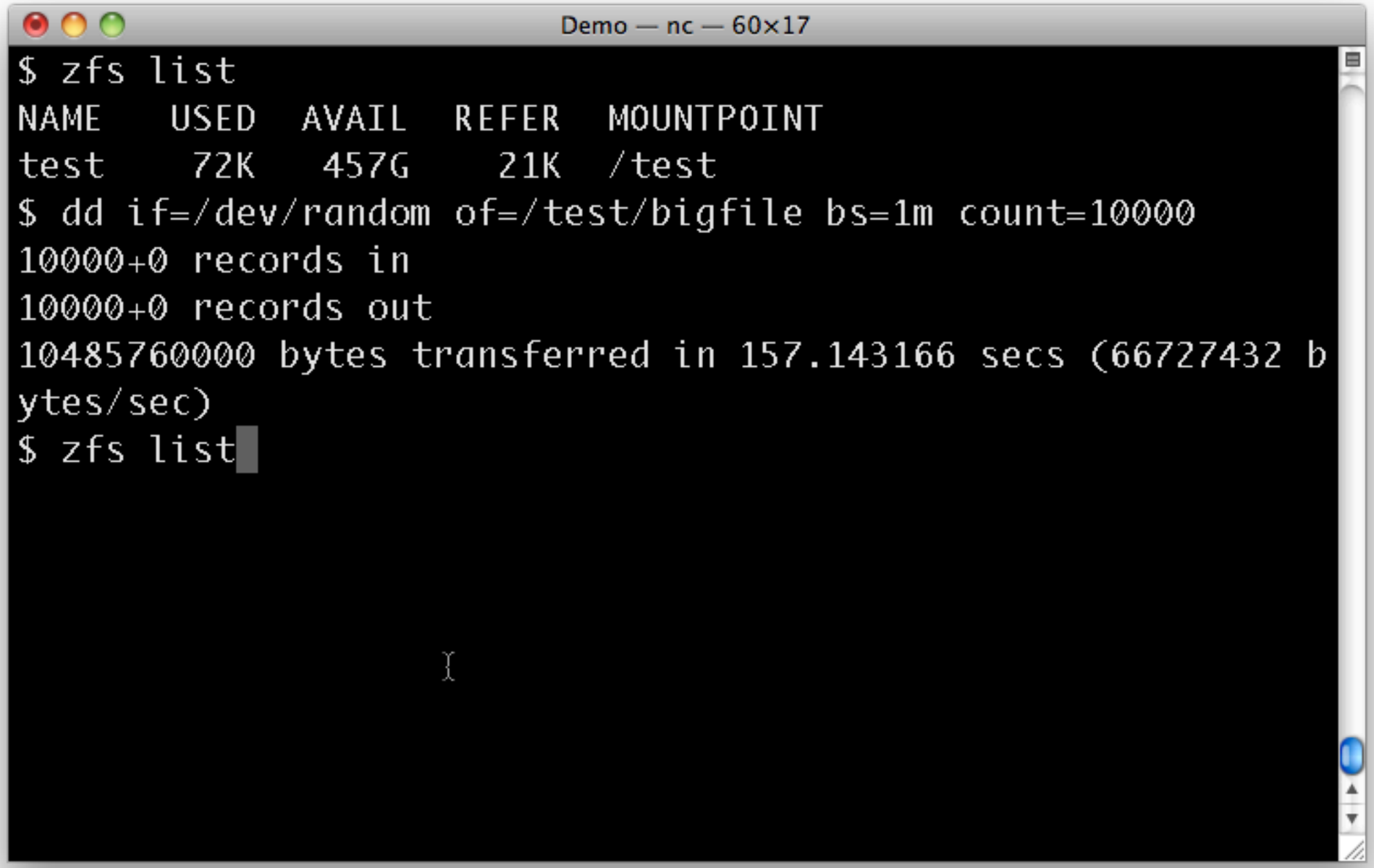

Demo — nc — 60×17
$ zfs list
NAME   USED AVAIL REFER MOUNTPOINT
test   72K   457G   21K /test
$ dd if=/dev/random of=/test/bigfile bs=1m count=10000

AEROSPACE



AEROSPACE

I picked 10GB arbitrarily here because it took a reasonable amount of time to write and verify.

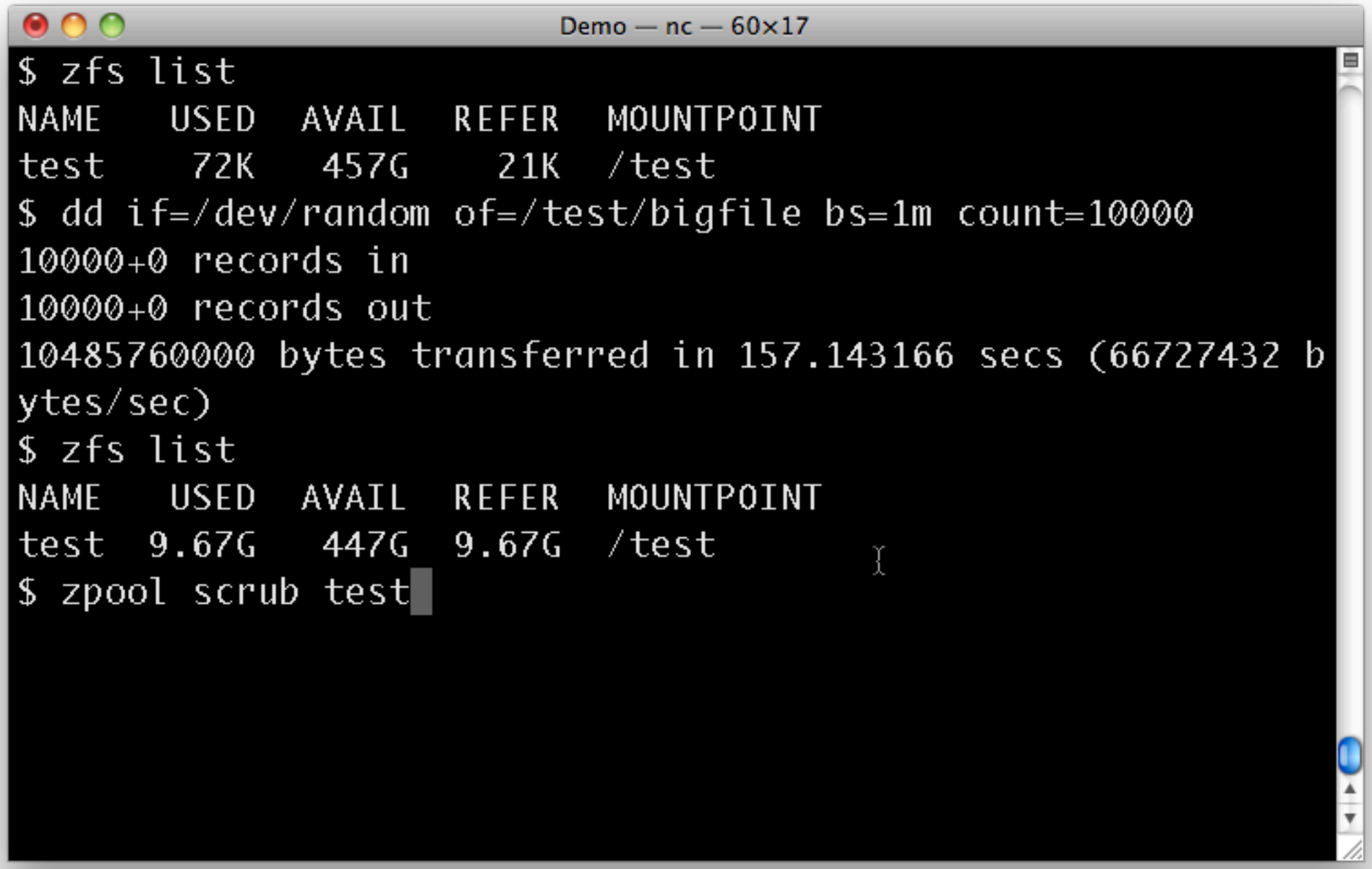

Demo — nc — 60×17
$ zfs list
NAME   USED AVAIL REFER MOUNTPOINT
test   72K   457G   21K /test
$ dd if=/dev/random of=/test/bigfile bs=1m count=10000
10000+0 records in
10000+0 records out
10485760000 bytes transferred in 157.143166 secs (66727432 b
ytes/sec)
$ zfs list

AEROSPACE

Demo — nc — 60×17

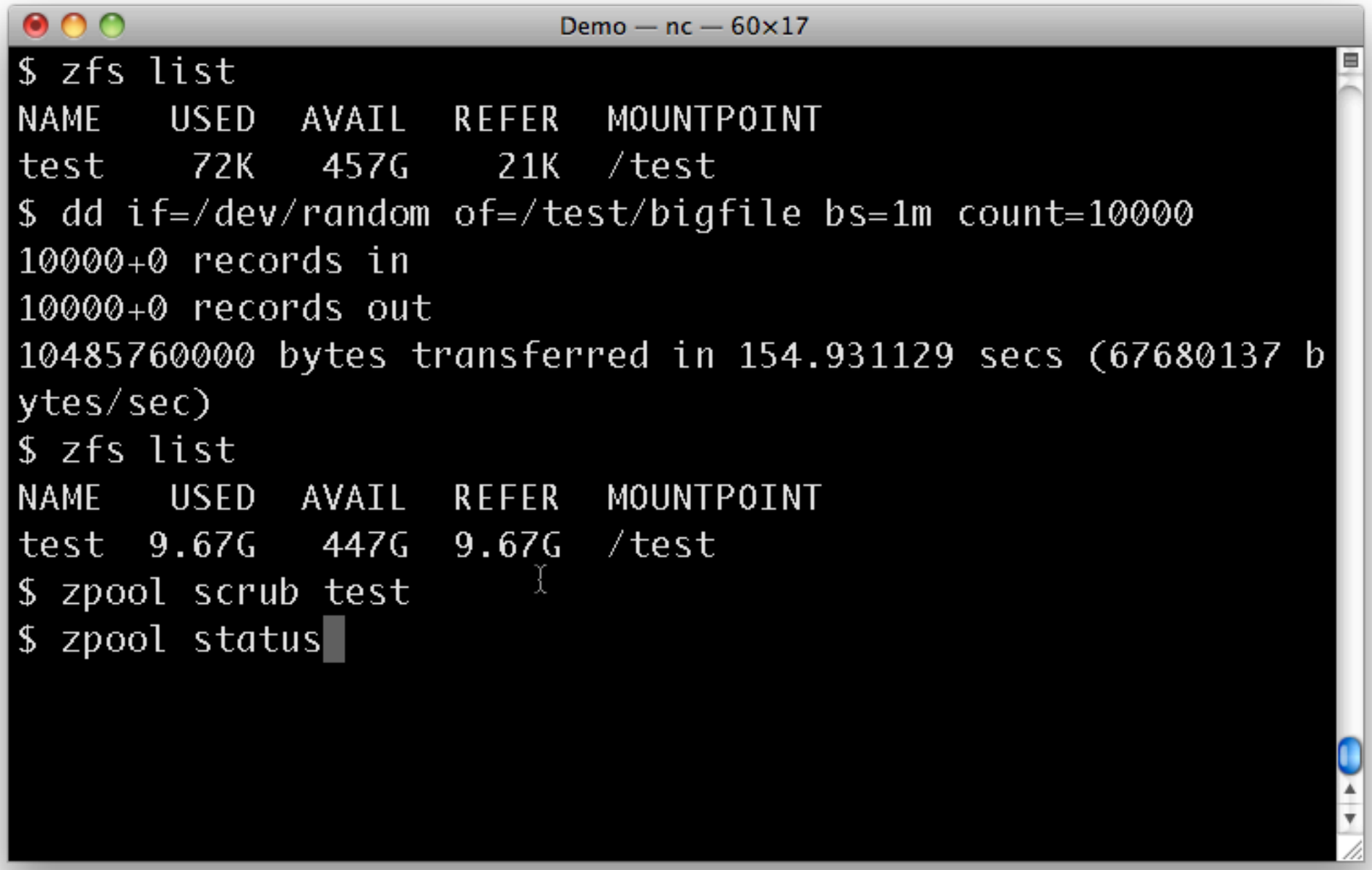
```
$ zfs list
NAME   USED  AVAIL  REFER  MOUNTPOINT
test    72K   457G    21K  /test
$ dd if=/dev/random of=/test/bigfile bs=1m count=10000
10000+0 records in
10000+0 records out
10485760000 bytes transferred in 157.143166 secs (66727432 b
ytes/sec)
$ zfs list
NAME   USED  AVAIL  REFER  MOUNTPOINT
test  9.67G   447G  9.67G  /test
$
```

AEROSPACE



AEROSPACE

Now we kick off a scrub.



AEROSPACE

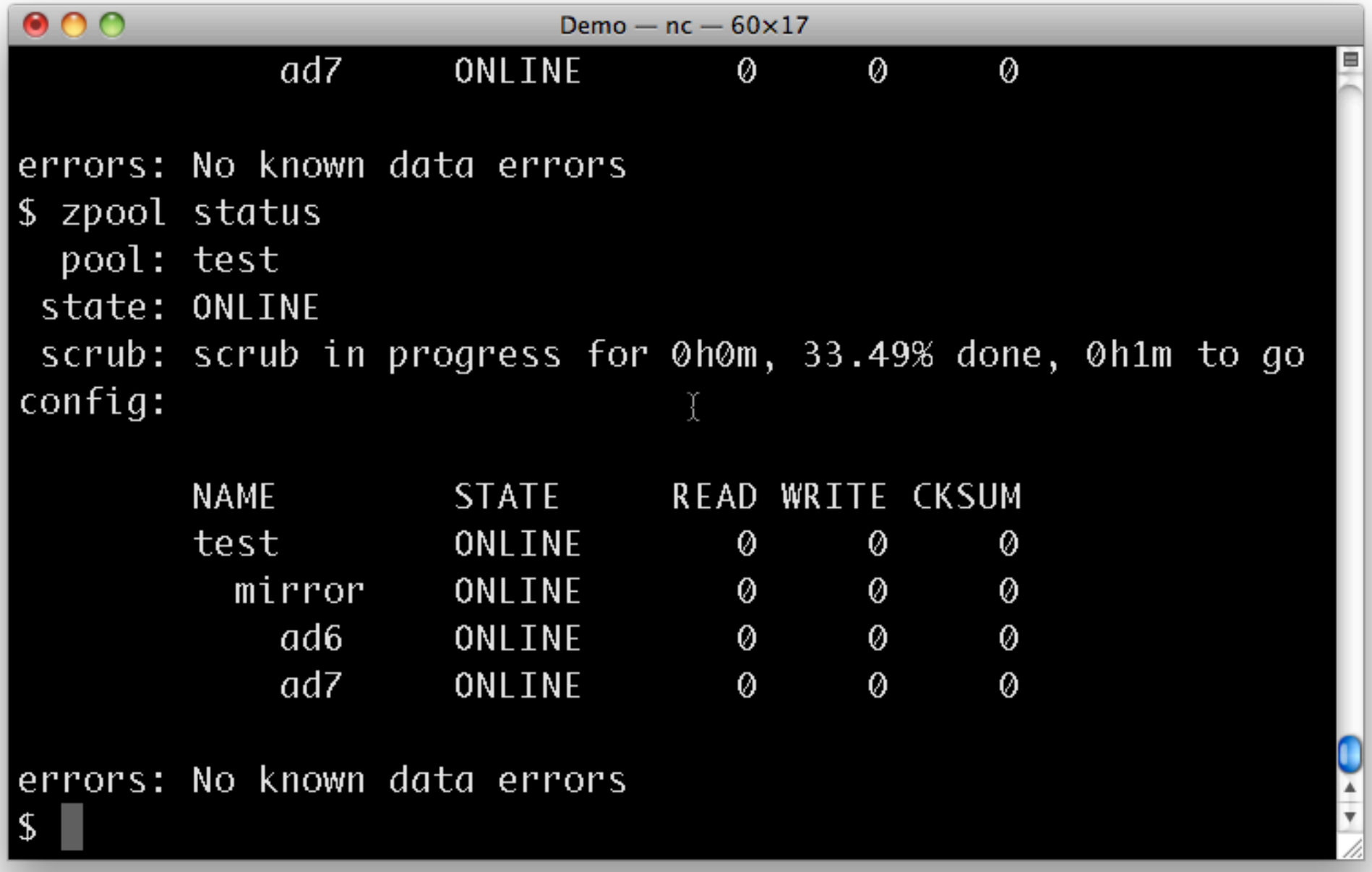
We can watch the status of the scrub with the zpool status command

Demo — nc — 60×17

```
NAME   USED  AVAIL  REFER  MOUNTPOINT
test  9.77G   447G  9.77G  /test
$ zpool scrub test
$ zpool status
  pool: test
 state: ONLINE
 scrub: scrub in progress for 0h0m, 10.88% done, 0h1m to go
config:

        NAME         STATE     READ WRITE CKSUM
        test         ONLINE       0     0     0
          mirror     ONLINE       0     0     0
            ad6      ONLINE       0     0     0
            ad7      ONLINE       0     0     0

errors: No known data errors
$
```

AEROSPACE

I highly recommend this process for any storage system even one you don't plan to use ZFS on.



AEROSPACE

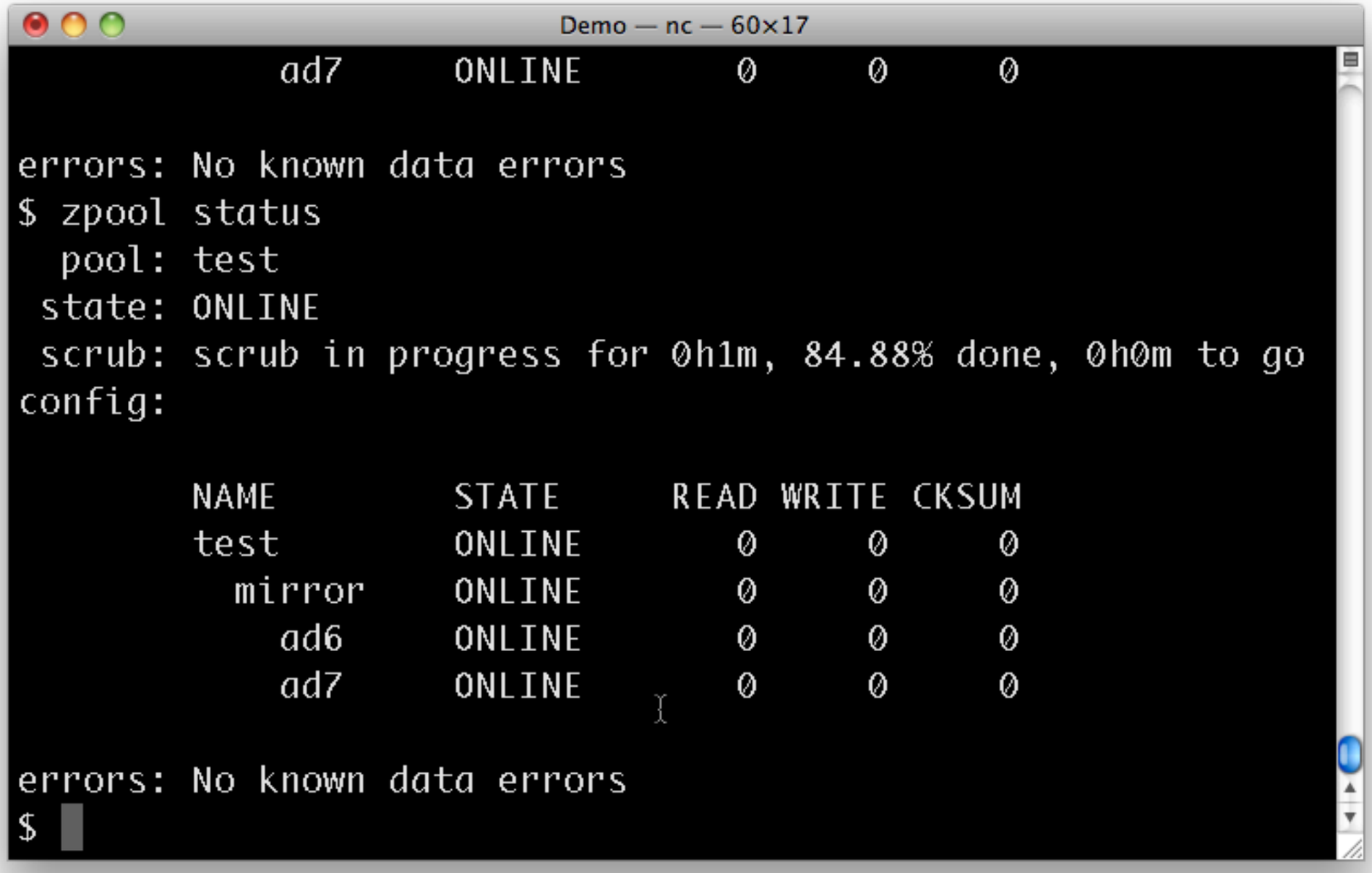
With a test similar to this we could reliably trigger data corruption on a Sun branded Adaptec RAID controller.

Demo — nc — 60×17

```
            ad7     ONLINE       0     0     0

errors: No known data errors
$ zpool status
  pool: test
 state: ONLINE
 scrub: scrub in progress for 0h1m, 52.99% done, 0h0m to go
config:

        NAME         STATE     READ WRITE CKSUM
        test         ONLINE       0     0     0
          mirror     ONLINE       0     0     0
            ad6      ONLINE       0     0     0
            ad7      ONLINE       0     0     0

errors: No known data errors
$
```

AEROSPACE

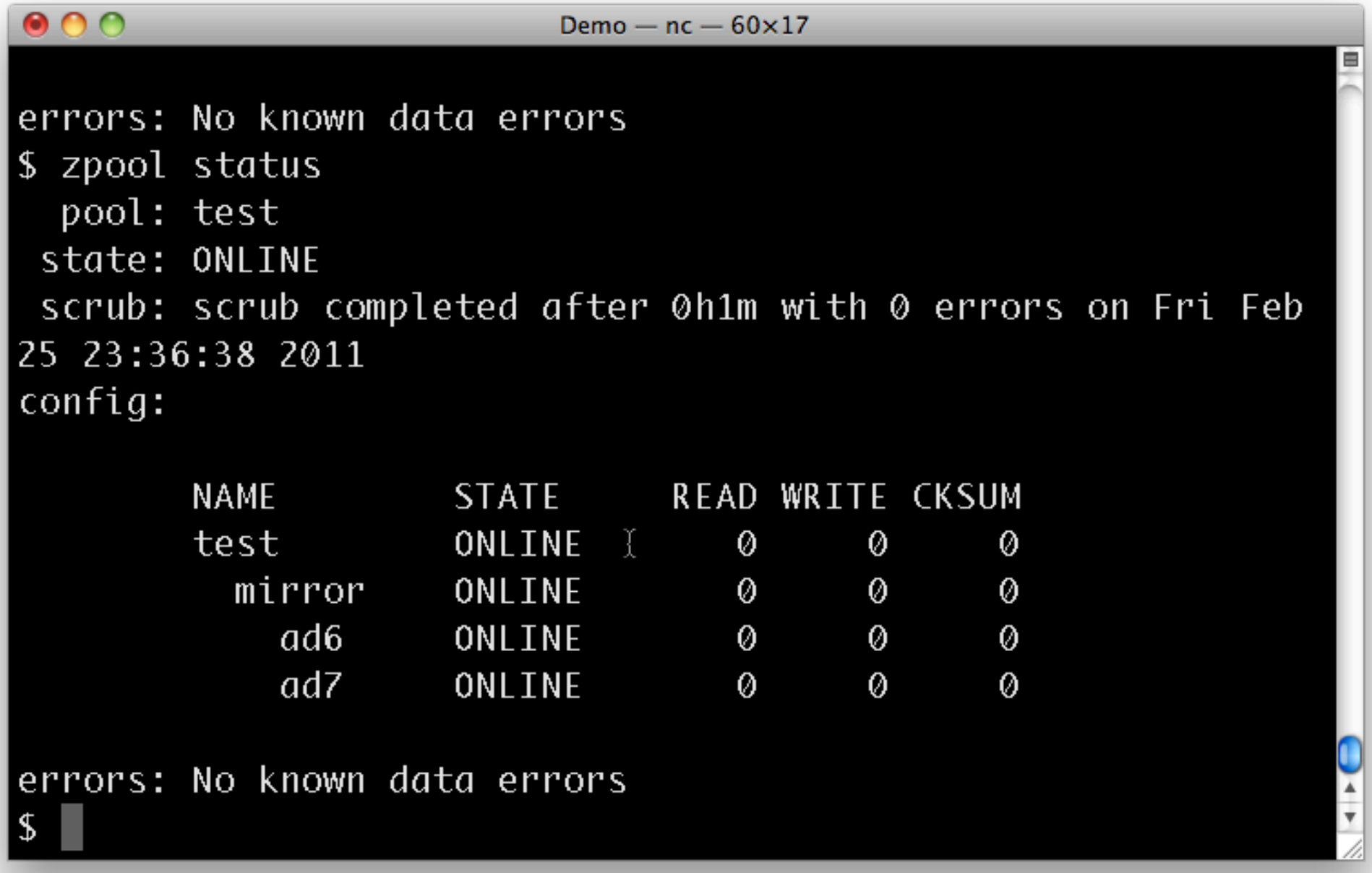
Once we'd spotted the issue with ZFS we found that we could replicate it with UFS or even with no file system.



AEROSPACE

Monday, September 26, 2011

All we had to do was write to two disks at the same time and eventually we'd get data corruption.



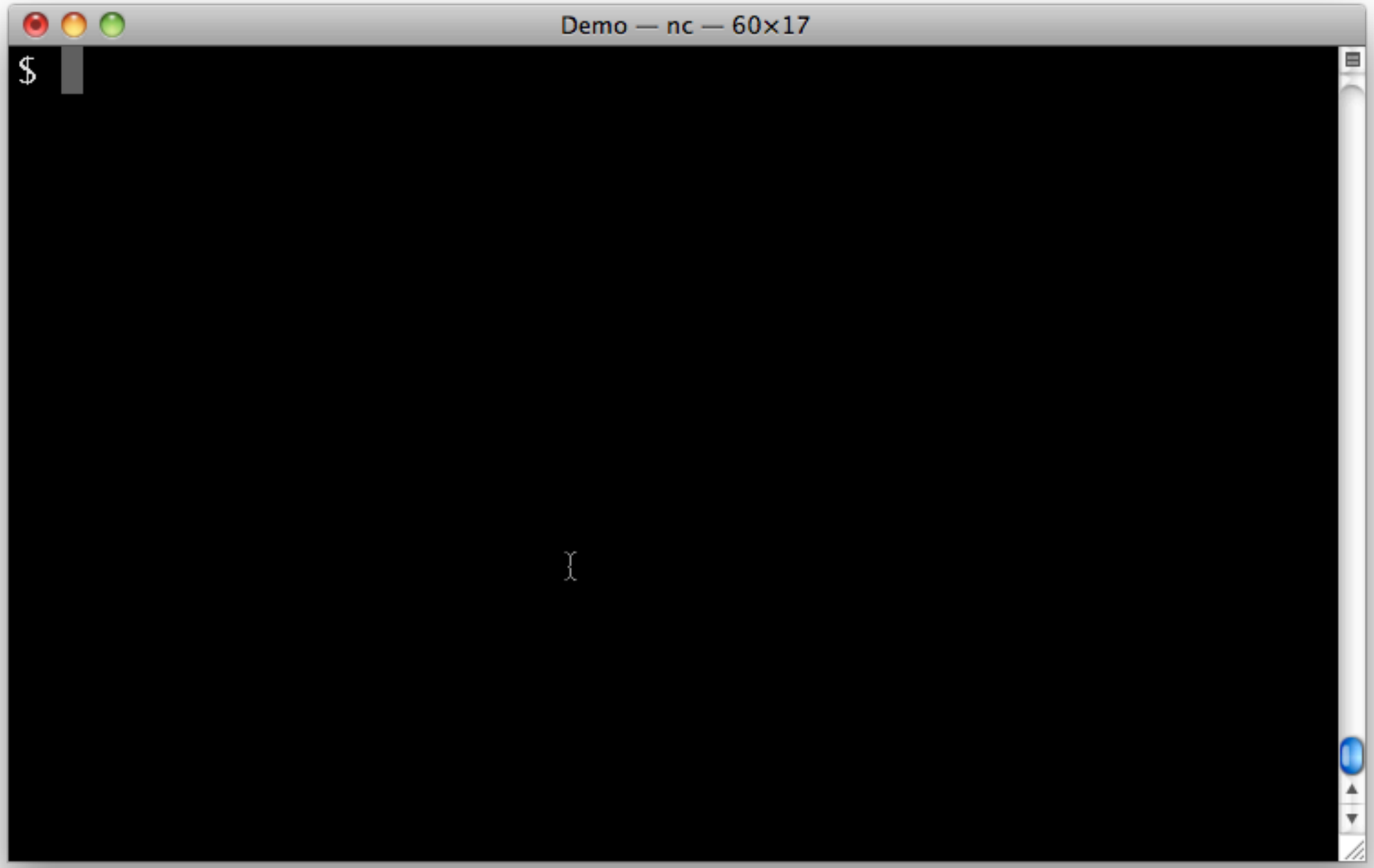
AEROSPACE

I wasn't able to install that card in this test box so the test completed without errors.
I've found errors that would otherwise have been hard to spot in multiple cases in the past so I can't recommend blasting some data onto a ZFS file system and then scrubbing the pool too much as a burn in test.

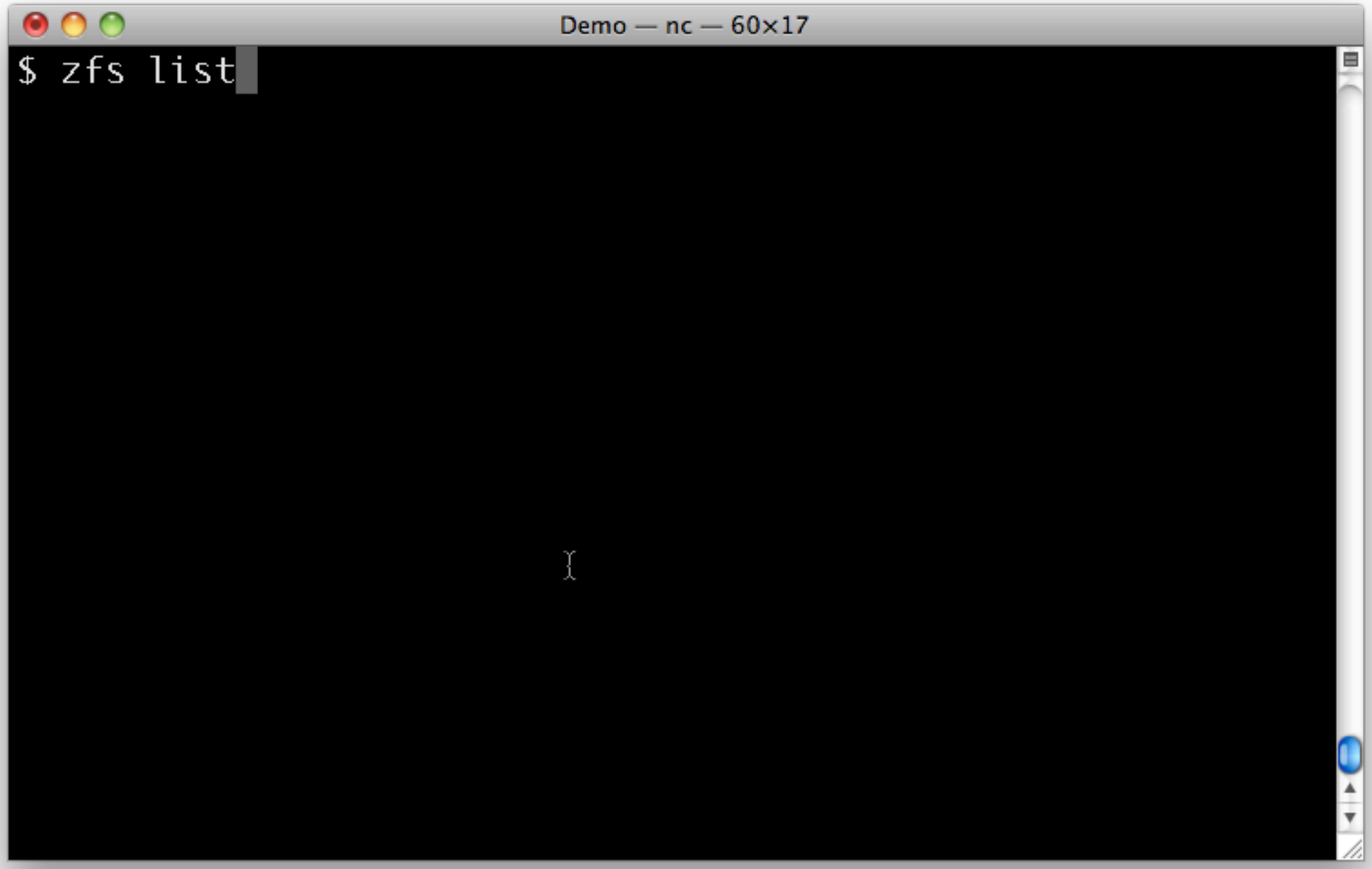
zfs(1)

AEROSPACE

Now that I've established some basics of pools, we can move on to the meat of the talk which is file system management. The zfs command manages file systems and volumes.

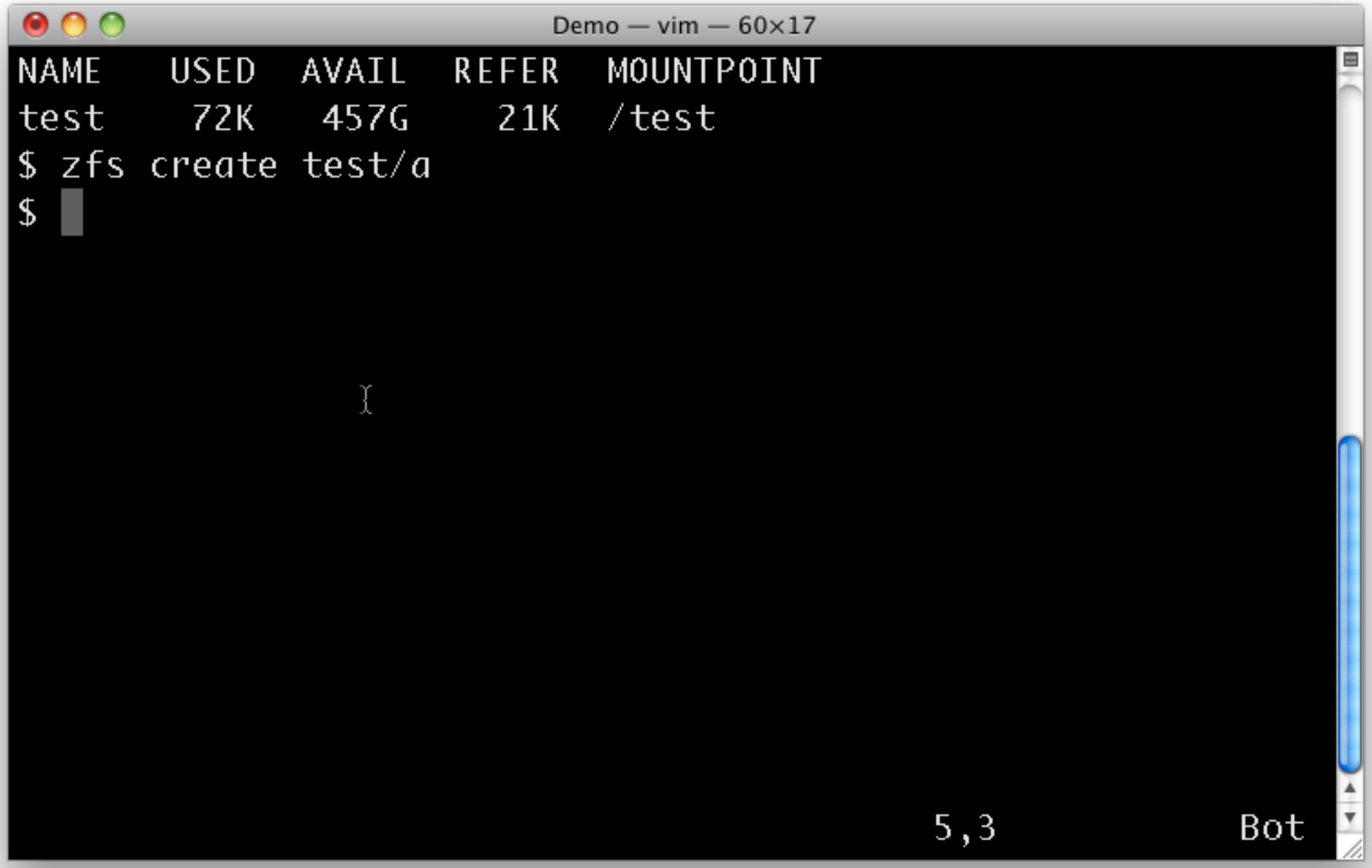


AEROSPACE

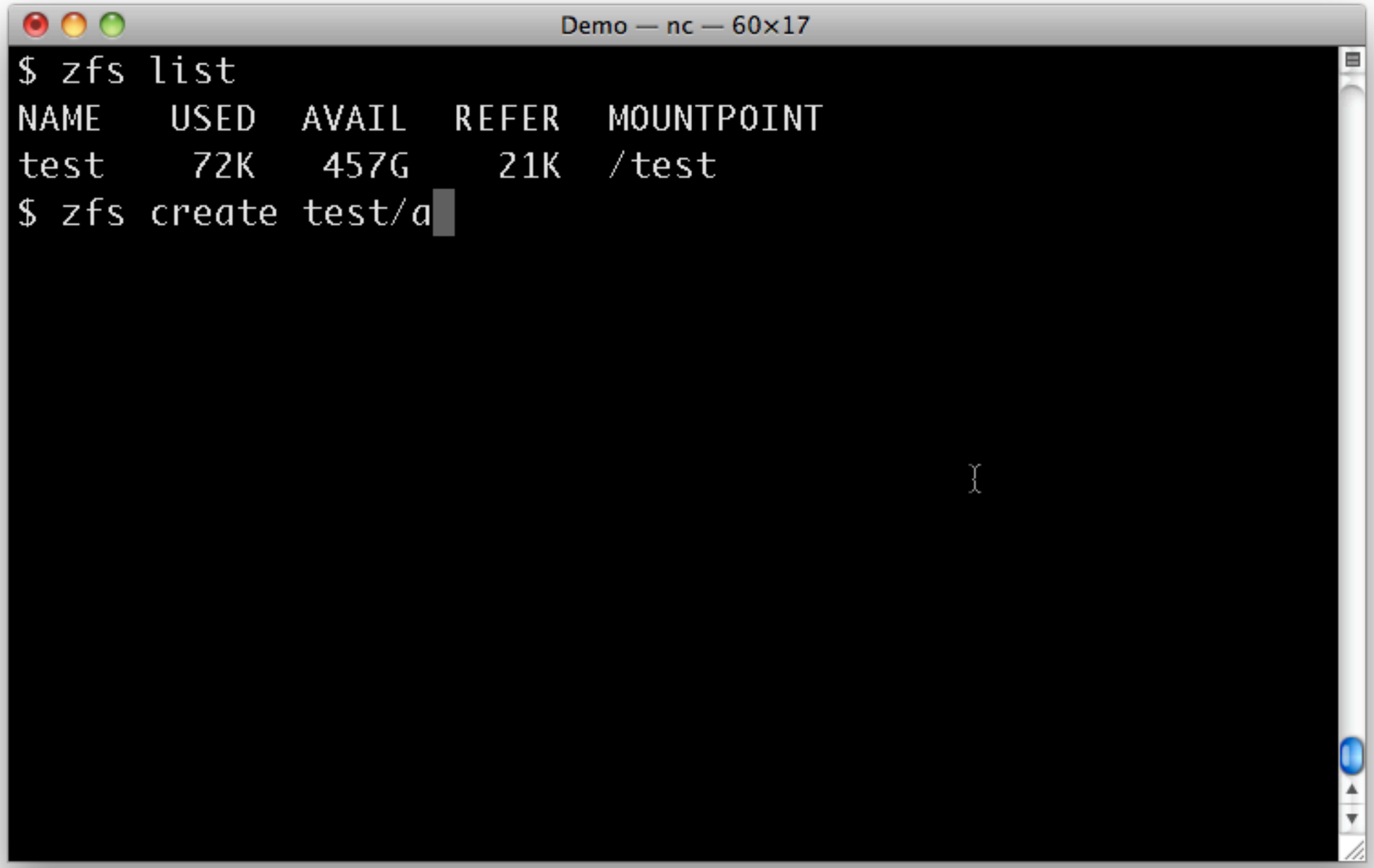


AEROSPACE

I've already shown you the zfs list command in passing. Now for some more interesting features.

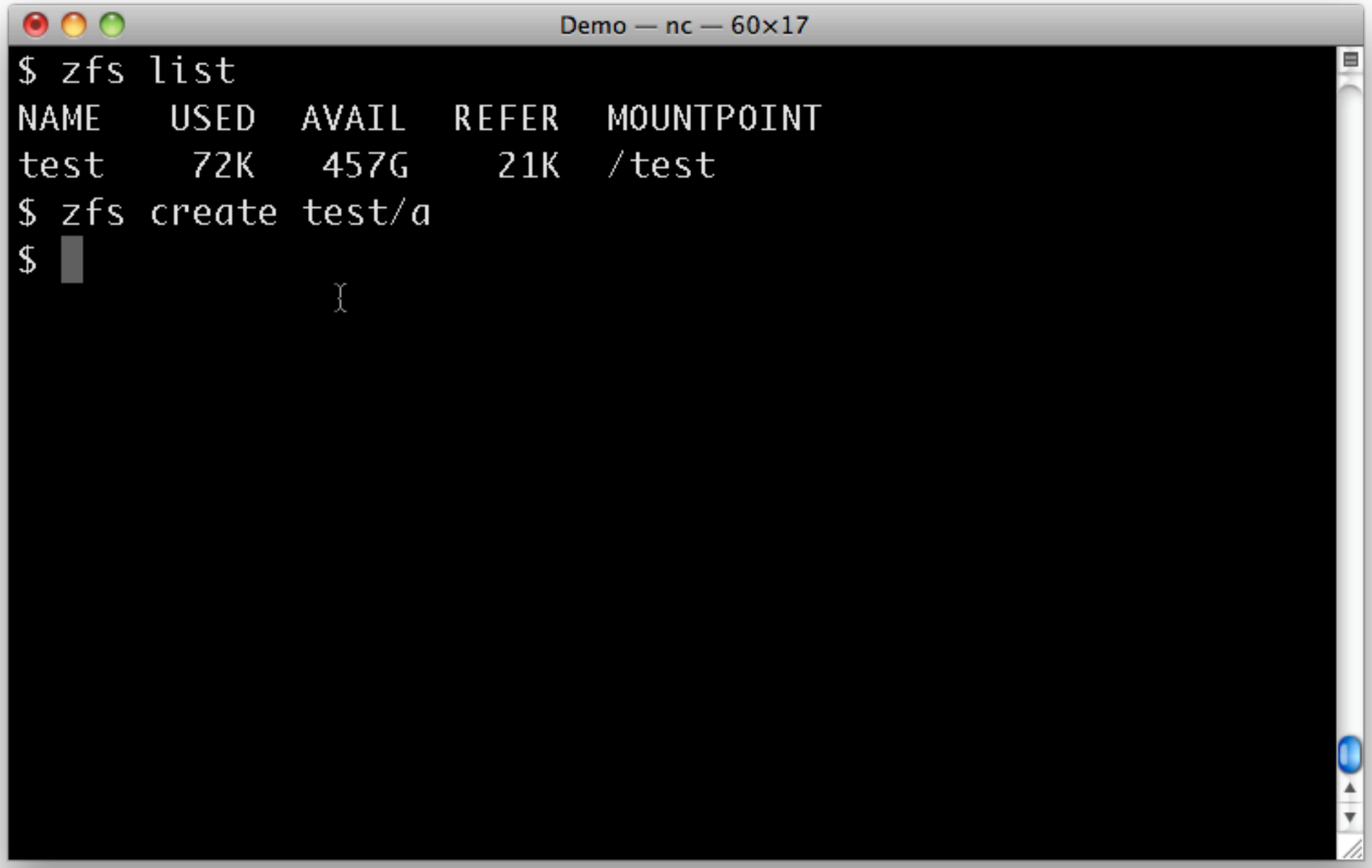

Demo — vim — 60×17
NAME USED AVAIL REFER MOUNTPOINT
test 72K 457G 21K /test
$ zfs create test/a
$
5,3 Bot

AEROSPACE



AEROSPACE

Let’s create some file systems


Demo — nc — 60×17
$ zfs list
NAME   USED AVAIL REFER MOUNTPOINT
test   72K   457G   21K /test
$ zfs create test/a
$

AEROSPACE

Demo — nc — 60×17

```
$ zfs list
NAME   USED  AVAIL  REFER  MOUNTPOINT
test    72K   457G    21K  /test
$ zfs create test/a
$ zfs create test/b
```

AEROSPACE

Demo — nc — 60×17

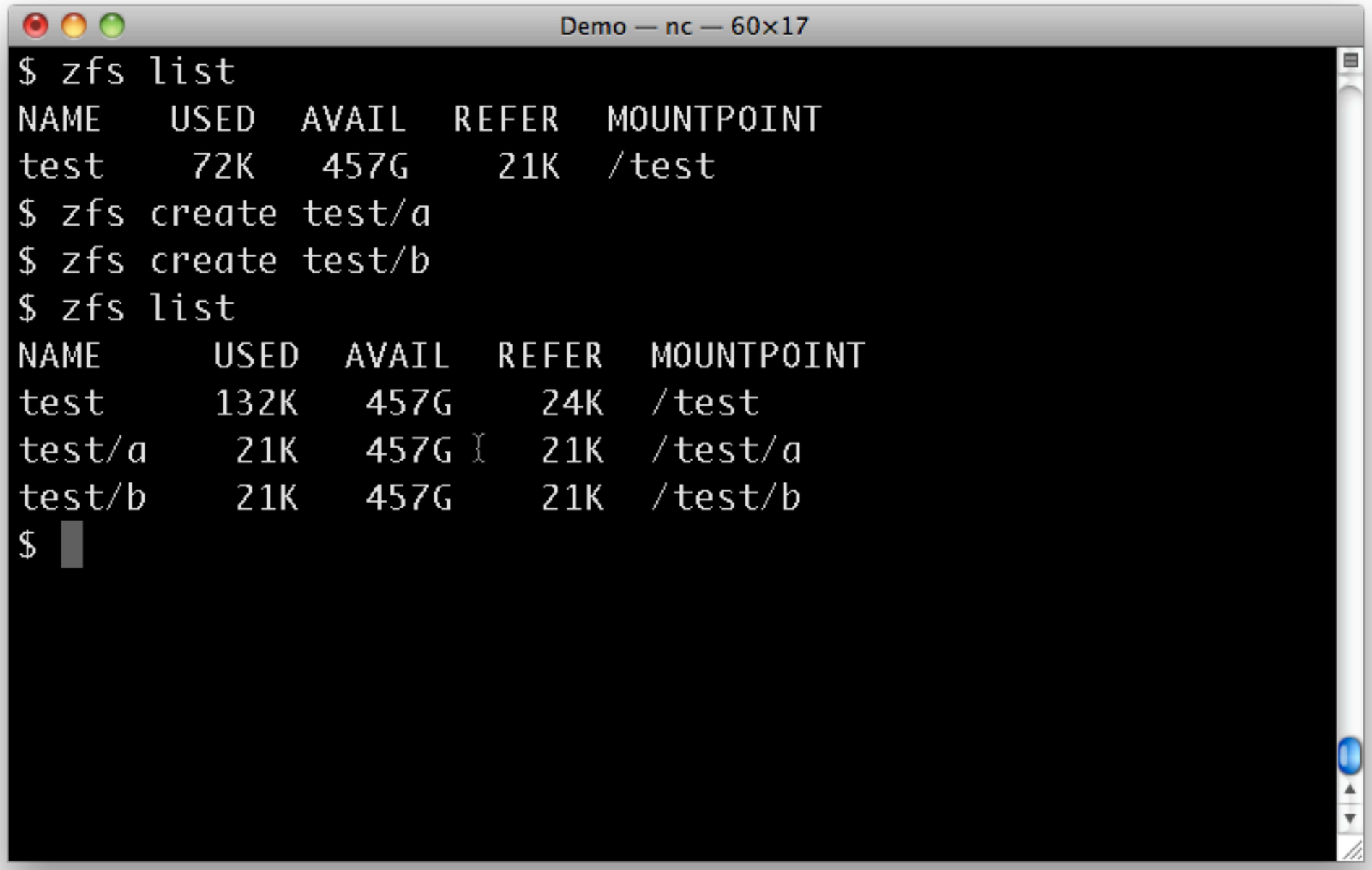
```
$ zfs list
NAME   USED  AVAIL  REFER  MOUNTPOINT
test    72K   457G    21K  /test
$ zfs create test/a
$ zfs create test/b
$
```

AEROSPACE

Demo — nc — 60×17

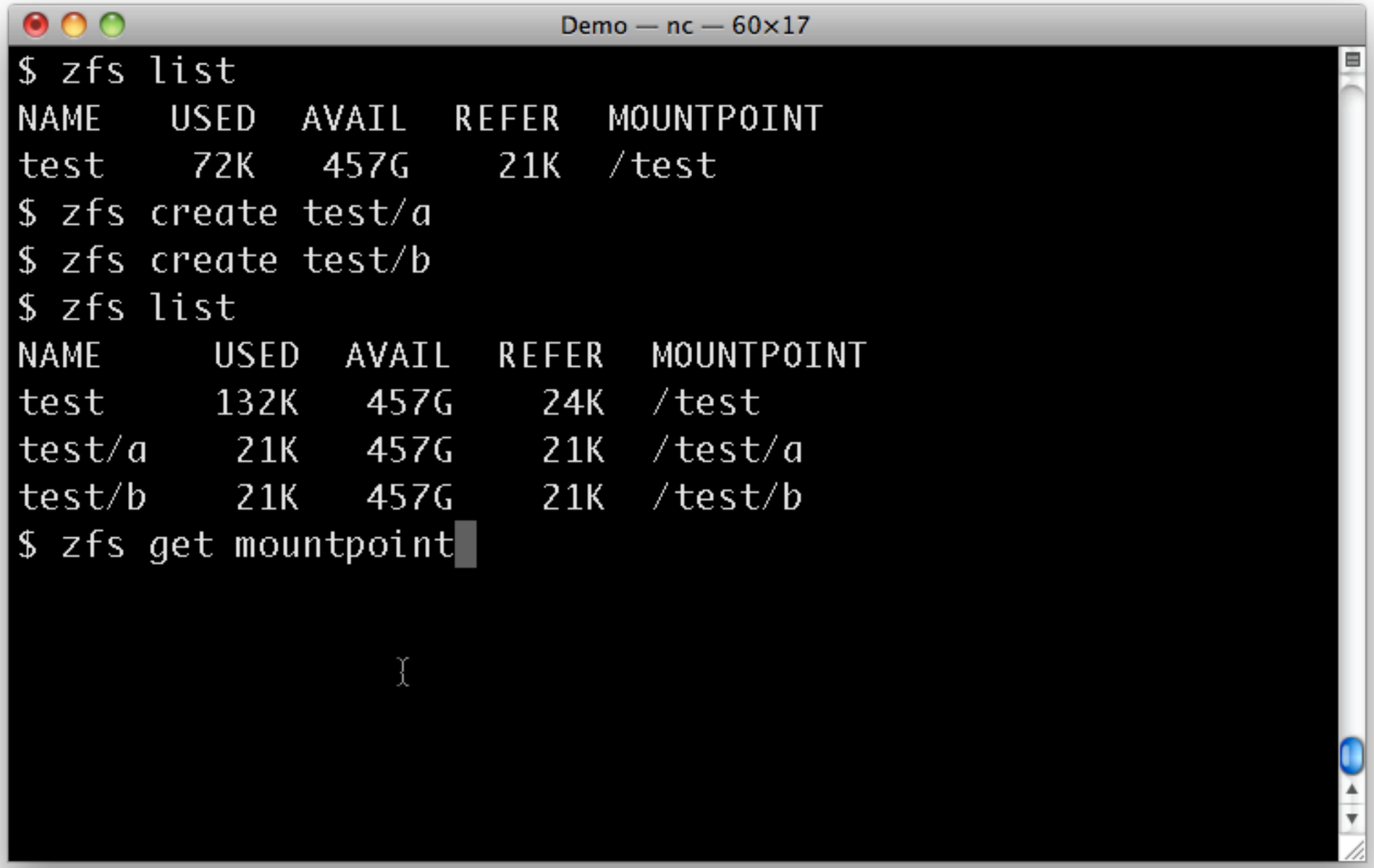
```
$ zfs list
NAME   USED  AVAIL  REFER  MOUNTPOINT
test   72K   457G    21K  /test
$ zfs create test/a
$ zfs create test/b
$ zfs list
```

AEROSPACE



AEROSPACE

By default a pool is mounted at /<poolname> and file systems are mounted relative to their parent.



AEROSPACE

Individual mount points are controlled by the mountpoint property.

Demo — nc — 60×17

```
$ zfs list
NAME   USED  AVAIL  REFER  MOUNTPOINT
test    72K   457G    21K  /test
$ zfs create test/a
$ zfs create test/b
$ zfs list
NAME      USED  AVAIL  REFER  MOUNTPOINT
test      132K   457G    24K  /test
test/a     21K   457G    21K  /test/a
test/b     21K   457G    21K  /test/b
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /test       default
test/a  mountpoint  /test/a     default
test/b  mountpoint  /test/b     default
$
```

AEROSPACE

Here we see that all mounts have the default value.

Demo — nc — 60×17

```
$ zfs list
NAME   USED  AVAIL  REFER  MOUNTPOINT
test    72K   457G    21K  /test
$ zfs create test/a
$ zfs create test/b
$ zfs list
NAME      USED  AVAIL  REFER  MOUNTPOINT
test      132K   457G    24K  /test
test/a     21K   457G    21K  /test/a
test/b     21K   457G    21K  /test/b
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /test       default
test/a  mountpoint  /test/a     default
test/b  mountpoint  /test/b     default
$ zfs set mountpoint=/data test
```

AEROSPACE

We can change the mount point by setting the property.

Demo — nc — 60×17

```
$ zfs list
NAME   USED  AVAIL  REFER  MOUNTPOINT
test    72K   457G    21K  /test
$ zfs create test/a
$ zfs create test/b
$ zfs list
NAME      USED  AVAIL  REFER  MOUNTPOINT
test      132K   457G    24K  /test
test/a     21K   457G    21K  /test/a
test/b     21K   457G    21K  /test/b
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /test       default
test/a  mountpoint  /test/a     default
test/b  mountpoint  /test/b     default
$ zfs set mountpoint=/data test
$
```

AEROSPACE

Demo — nc — 60×17

```
$ zfs list
NAME   USED  AVAIL  REFER  MOUNTPOINT
test    72K   457G    21K  /test
$ zfs create test/a
$ zfs create test/b
$ zfs list
NAME      USED  AVAIL  REFER  MOUNTPOINT
test      132K   457G    24K  /test
test/a     21K   457G    21K  /test/a
test/b     21K   457G    21K  /test/b
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /test       default
test/a  mountpoint  /test/a     default
test/b  mountpoint  /test/b     default
$ zfs set mountpoint=/data test
$ zfs get mountpoint
```

AEROSPACE

Demo — nc — 60×17

```
$ zfs list
NAME      USED  AVAIL  REFER  MOUNTPOINT
test      132K   457G    24K  /test
test/a     21K   457G    21K  /test/a
test/b     21K   457G    21K  /test/b
$ zfs get mountpoint
NAME     PROPERTY    VALUE       SOURCE
test     mountpoint  /test       default
test/a   mountpoint  /test/a     default
test/b   mountpoint  /test/b     default
$ zfs set mountpoint=/data test
$ zfs get mountpoint
NAME     PROPERTY    VALUE       SOURCE
test     mountpoint  /data       local
test/a   mountpoint  /data/a     inherited from test
test/b   mountpoint  /data/b     inherited from test
$
```

AEROSPACE

Monday, September 26, 2011

Now all of the file systems are mounted under /data and you can see how the property on test has flowed down to it's children.

Demo — nc — 60×17

```
$ zfs list
NAME      USED  AVAIL  REFER  MOUNTPOINT
test      132K   457G    24K  /test
test/a     21K   457G    21K  /test/a
test/b     21K   457G    21K  /test/b
$ zfs get mountpoint
NAME     PROPERTY    VALUE       SOURCE
test     mountpoint  /test       default
test/a   mountpoint  /test/a     default
test/b   mountpoint  /test/b     default
$ zfs set mountpoint=/data test
$ zfs get mountpoint
NAME     PROPERTY    VALUE       SOURCE
test     mountpoint  /data       local
test/a   mountpoint  /data/a     inherited from test
test/b   mountpoint  /data/b     inherited from test
$ zfs set mountpoint=/a test/a
```

AEROSPACE

Demo — nc — 60×17

```
NAME      USED  AVAIL  REFER  MOUNTPOINT
test      132K   457G    24K  /test
test/a     21K   457G    21K  /test/a
test/b     21K   457G    21K  /test/b
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /test       default
test/a  mountpoint  /test/a     default
test/b  mountpoint  /test/b     default
$ zfs set mountpoint=/data test
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /data       local
test/a  mountpoint  /data/a     inherited from test
test/b  mountpoint  /data/b     inherited from test
$ zfs set mountpoint=/a test/a
$
```

AEROSPACE

Demo — nc — 60×17

```
NAME      USED  AVAIL  REFER  MOUNTPOINT
test      132K   457G    24K  /test
test/a     21K   457G    21K  /test/a
test/b     21K   457G    21K  /test/b
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /test       default
test/a  mountpoint  /test/a     default
test/b  mountpoint  /test/b     default
$ zfs set mountpoint=/data test
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /data       local
test/a  mountpoint  /data/a     inherited from test
test/b  mountpoint  /data/b     inherited from test
$ zfs set mountpoint=/a test/a
$ zfs get mountpoint
```

AEROSPACE

Demo — nc — 60×17

```
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /test       default
test/a  mountpoint  /test/a     default
test/b  mountpoint  /test/b     default
$ zfs set mountpoint=/data test
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /data       local
test/a  mountpoint  /data/a     inherited from test
test/b  mountpoint  /data/b     inherited from test
$ zfs set mountpoint=/a test/a
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /data       local
test/a  mountpoint  /a          local
test/b  mountpoint  /data/b     inherited from test
$
```

AEROSPACE

Now, if we change our mind and want to revert those changes.

Demo — nc — 60×17

```
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /test       default
test/a  mountpoint  /test/a     default
test/b  mountpoint  /test/b     default
$ zfs set mountpoint=/data test
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /data       local
test/a  mountpoint  /data/a     inherited from test
test/b  mountpoint  /data/b     inherited from test
$ zfs set mountpoint=/a test/a
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /data       local
test/a  mountpoint  /a          local
test/b  mountpoint  /data/b     inherited from test
$ zfs inherit mountpoint test
```

AEROSPACE

Monday, September 26, 2011

We use the inherit command. I find it a bit unintuitive that the opposite of “set” in “inherit” and not “unset”, but to makes some sense.

Demo — nc — 60×17

```
test    mountpoint  /test       default
test/a  mountpoint  /test/a     default
test/b  mountpoint  /test/b     default
$ zfs set mountpoint=/data test
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /data       local
test/a  mountpoint  /data/a     inherited from test
test/b  mountpoint  /data/b     inherited from test
$ zfs set mountpoint=/a test/a
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /data       local
test/a  mountpoint  /a          local
test/b  mountpoint  /data/b     inherited from test
$ zfs inherit mountpoint test
$
```

AEROSPACE

Demo — nc — 60×17

```
test    mountpoint  /test       default
test/a  mountpoint  /test/a     default
test/b  mountpoint  /test/b     default
$ zfs set mountpoint=/data test
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /data       local
test/a  mountpoint  /data/a     inherited from test
test/b  mountpoint  /data/b     inherited from test
$ zfs set mountpoint=/a test/a
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /data       local
test/a  mountpoint  /a          local
test/b  mountpoint  /data/b     inherited from test
$ zfs inherit mountpoint test
$ zfs get mountpoint
```

AEROSPACE

Demo — nc — 60×17

```
NAME     PROPERTY    VALUE       SOURCE
test     mountpoint  /data       local
test/a   mountpoint  /data/a     inherited from test
test/b   mountpoint  /data/b     inherited from test
$ zfs set mountpoint=/a test/a
$ zfs get mountpoint
NAME     PROPERTY    VALUE       SOURCE
test     mountpoint  /data       local
test/a   mountpoint  /a          local
test/b   mountpoint  /data/b     inherited from test
$ zfs inherit mountpoint test
$ zfs get mountpoint
NAME     PROPERTY    VALUE       SOURCE
test     mountpoint  /test       default
test/a   mountpoint  /a          local
test/b   mountpoint  /test/b     default
$
```

AEROSPACE

This shows the action of the first inherit command. Note that test/a is still mounted at /a.

Demo — nc — 60×17

```
NAME     PROPERTY    VALUE         SOURCE
test     mountpoint  /data         local
test/a   mountpoint  /data/a       inherited from test
test/b   mountpoint  /data/b       inherited from test
$ zfs set mountpoint=/a test/a
$ zfs get mountpoint
NAME     PROPERTY    VALUE         SOURCE
test     mountpoint  /data         local
test/a   mountpoint  /a            local
test/b   mountpoint  /data/b       inherited from test
$ zfs inherit mountpoint test
$ zfs get mountpoint
NAME     PROPERTY    VALUE         SOURCE
test     mountpoint  /test         default
test/a   mountpoint  /a            local
test/b   mountpoint  /test/b       default
$ zfs inherit mountpoint test/a
```

AEROSPACE

One more inherit...

Demo — nc — 60×17

```
test    mountpoint  /data       local
test/a  mountpoint  /data/a     inherited from test
test/b  mountpoint  /data/b     inherited from test
$ zfs set mountpoint=/a test/a
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /data       local
test/a  mountpoint  /a          local
test/b  mountpoint  /data/b     inherited from test
$ zfs inherit mountpoint test
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /test       default
test/a  mountpoint  /a          local
test/b  mountpoint  /test/b     default
$ zfs inherit mountpoint test/a
$
```

AEROSPACE

Demo — nc — 60×17

```
test    mountpoint  /data       local
test/a  mountpoint  /data/a     inherited from test
test/b  mountpoint  /data/b     inherited from test
$ zfs set mountpoint=/a test/a
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /data       local
test/a  mountpoint  /a          local
test/b  mountpoint  /data/b     inherited from test
$ zfs inherit mountpoint test
$ zfs get mountpoint
NAME    PROPERTY    VALUE       SOURCE
test    mountpoint  /test       default
test/a  mountpoint  /a          local
test/b  mountpoint  /test/b     default
$ zfs inherit mountpoint test/a
$ zfs get mountpoint
```

AEROSPACE

Demo — nc — 60×17

```
NAME     PROPERTY    VALUE         SOURCE
test     mountpoint  /data         local
test/a   mountpoint  /a            local
test/b   mountpoint  /data/b       inherited from test
$ zfs inherit mountpoint test
$ zfs get mountpoint
NAME     PROPERTY    VALUE         SOURCE
test     mountpoint  /test         default
test/a   mountpoint  /a            local
test/b   mountpoint  /test/b       default
$ zfs inherit mountpoint test/a
$ zfs get mountpoint
NAME     PROPERTY    VALUE         SOURCE
test     mountpoint  /test         default
test/a   mountpoint  /test/a       default
test/b   mountpoint  /test/b       default
$
```

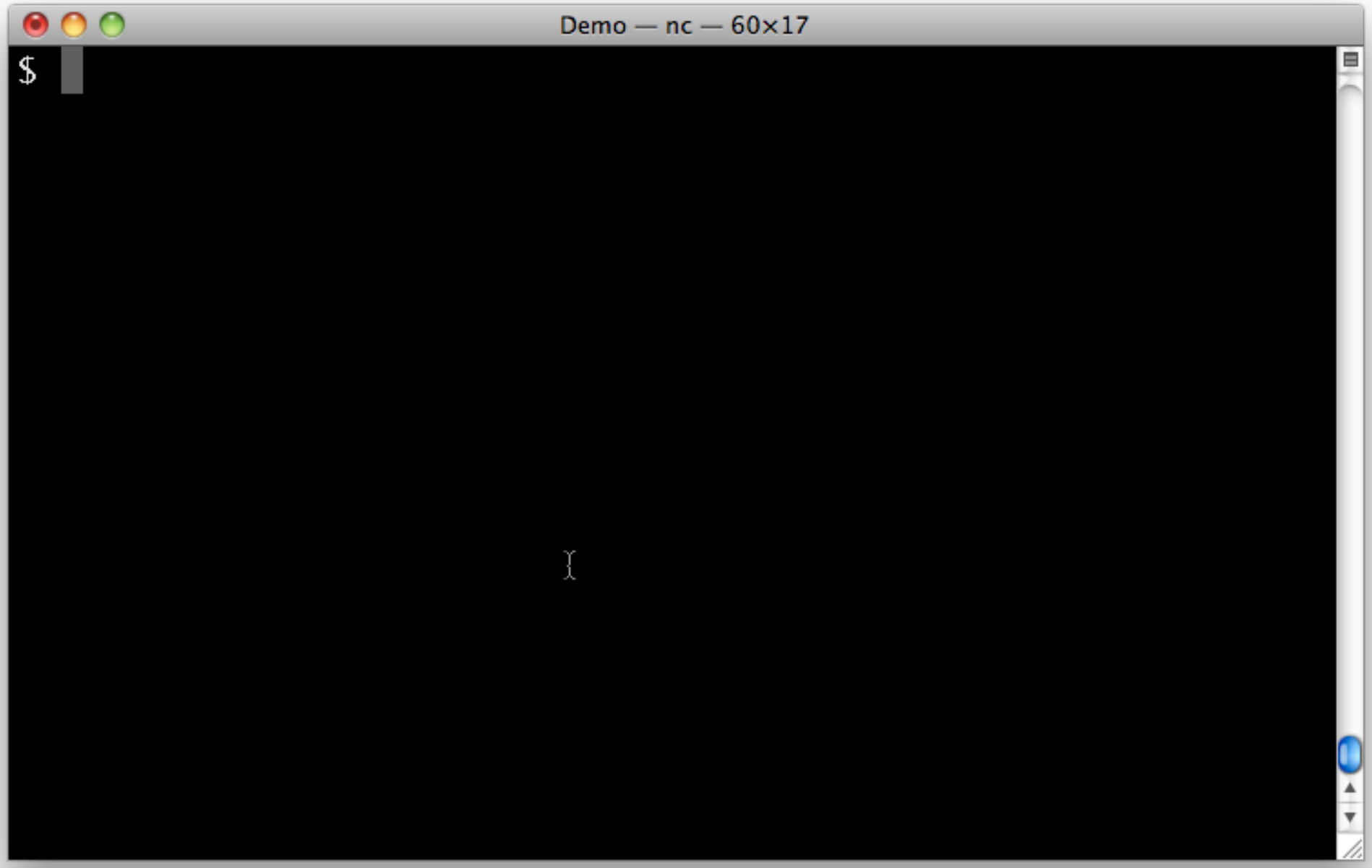
AEROSPACE

And we're back where we were.

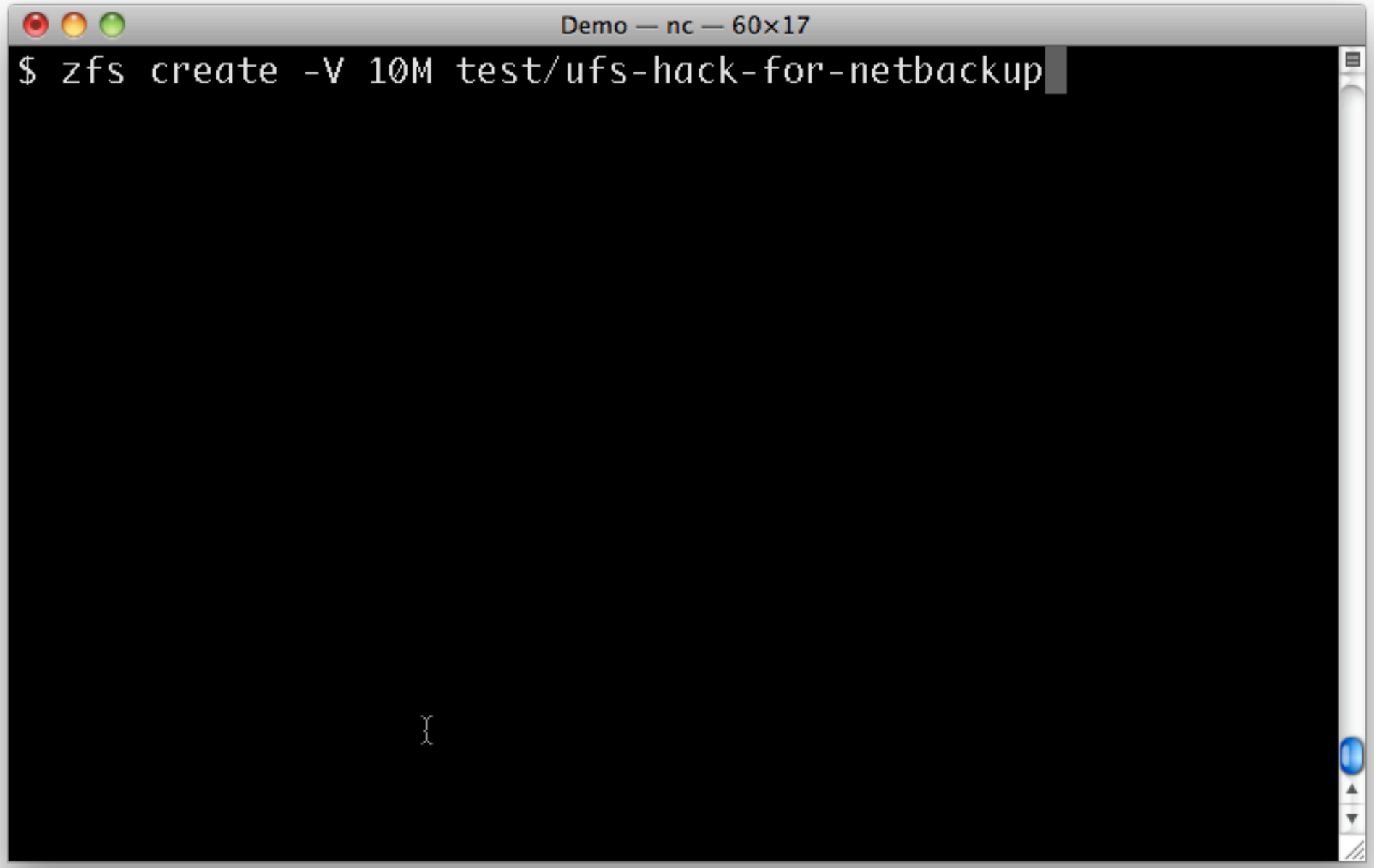
Volumes

AEROSPACE

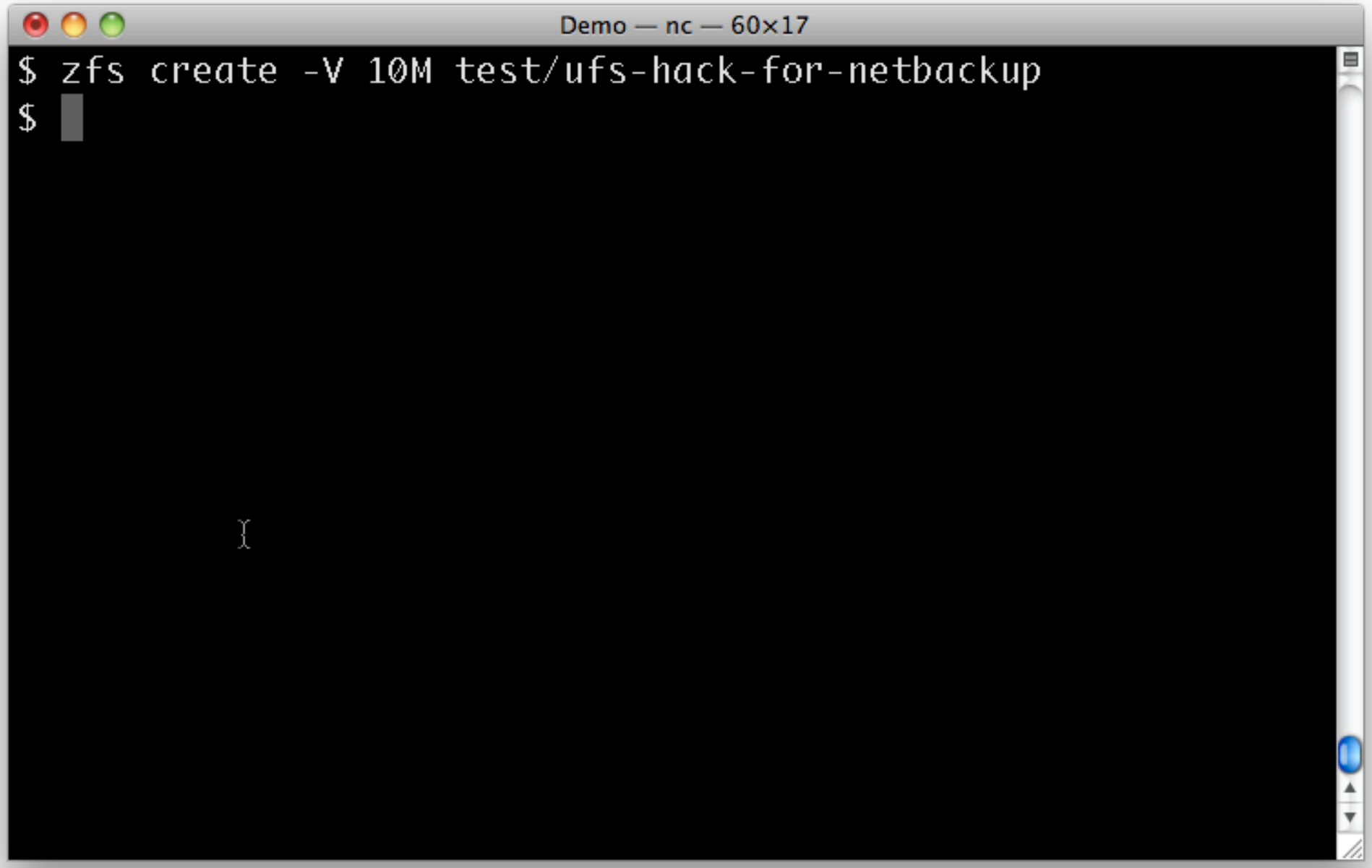
In addition to file systems, zfs allows the creation of volumes which are simply block devices. On FreeBSD they are just GEOMs.



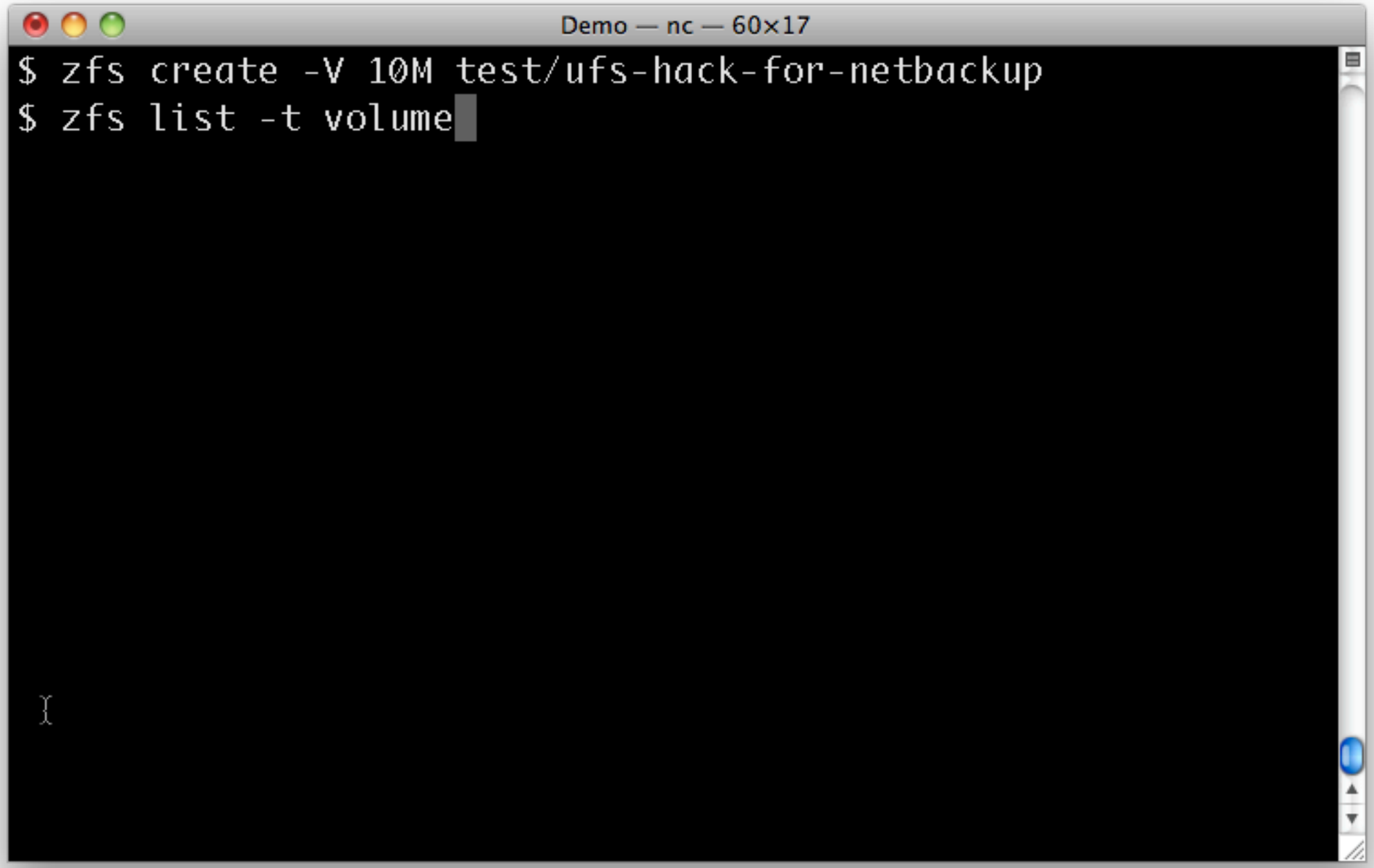
AEROSPACE



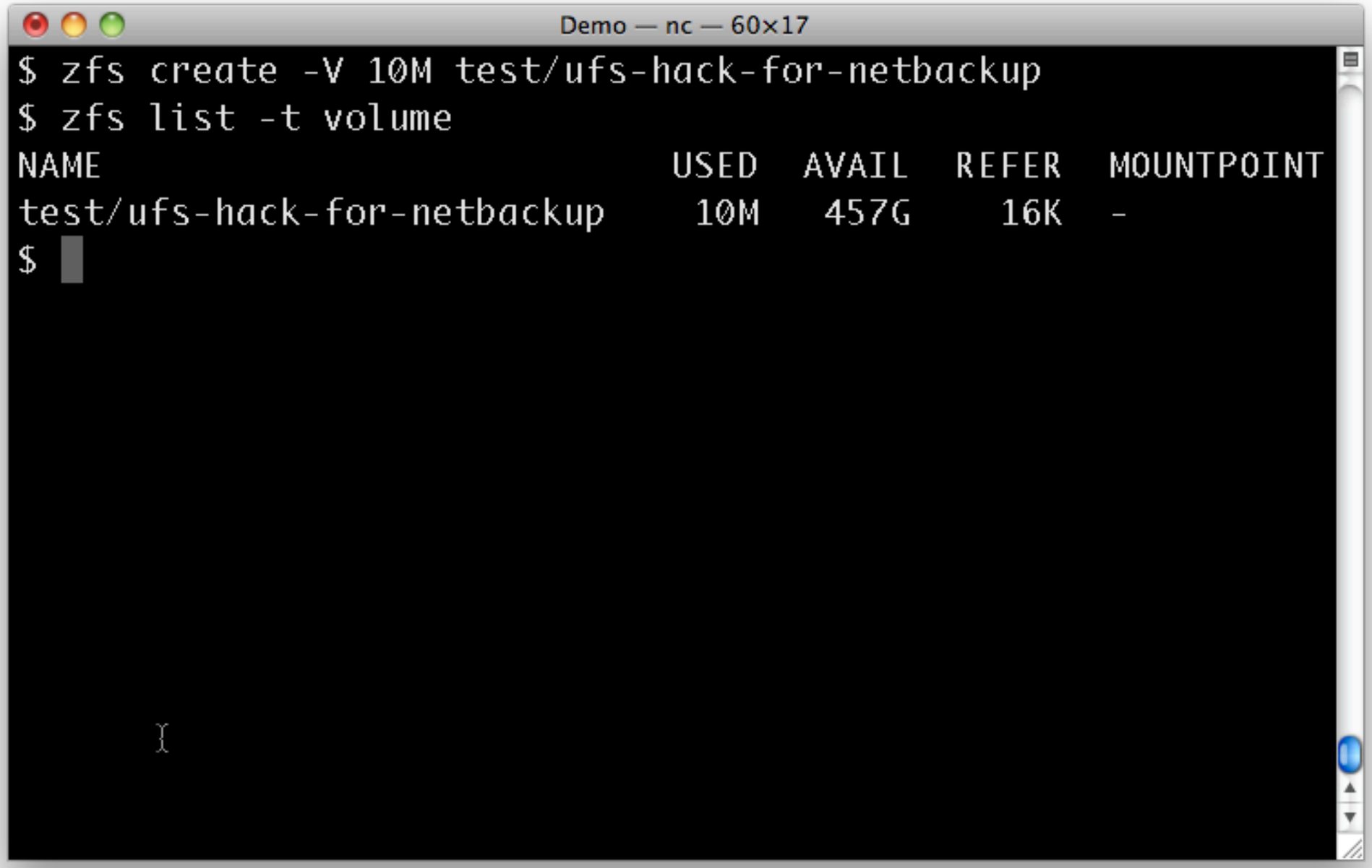
AEROSPACE



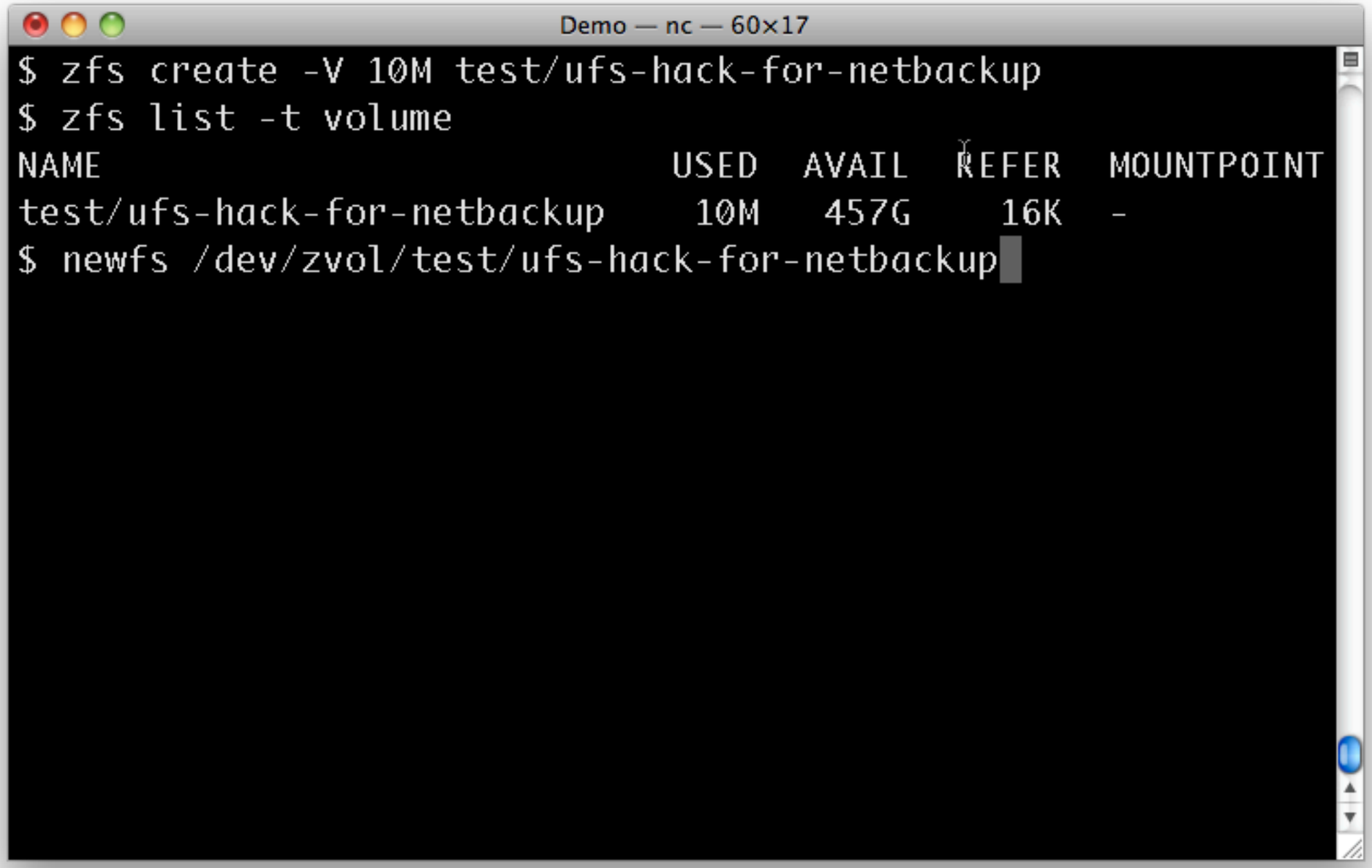
AEROSPACE


Demo — nc — 60×17
$ zfs create -V 10M test/ufs-hack-for-netbackup
$ zfs list -t volume

AEROSPACE


Demo — nc — 60×17
$ zfs create -V 10M test/ufs-hack-for-netbackup
$ zfs list -t volume
NAME   USED AVAIL REFER MOUNTPOINT
test/ufs-hack-for-netbackup   10M   457G   16K -
$

AEROSPACE


Demo — nc — 60×17
$ zfs create -V 10M test/ufs-hack-for-netbackup
$ zfs list -t volume
NAME   USED AVAIL REFER MOUNTPOINT
test/ufs-hack-for-netbackup   10M   457G   16K -
$ newfs /dev/zvol/test/ufs-hack-for-netbackup

AEROSPACE

Demo — nc — 60×17

```
$ zfs create -V 10M test/ufs-hack-for-netbackup
$ zfs list -t volume
NAME                              USED  AVAIL  REFER  MOUNTPOINT
test/ufs-hack-for-netbackup        10M   457G    16K  -
$ newfs /dev/zvol/test/ufs-hack-for-netbackup
/dev/zvol/test/ufs-hack-for-netbackup: 10.0MB (20480 sectors
) block size 16384, fragment size 2048
        using 4 cylinder groups of 2.52MB, 161 blks, 384 ino
des.
super-block backups (for fsck -b #) at:
 160, 5312, 10464, 15616
$
```

AEROSPACE

Demo — nc — 60×17

```
$ zfs create -V 10M test/ufs-hack-for-netbackup
$ zfs list -t volume
NAME                                 USED  AVAIL  REFER  MOUNTPOINT
test/ufs-hack-for-netbackup           10M   457G    16K  -
$ newfs /dev/zvol/test/ufs-hack-for-netbackup
/dev/zvol/test/ufs-hack-for-netbackup: 10.0MB (20480 sectors
) block size 16384, fragment size 2048
        using 4 cylinder groups of 2.52MB, 161 blks, 384 ino
des.
super-block backups (for fsck -b #) at:
 160, 5312, 10464, 15616
$ mkdir /var/ufs-hack-for-netbackup
```

AEROSPACE

Demo — nc — 60×17

```
$ zfs create -V 10M test/ufs-hack-for-netbackup
$ zfs list -t volume
NAME                              USED  AVAIL  REFER  MOUNTPOINT
test/ufs-hack-for-netbackup        10M   457G    16K  -
$ newfs /dev/zvol/test/ufs-hack-for-netbackup
/dev/zvol/test/ufs-hack-for-netbackup: 10.0MB (20480 sectors
) block size 16384, fragment size 2048
        using 4 cylinder groups of 2.52MB, 161 blks, 384 ino
des.
super-block backups (for fsck -b #) at:
 160, 5312, 10464, 15616
$ mkdir /var/ufs-hack-for-netbackup
$ mount /dev/zvol/test/ufs-hack-for-netbackup /var/ufs-hack-
for-netbackup
```

AEROSPACE

Demo — nc — 60×17

```
$ zfs create -V 10M test/ufs-hack-for-netbackup
$ zfs list -t volume
NAME                              USED  AVAIL  REFER  MOUNTPOINT
test/ufs-hack-for-netbackup        10M   457G    16K  -
$ newfs /dev/zvol/test/ufs-hack-for-netbackup
/dev/zvol/test/ufs-hack-for-netbackup: 10.0MB (20480 sectors
) block size 16384, fragment size 2048
        using 4 cylinder groups of 2.52MB, 161 blks, 384 ino
des.
super-block backups (for fsck -b #) at:
 160, 5312, 10464, 15616
$ mkdir /var/ufs-hack-for-netbackup
$ mount /dev/zvol/test/ufs-hack-for-netbackup /var/ufs-hack-
for-netbackup
$
```

AEROSPACE

Demo — nc — 60×17

```
$ zfs create -V 10M test/ufs-hack-for-netbackup
$ zfs list -t volume
NAME                              USED  AVAIL  REFER  MOUNTPOINT
test/ufs-hack-for-netbackup        10M   457G    16K  -
$ newfs /dev/zvol/test/ufs-hack-for-netbackup
/dev/zvol/test/ufs-hack-for-netbackup: 10.0MB (20480 sectors
) block size 16384, fragment size 2048
        using 4 cylinder groups of 2.52MB, 161 blks, 384 ino
des.
super-block backups (for fsck -b #) at:
 160, 5312, 10464, 15616
$ mkdir /var/ufs-hack-for-netbackup
$ mount /dev/zvol/test/ufs-hack-for-netbackup /var/ufs-hack-
for-netbackup
$ df /var/ufs-hack-for-netbackup
```

AEROSPACE

Demo — nc — 60×17

```
test/ufs-hack-for-netbackup    10M   457G    16K   -
$ newfs /dev/zvol/test/ufs-hack-for-netbackup
/dev/zvol/test/ufs-hack-for-netbackup: 10.0MB (20480 sectors
) block size 16384, fragment size 2048
        using 4 cylinder groups of 2.52MB, 161 blks, 384 ino
des.
super-block backups (for fsck -b #) at:
 160, 5312, 10464, 15616
$ mkdir /var/ufs-hack-for-netbackup
$ mount /dev/zvol/test/ufs-hack-for-netbackup /var/ufs-hack-
for-netbackup
$ df /var/ufs-hack-for-netbackup
Filesystem                                  512-blocks Used Avail
Capacity  Mounted on
/dev/zvol/test/ufs-hack-for-netbackup          19292    8 17744
    0%    /var/ufs-hack-for-netbackup
$
```

AEROSPACE

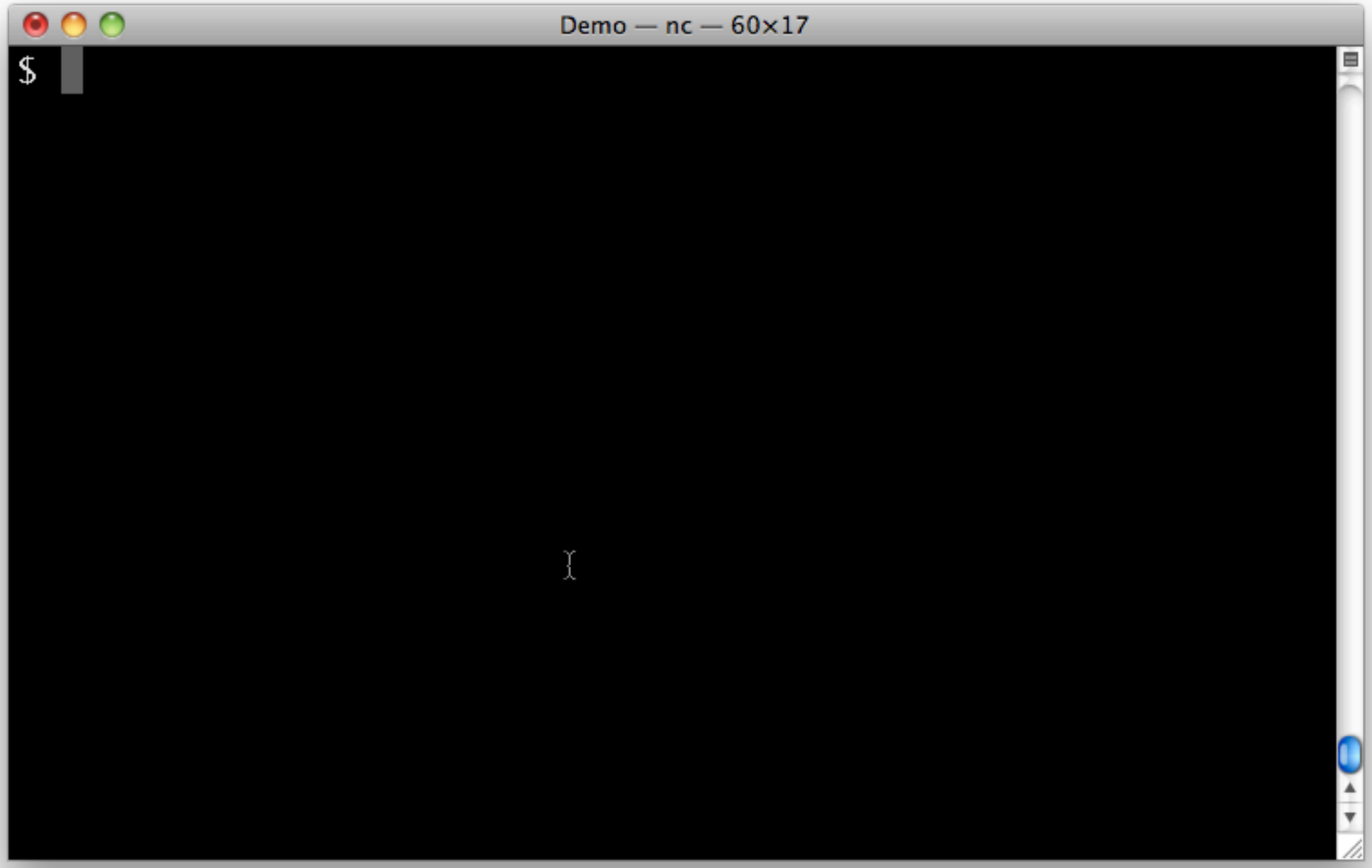
Monday, September 26, 2011

The name used there comes from our ZFS only systems. The Netbackup client crashes if there aren't any file systems it recognizes as local. Since it's a 5.x binary they didn't know about ZFS then so we need a local UFS file system. One note about this hack, we mount the file system late with the late option in /etc/fstab.

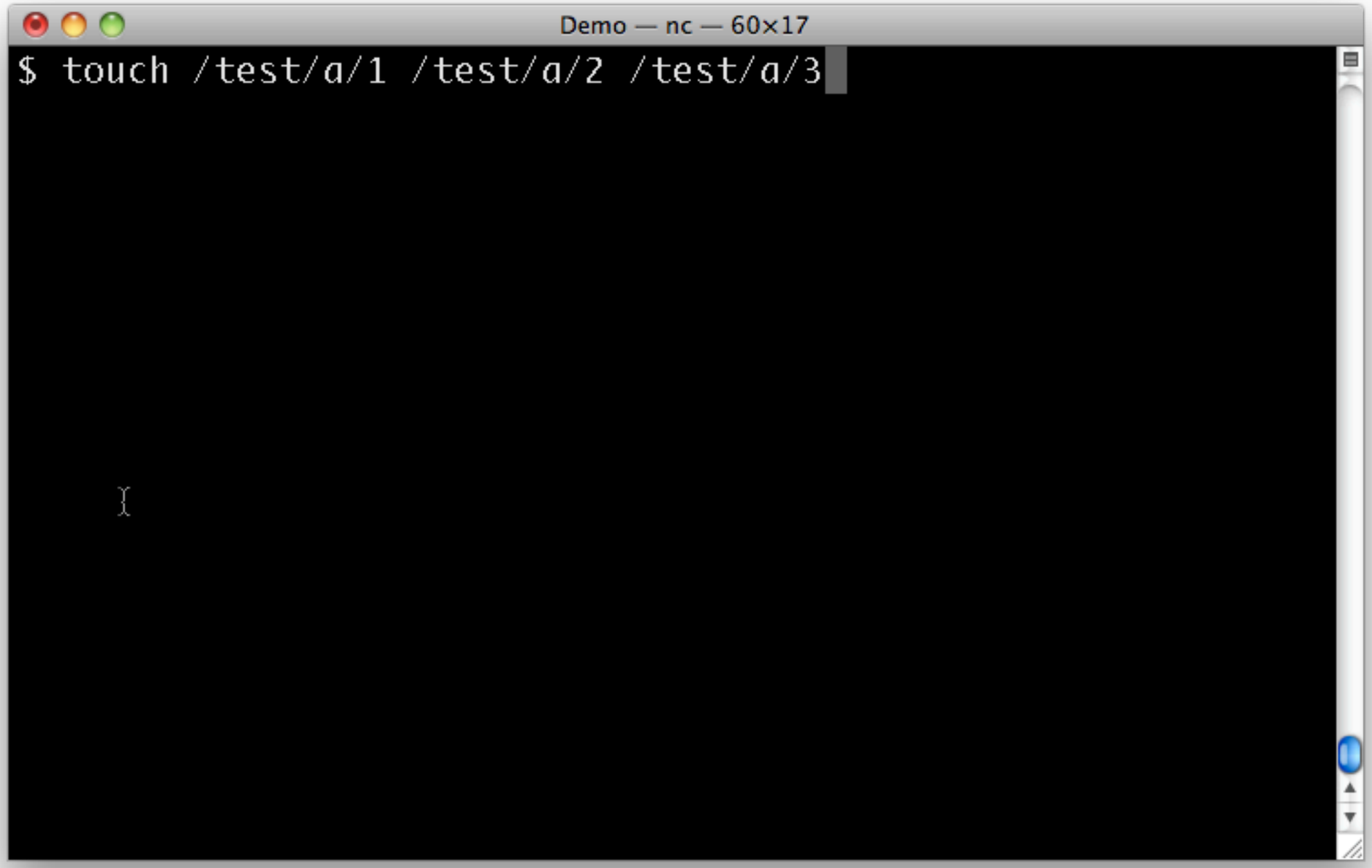
Snapshots

AEROSPACE

One of the more useful administrative features of ZFS is the ability to take snapshots. A number of features are built around them and many of the examples we will show later use them. Before we get to those, some simple snapshot basics.

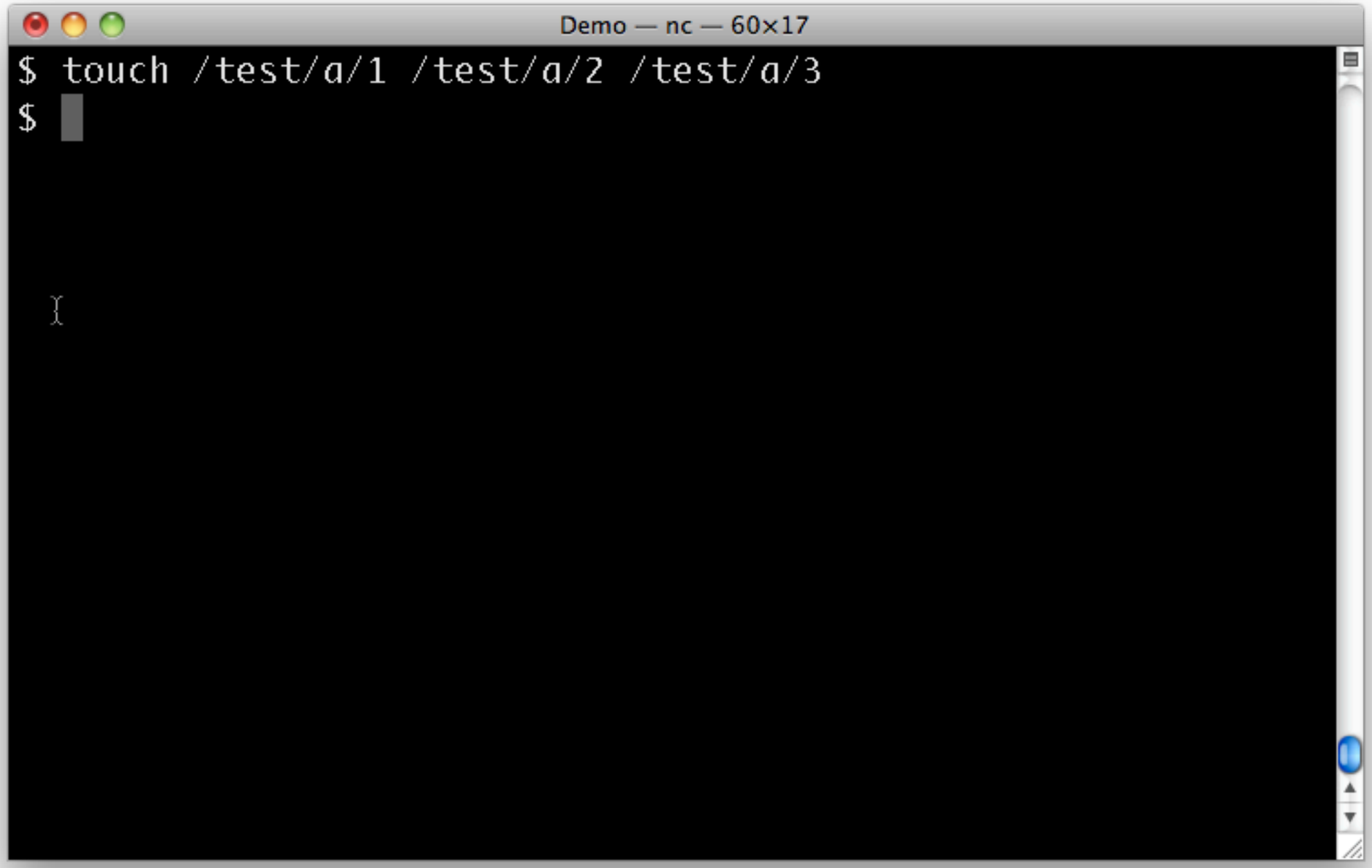


AEROSPACE

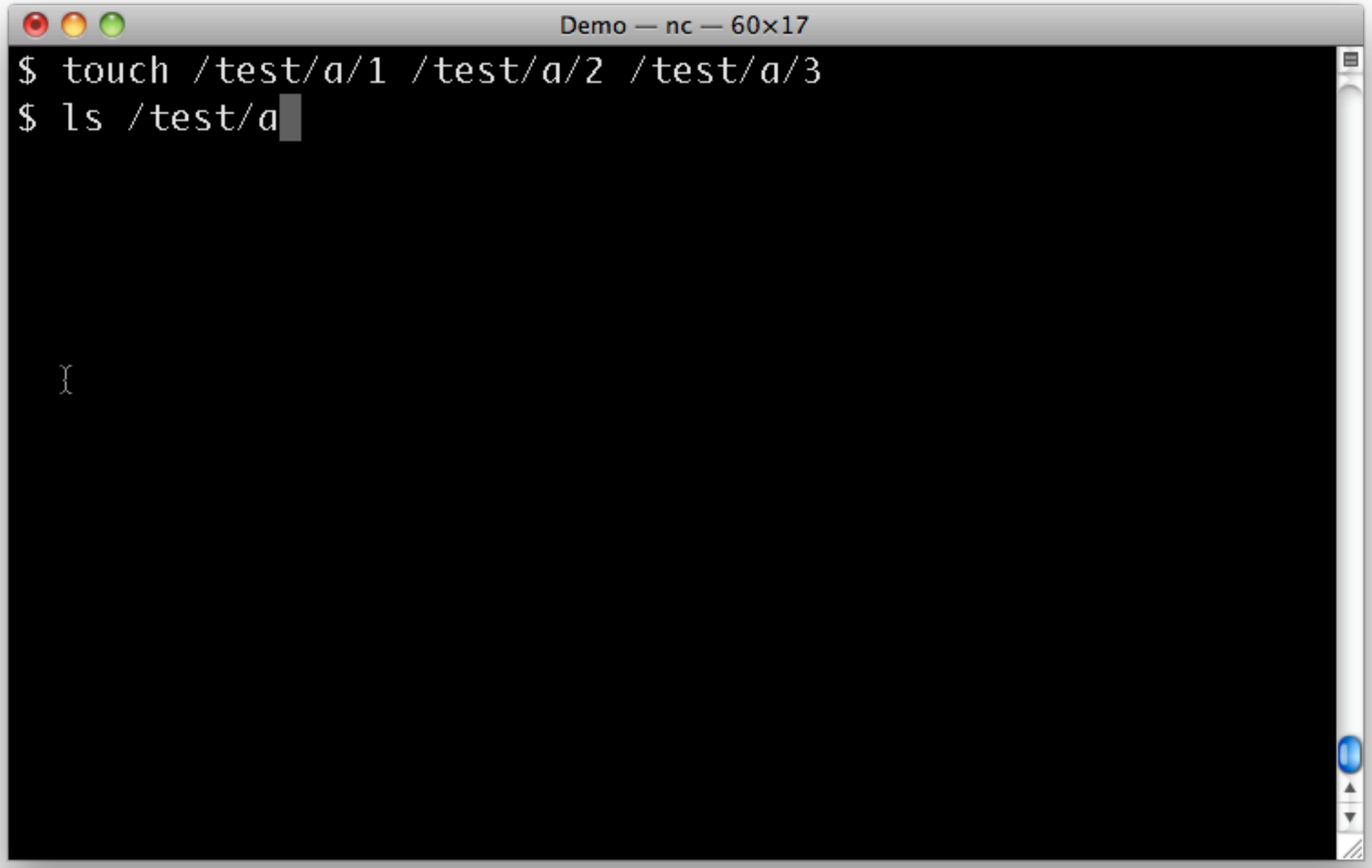


AEROSPACE

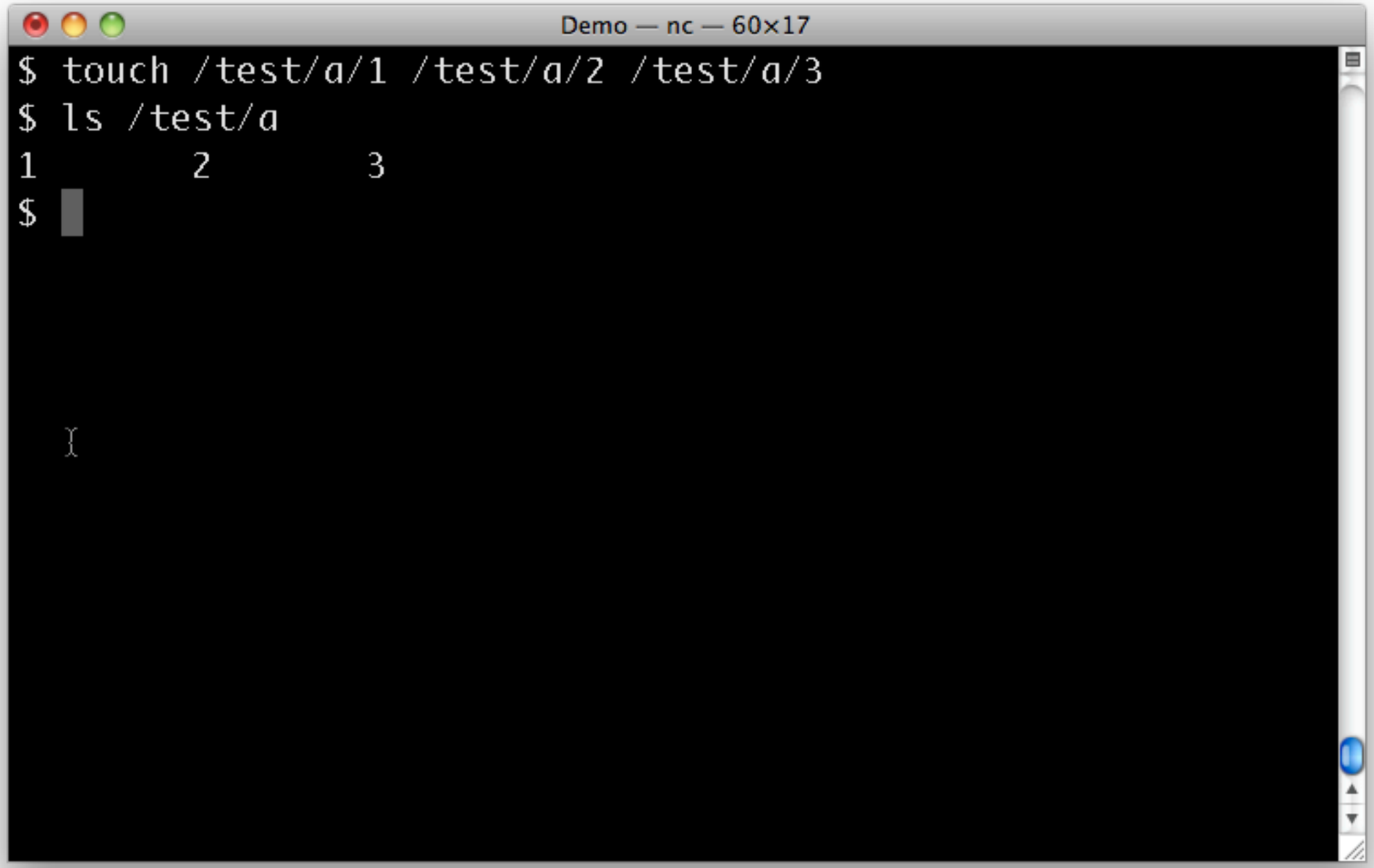
Before we start messing with snapshots, let’s put some trivial contents in the test tree.



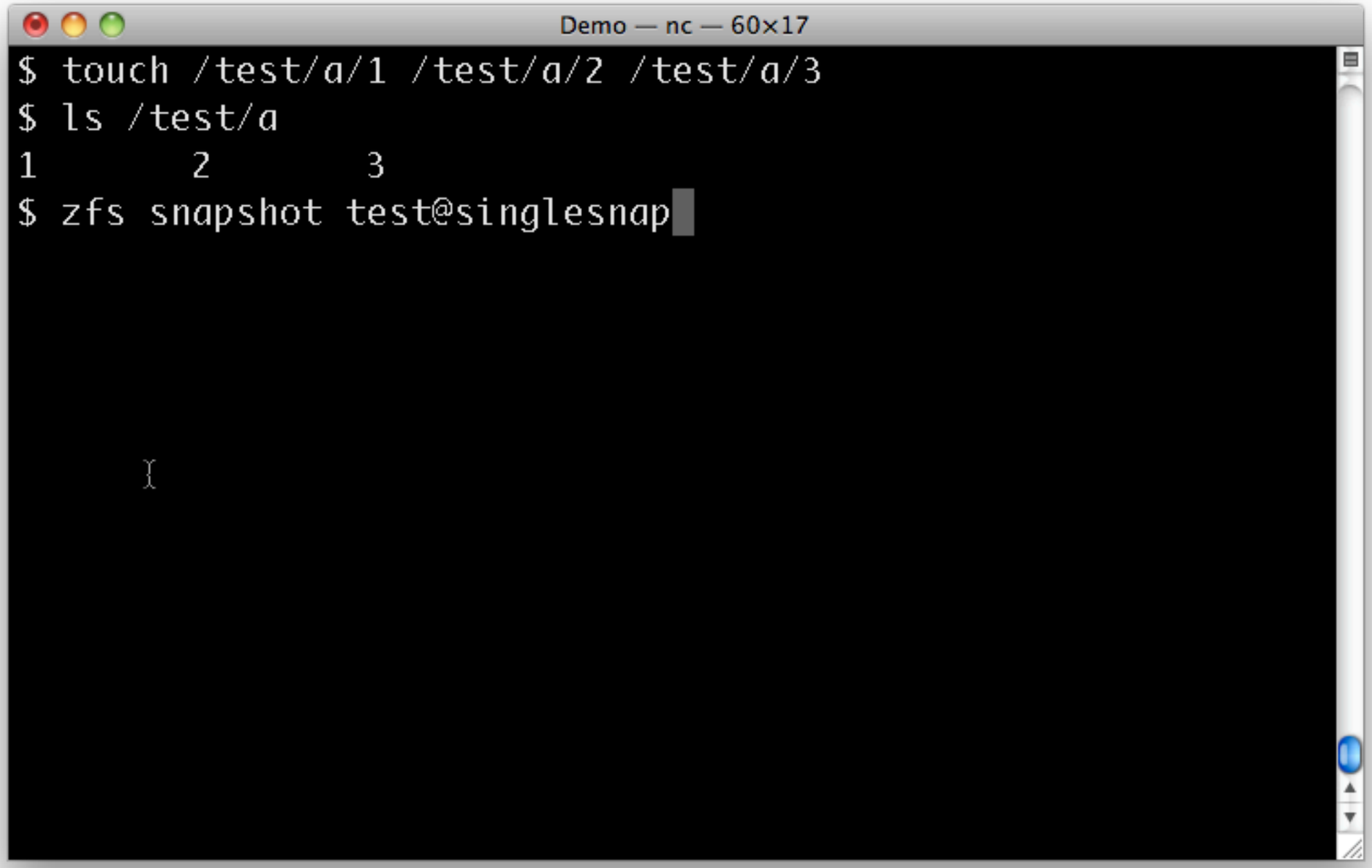
AEROSPACE



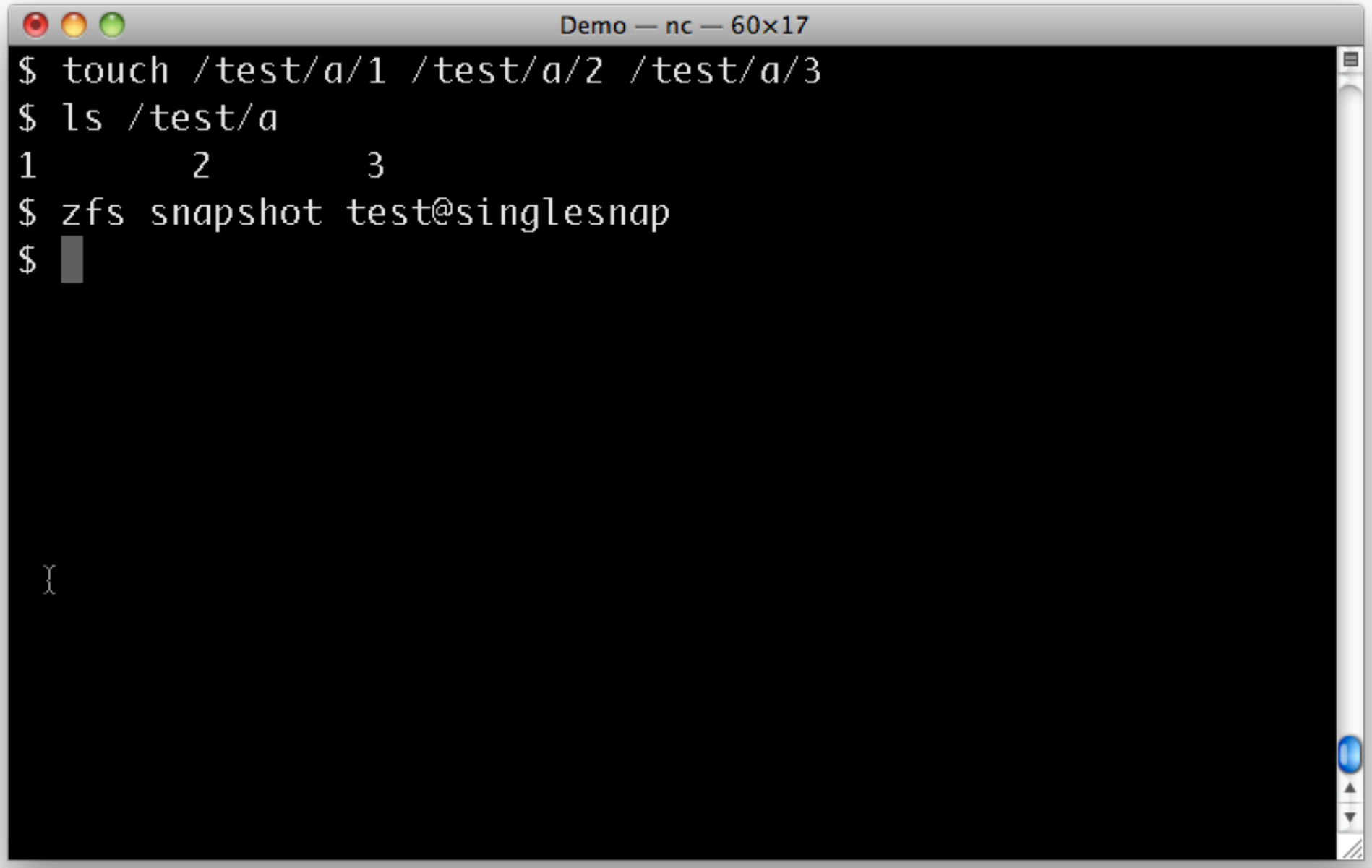
AEROSPACE



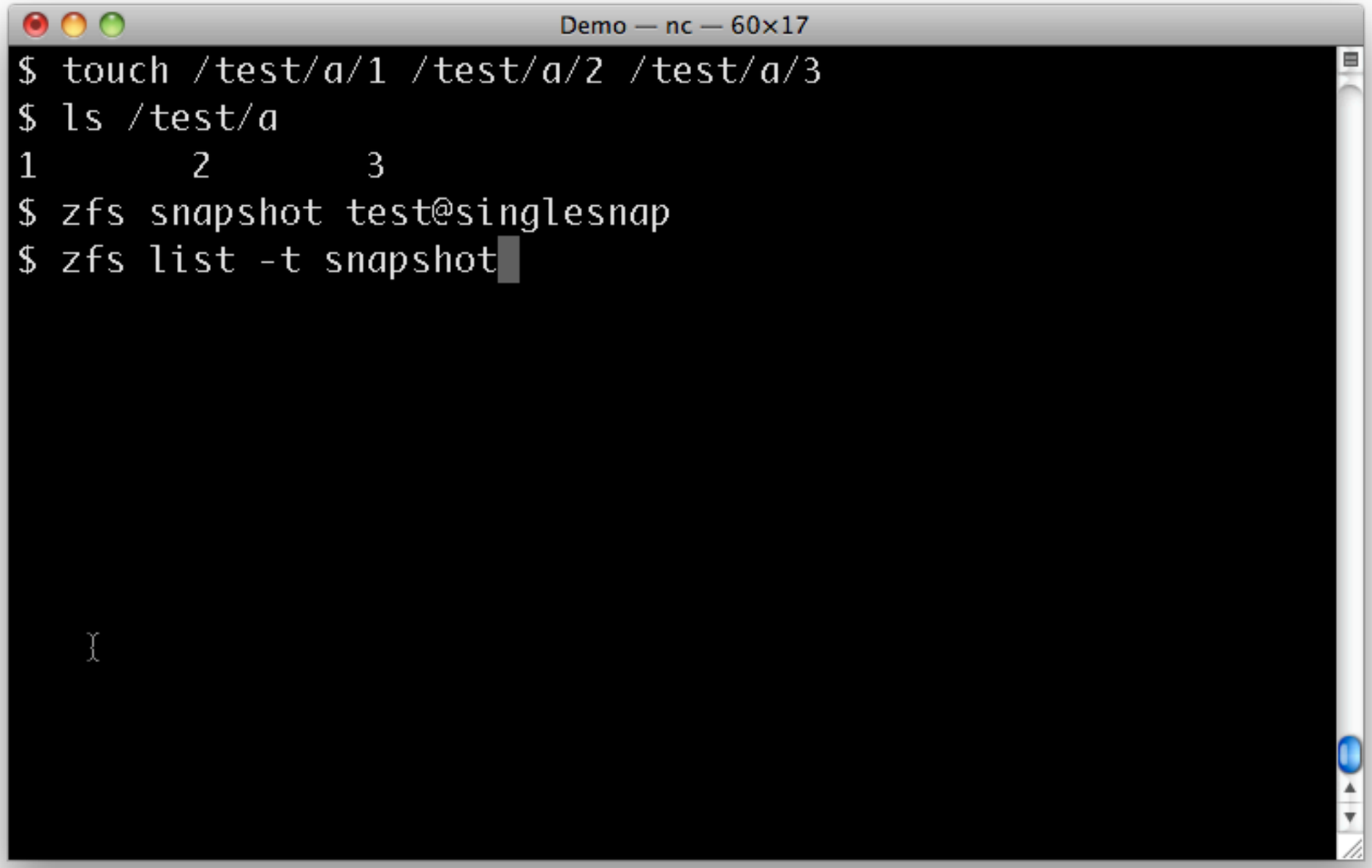
AEROSPACE


Demo — nc — 60×17
$ touch /test/a/1 /test/a/2 /test/a/3
$ ls /test/a
1	2	3
$ zfs snapshot test@singlesnap

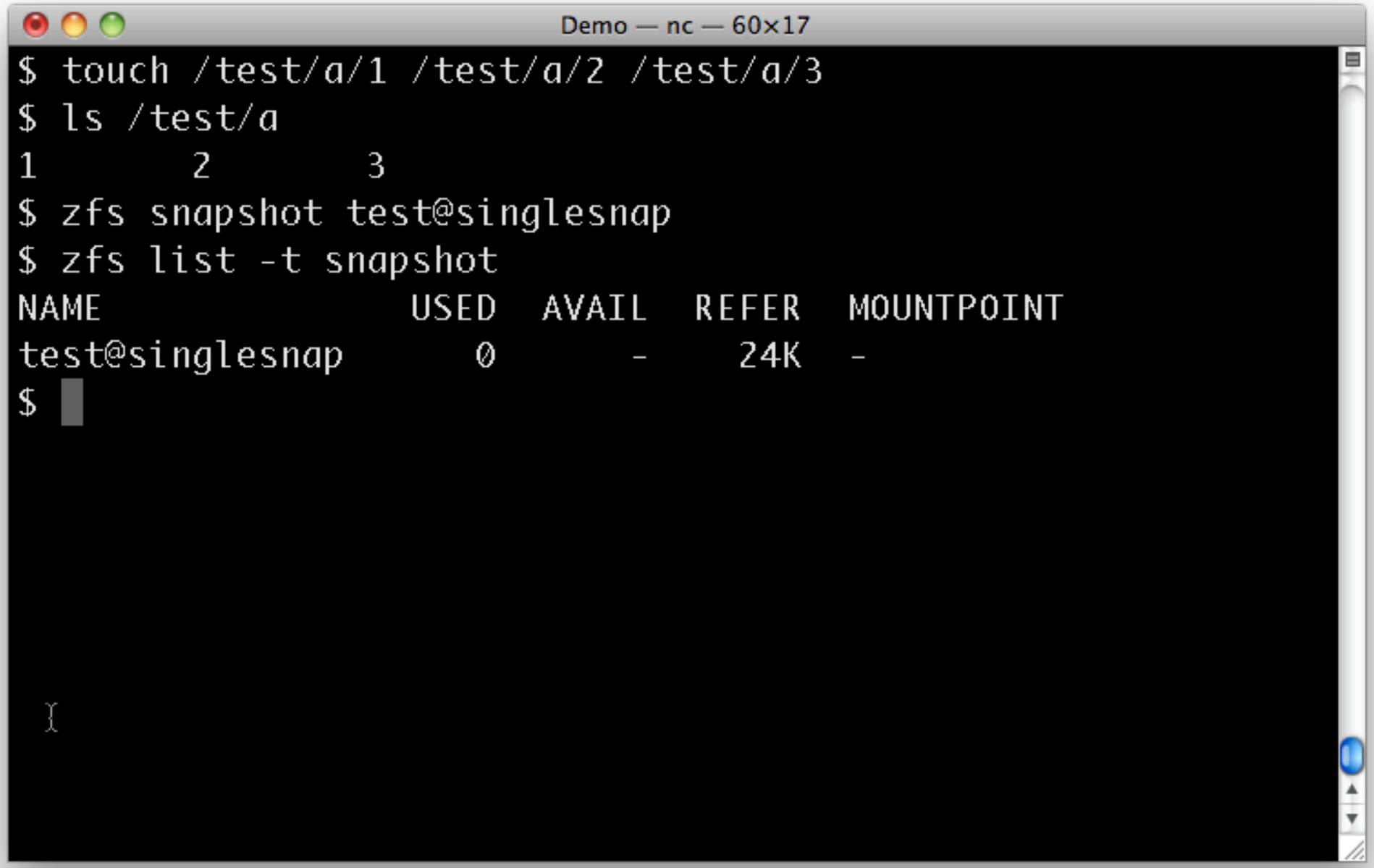
AEROSPACE


Demo — nc — 60×17
$ touch /test/a/1 /test/a/2 /test/a/3
$ ls /test/a
1	2	3
$ zfs snapshot test@singlesnap
$

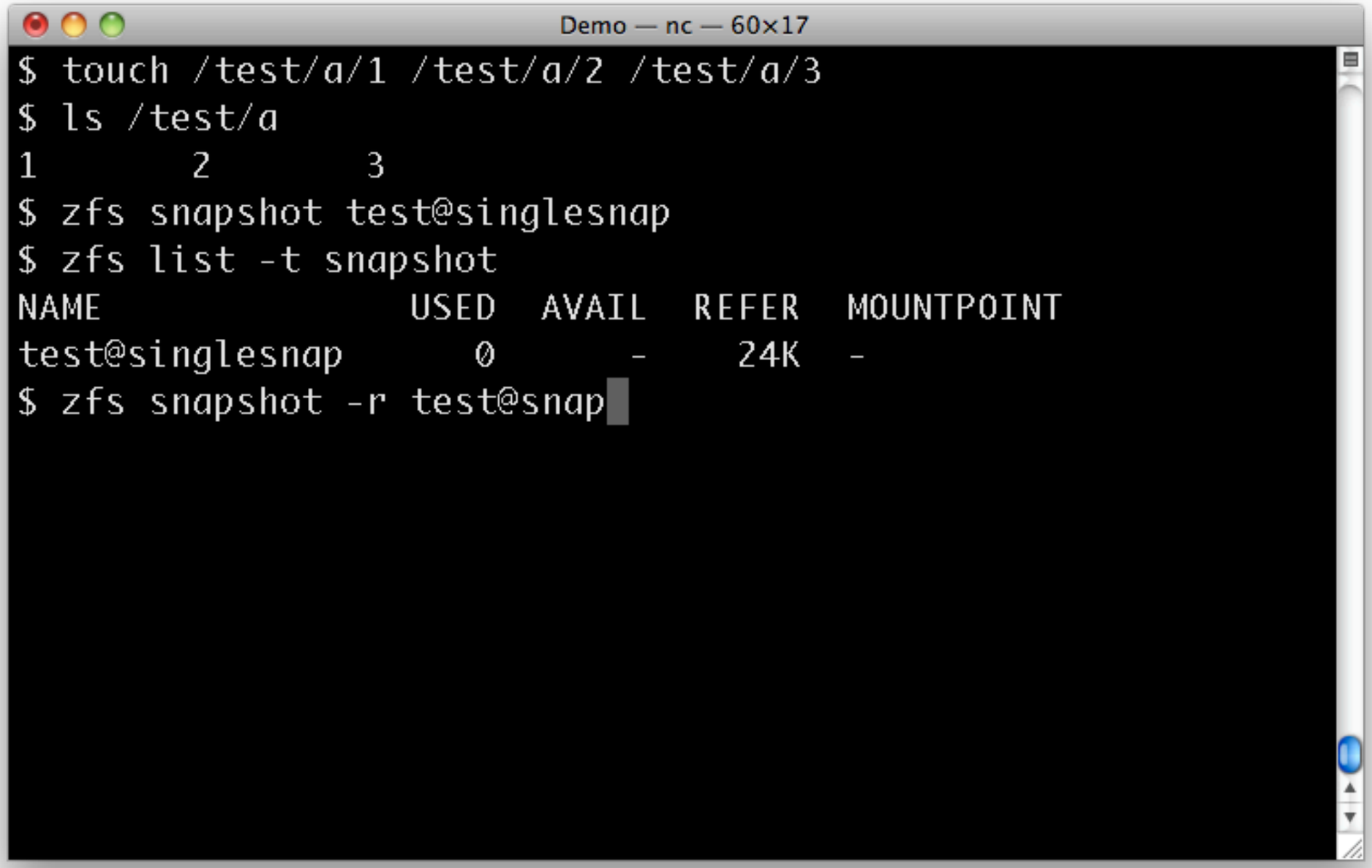
AEROSPACE


Demo — nc — 60×17
$ touch /test/a/1 /test/a/2 /test/a/3
$ ls /test/a
1	2	3
$ zfs snapshot test@singlesnap
$ zfs list -t snapshot

AEROSPACE

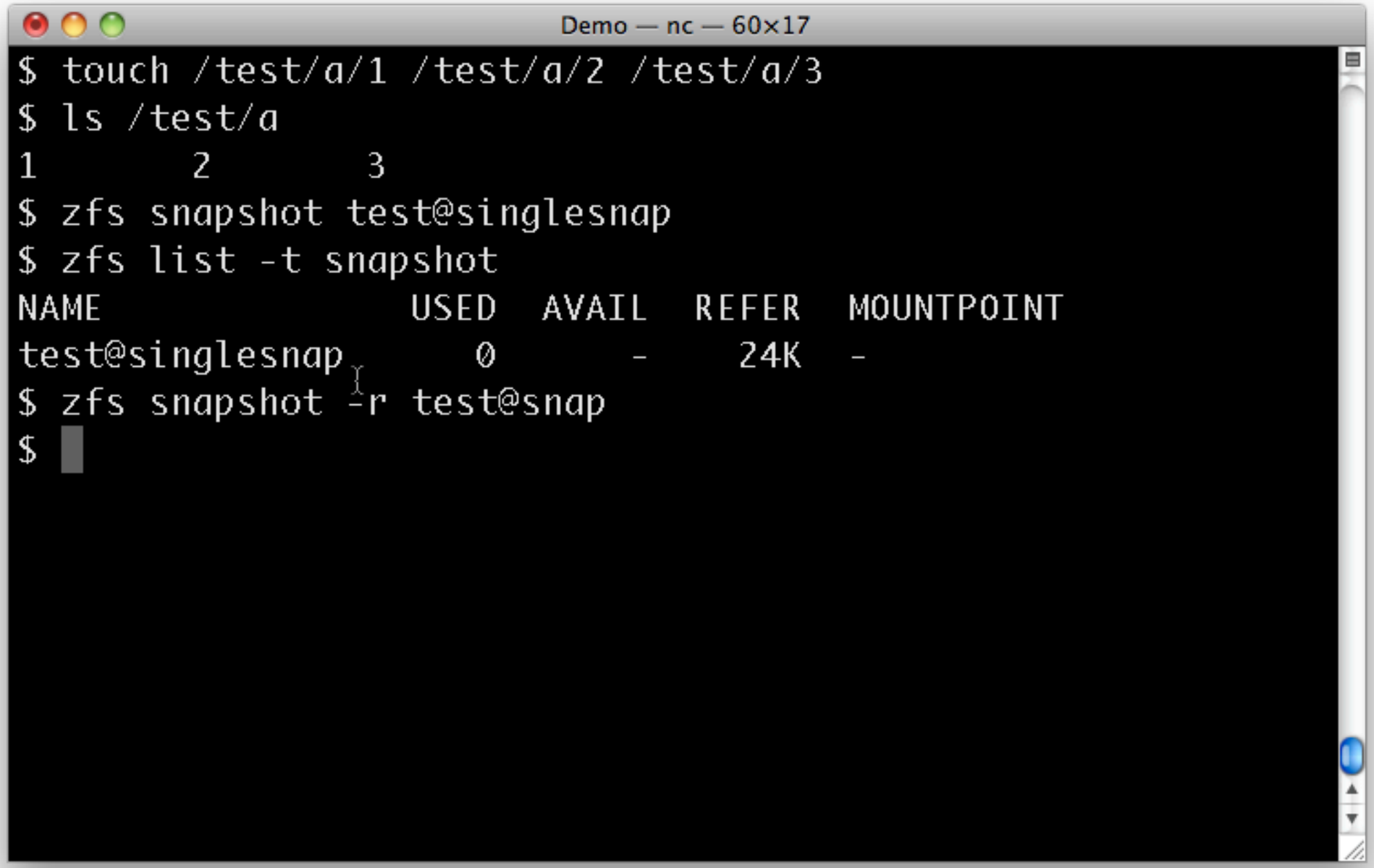


By default snapshots are on a single filesystem or volume



AEROSPACE

You can also make recursive snapshots to cover a whole hierarchy.


Demo — nc — 60×17
$ touch /test/a/1 /test/a/2 /test/a/3
$ ls /test/a
1	2	3
$ zfs snapshot test@singlesnap
$ zfs list -t snapshot
NAME	USED	AVAIL	REFER	MOUNTPOINT
test@singlesnap	0	-	24K	-
$ zfs snapshot -r test@snap
$

AEROSPACE

Demo — nc — 60×17

```
$ touch /test/a/1 /test/a/2 /test/a/3
$ ls /test/a
1       2       3
$ zfs snapshot test@singlesnap
$ zfs list -t snapshot
NAME                 USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap         0      -    24K  -
$ zfs snapshot -r test@snap
$ zfs list -t snapshot
```

AEROSPACE

Demo — nc — 60×17

```
$ touch /test/a/1 /test/a/2 /test/a/3
$ ls /test/a
1       2       3
$ zfs snapshot test@singlesnap
$ zfs list -t snapshot
NAME                 USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap         0      -    24K  -
$ zfs snapshot -r test@snap
$ zfs list -t snapshot
NAME                 USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap         0      -    24K  -
test@snap               0      -    24K  -
test/a@snap             0      -    22K  -
test/b@snap             0      -    21K  -
$
```

AEROSPACE

Demo — nc — 60×17

```
$ touch /test/a/1 /test/a/2 /test/a/3
$ ls /test/a
1       2       3
$ zfs snapshot test@singlesnap
$ zfs list -t snapshot
NAME                USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap        0      -    24K  -
$ zfs snapshot -r test@snap
$ zfs list -t snapshot
NAME                USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap        0      -    24K  -
test@snap              0      -    24K  -
test/a@snap            0      -    22K  -
test/b@snap            0      -    21K  -
$ zfs destroy test@singlesnap
```

AEROSPACE

Monday, September 26, 2011

We won't use the non-recursive snapshot so I'll get it out of the way. You remove snapshots with the destroy command.

Demo — nc — 60×17

```
$ touch /test/a/1 /test/a/2 /test/a/3
$ ls /test/a
1       2       3
$ zfs snapshot test@singlesnap
$ zfs list -t snapshot
NAME                USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap        0      -    24K  -
$ zfs snapshot -r test@snap
$ zfs list -t snapshot
NAME                USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap        0      -    24K  -
test@snap              0      -    24K  -
test/a@snap            0      -    22K  -
test/b@snap            0      -    21K  -
$ zfs destroy test@singlesnap
$
```

AEROSPACE

Demo — nc — 60×17

```
$ touch /test/a/1 /test/a/2 /test/a/3
$ ls /test/a
1       2       3
$ zfs snapshot test@singlesnap
$ zfs list -t snapshot
NAME                USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap        0      -    24K  -
$ zfs snapshot -r test@snap
$ zfs list -t snapshot
NAME                USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap        0      -    24K  -
test@snap              0      -    24K  -
test/a@snap            0      -    22K  -
test/b@snap            0      -    21K  -
$ zfs destroy test@singlesnap
$ rm /test/a/*
```

AEROSPACE

Demo — nc — 60×17

```
$ touch /test/a/1 /test/a/2 /test/a/3
$ ls /test/a
1       2       3
$ zfs snapshot test@singlesnap
$ zfs list -t snapshot
NAME                 USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap         0      -    24K  -
$ zfs snapshot -r test@snap
$ zfs list -t snapshot
NAME                 USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap         0      -    24K  -
test@snap               0      -    24K  -
test/a@snap             0      -    22K  -
test/b@snap             0      -    21K  -
$ zfs destroy test@singlesnap
$ rm /test/a/*
$
```

AEROSPACE

Demo — nc — 60×17

```
$ touch /test/a/1 /test/a/2 /test/a/3
$ ls /test/a
1       2       3
$ zfs snapshot test@singlesnap
$ zfs list -t snapshot
NAME                USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap        0      -    24K  -
$ zfs snapshot -r test@snap
$ zfs list -t snapshot
NAME                USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap        0      -    24K  -
test@snap              0      -    24K  -
test/a@snap            0      -    22K  -
test/b@snap            0      -    21K  -
$ zfs destroy test@singlesnap
$ rm /test/a/*
$ ls /test/a
```

AEROSPACE

Demo — nc — 60×17

```
$ ls /test/a
1       2       3
$ zfs snapshot test@singlesnap
$ zfs list -t snapshot
NAME                 USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap         0      -    24K  -
$ zfs snapshot -r test@snap
$ zfs list -t snapshot
NAME                 USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap         0      -    24K  -
test@snap               0      -    24K  -
test/a@snap             0      -    22K  -
test/b@snap             0      -    21K  -
$ zfs destroy test@singlesnap
$ rm /test/a/*
$ ls /test/a
$
```

AEROSPACE

Demo — nc — 60×17

```
$ ls /test/a
1       2       3
$ zfs snapshot test@singlesnap
$ zfs list -t snapshot
NAME                  USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap          0      -    24K  -
$ zfs snapshot -r test@snap
$ zfs list -t snapshot
NAME                  USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap          0      -    24K  -
test@snap                0      -    24K  -
test/a@snap              0      -    22K  -
test/b@snap              0      -    21K  -
$ zfs destroy test@singlesnap
$ rm /test/a/
$ ls /test/a
$
```

OOPS!

AEROSPACE

Demo — nc — 60×17

```
$ ls /test/a
1       2       3
$ zfs snapshot test@singlesnap
$ zfs list -t snapshot
NAME                USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap        0      -    24K  -
$ zfs snapshot -r test@snap
$ zfs list -t snapshot
NAME                USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap        0      -    24K  -
test@snap              0      -    24K  -
test/a@snap            0      -    22K  -
test/b@snap            0      -    21K  -
$ zfs destroy test@singlesnap
$ rm /test/a/*
$ ls /test/a
$ ls /test/a/.zfs/snapshot
```

AEROSPACE

Demo — nc — 60×17

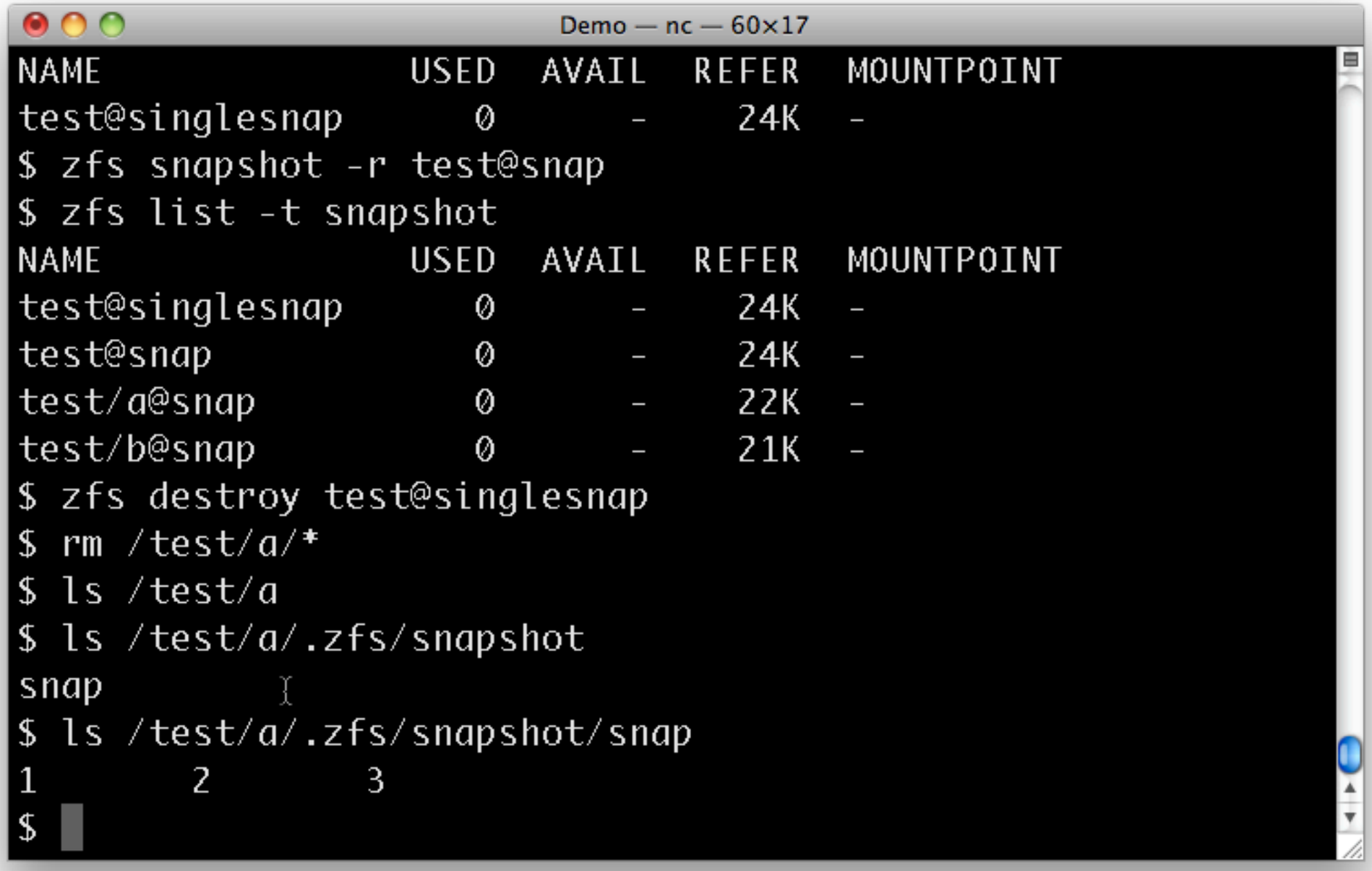
```
$ zfs snapshot test@singlesnap
$ zfs list -t snapshot
NAME                 USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap         0      -    24K  -
$ zfs snapshot -r test@snap
$ zfs list -t snapshot
NAME                 USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap         0      -    24K  -
test@snap               0      -    24K  -
test/a@snap             0      -    22K  -
test/b@snap             0      -    21K  -
$ zfs destroy test@singlesnap
$ rm /test/a/*
$ ls /test/a
$ ls /test/a/.zfs/snapshot
snap
$
```

AEROSPACE

Demo — nc — 60×17

```
$ zfs snapshot test@singlesnap
$ zfs list -t snapshot
NAME                 USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap         0      -    24K  -
$ zfs snapshot -r test@snap
$ zfs list -t snapshot
NAME                 USED  AVAIL  REFER  MOUNTPOINT
test@singlesnap         0      -    24K  -
test@snap               0      -    24K  -
test/a@snap             0      -    22K  -
test/b@snap             0      -    21K  -
$ zfs destroy test@singlesnap
$ rm /test/a/*
$ ls /test/a
$ ls /test/a/.zfs/snapshot
snap
$ ls /test/a/.zfs/snapshot/snap
```

AEROSPACE



AEROSPACE

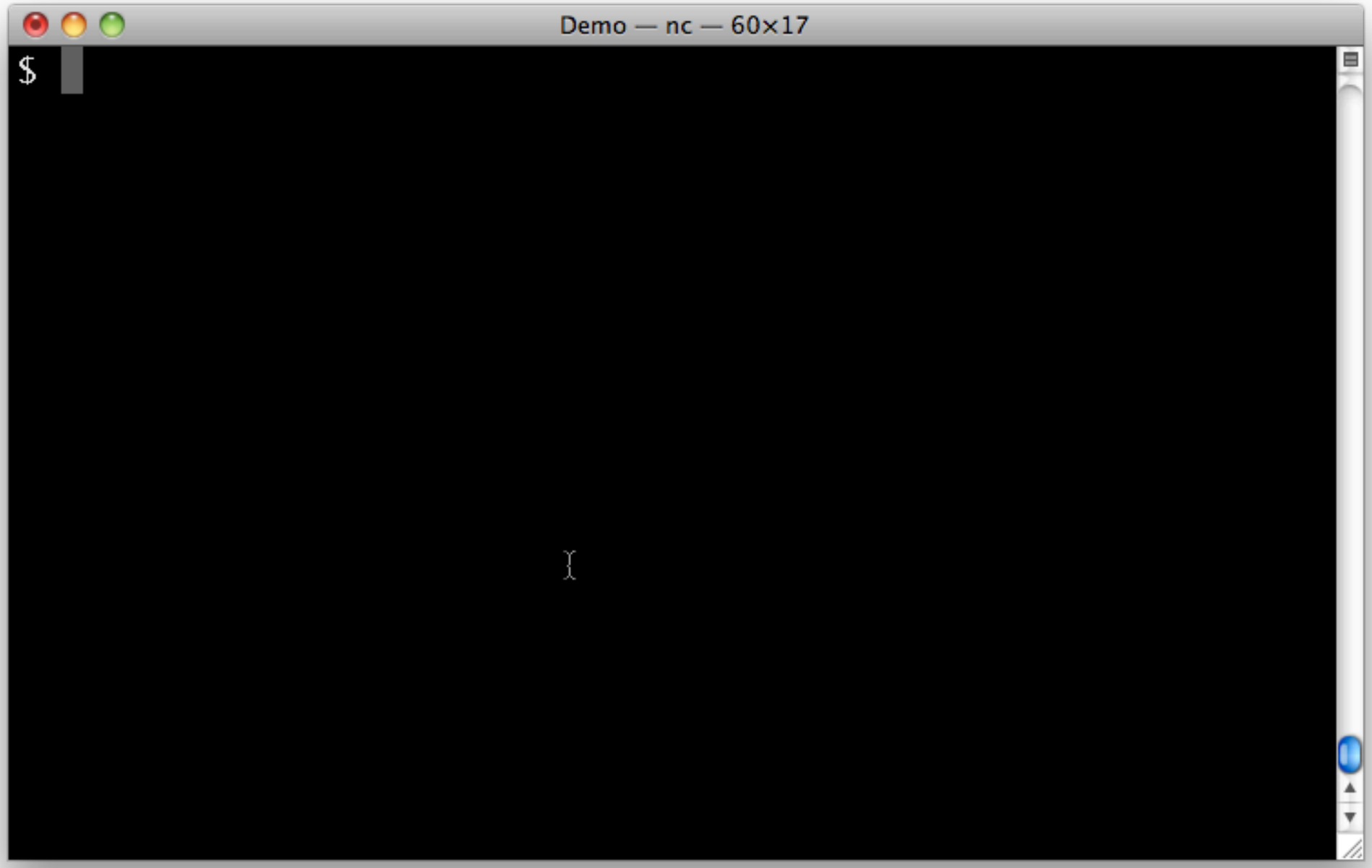
Monday, September 26, 2011

For a simple change like this, we might just copy the files out of the snapshot to recover them. However, ZFS gives us other options that might be more appropriate in more complicated cases.

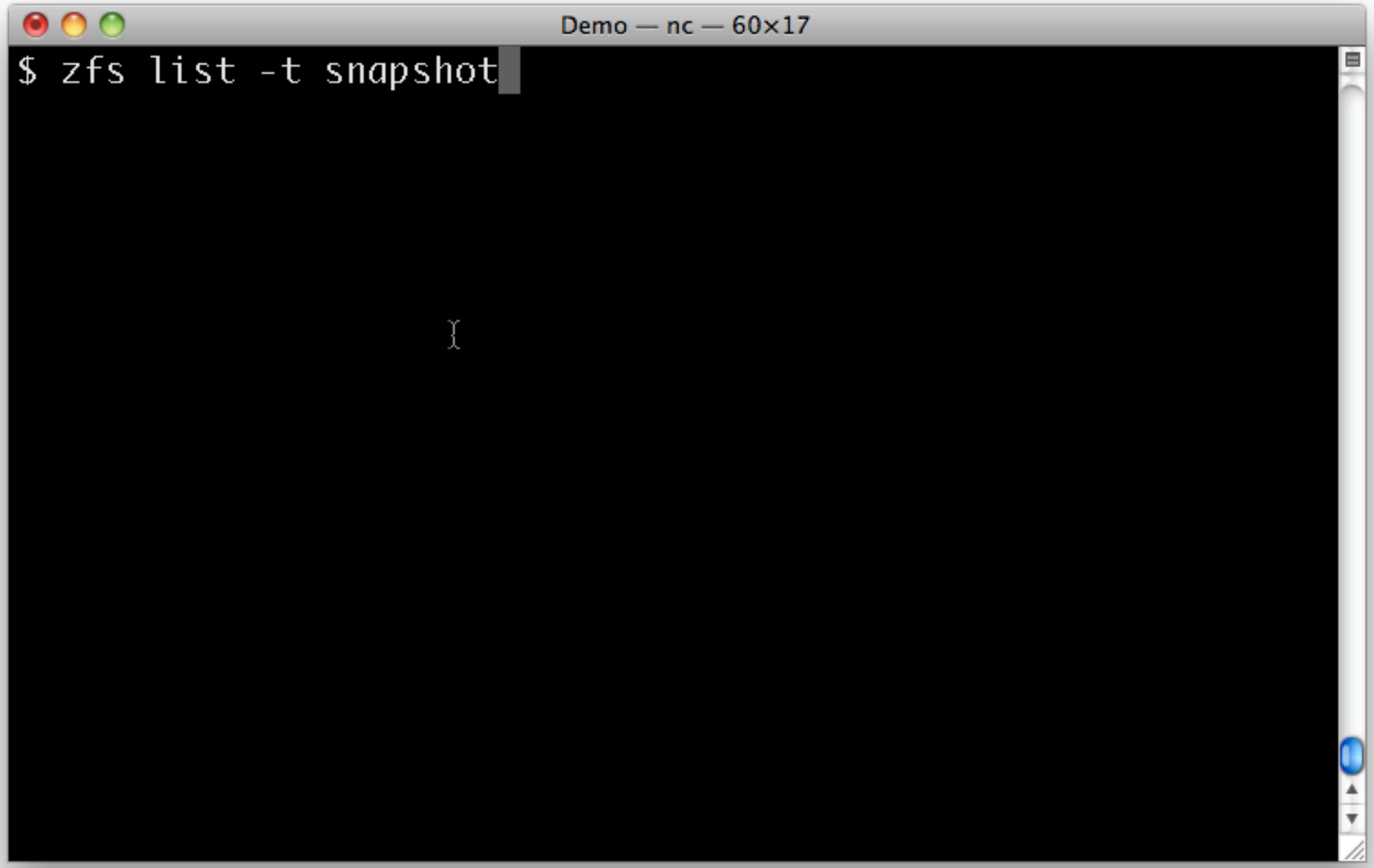
Clones

AEROSPACE

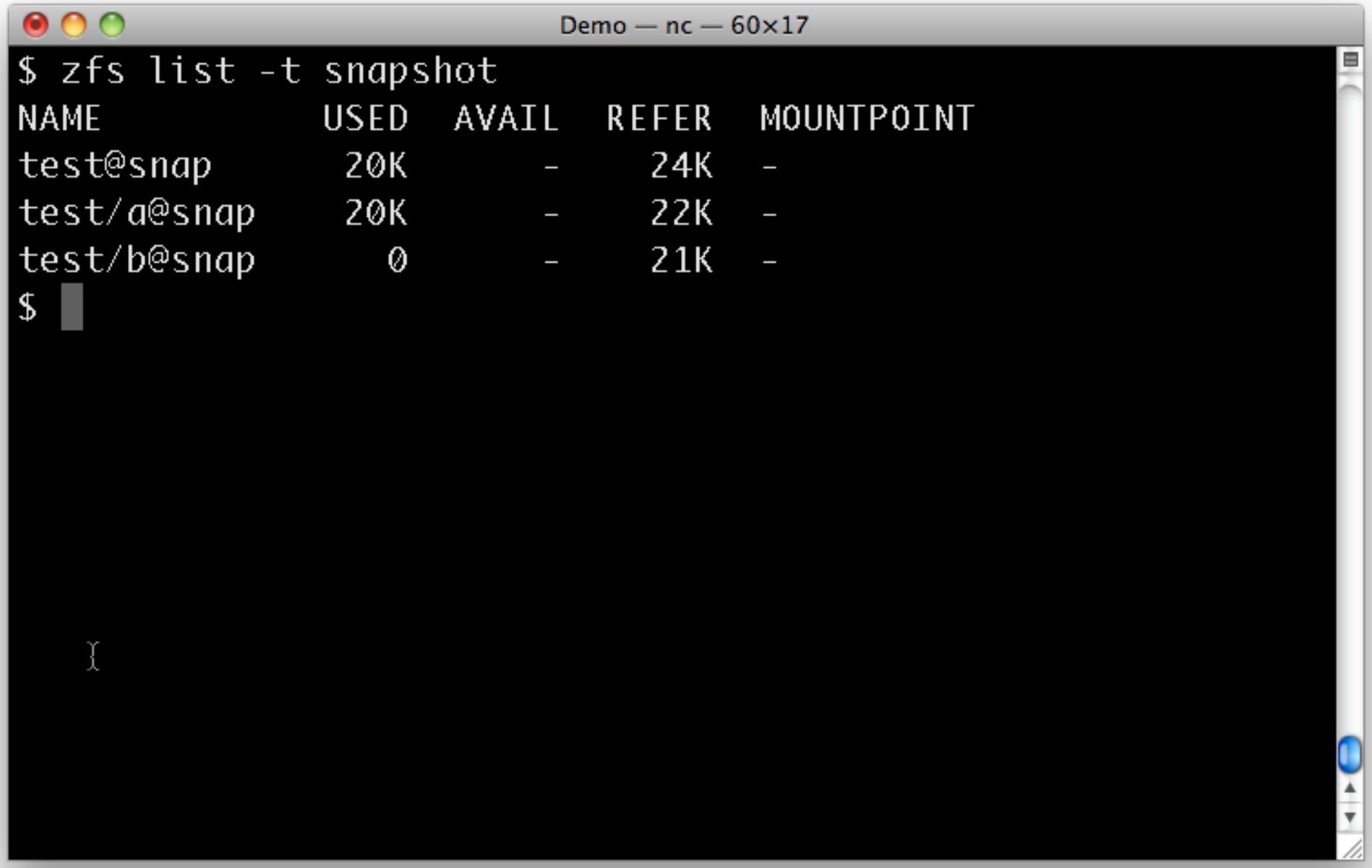
While piecemeal restoration of files is useful and can be a great self service mechanism for users it's not ideal for complex problems. Snapshot clones provide a more full features mechanism for access to snapshots.



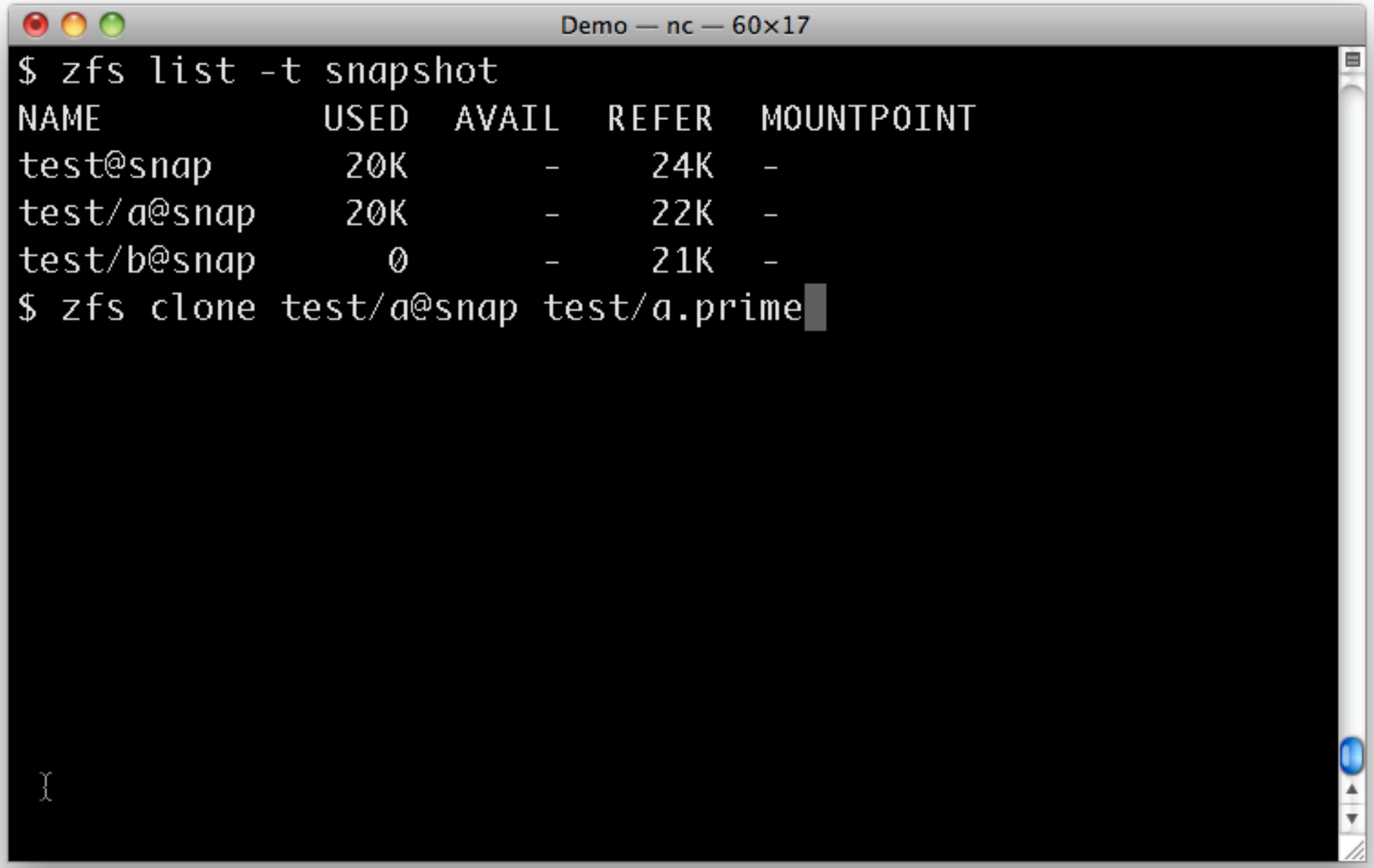
AEROSPACE



AEROSPACE

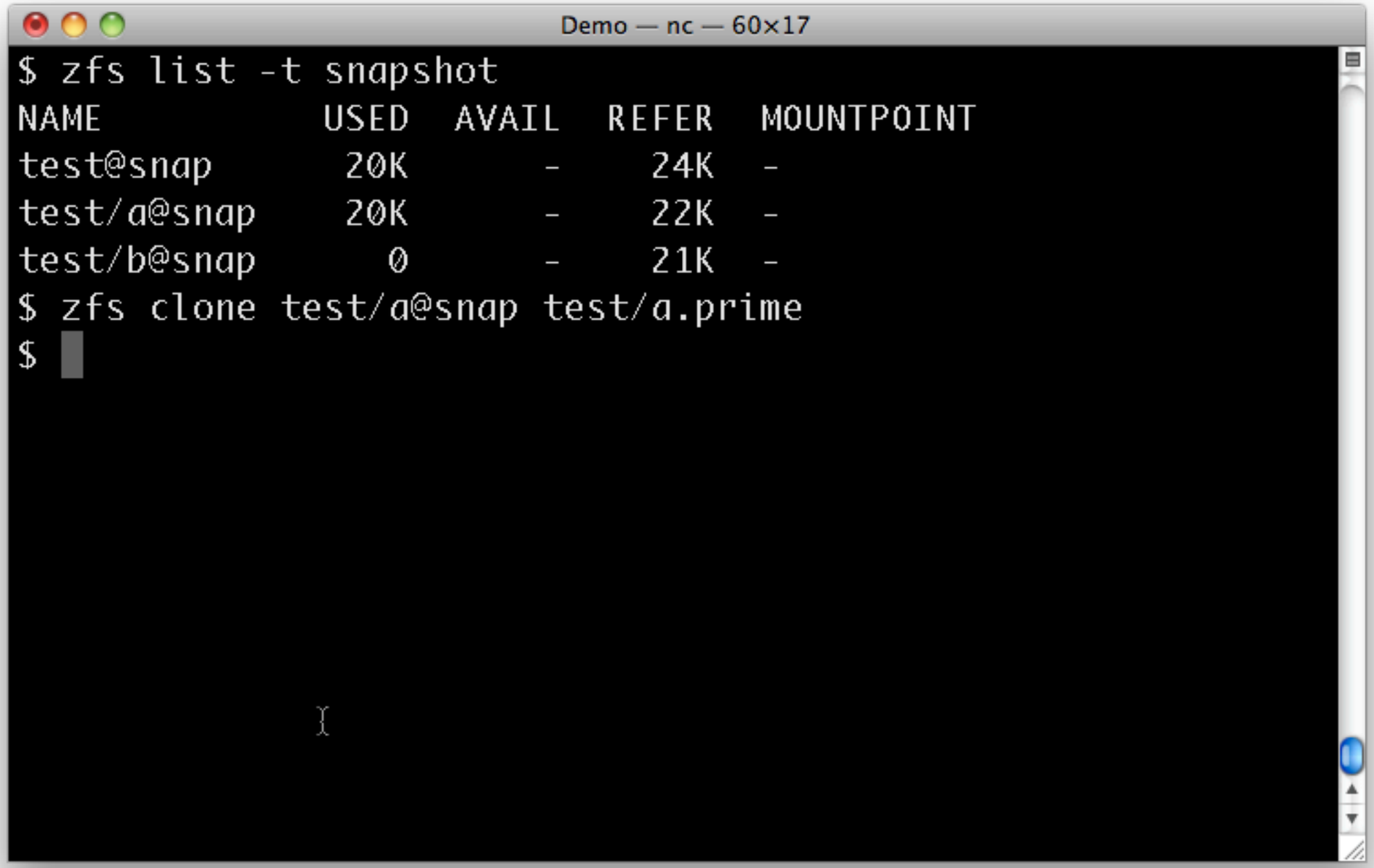

Demo — nc — 60×17
$ zfs list -t snapshot
NAME USED AVAIL REFER MOUNTPOINT
test@snap 20K - 24K -
test/a@snap 20K - 22K -
test/b@snap 0 - 21K -
$

AEROSPACE

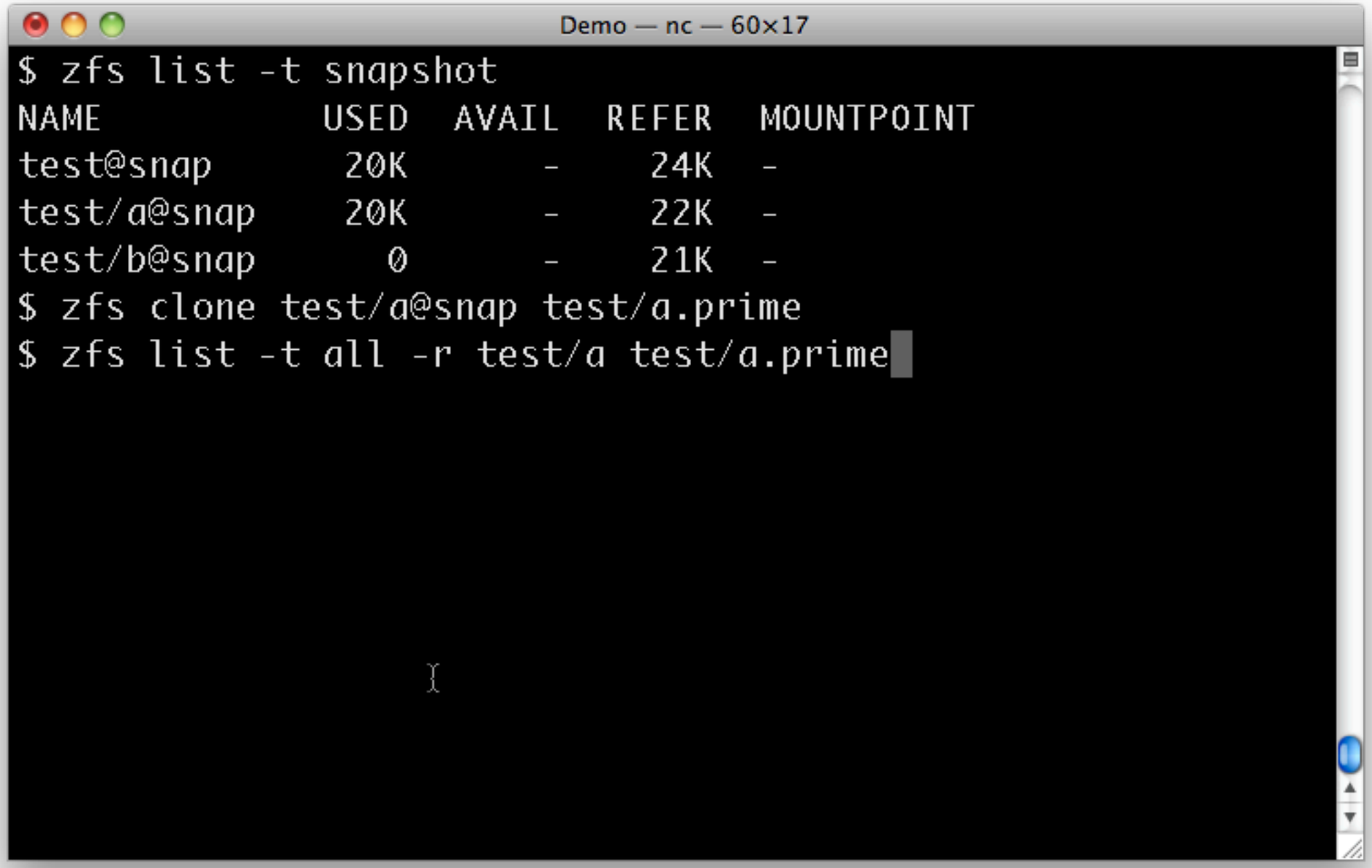


AEROSPACE

A clone is a writable copy of the snapshot. In effect a fork of it.


Demo — nc — 60×17
$ zfs list -t snapshot
NAME USED AVAIL REFER MOUNTPOINT
test@snap 20K - 24K -
test/a@snap 20K - 22K -
test/b@snap 0 - 21K -
$ zfs clone test/a@snap test/a.prime
$

AEROSPACE


Demo — nc — 60×17
$ zfs list -t snapshot
NAME USED AVAIL REFER MOUNTPOINT
test@snap 20K - 24K -
test/a@snap 20K - 22K -
test/b@snap 0 - 21K -
$ zfs clone test/a@snap test/a.prime
$ zfs list -t all -r test/a test/a.prime

AEROSPACE

Demo — nc — 60×17

```
$ zfs list -t snapshot
NAME           USED  AVAIL  REFER  MOUNTPOINT
test@snap       20K      -    24K  -
test/a@snap     20K      -    22K  -
test/b@snap       0      -    21K  -
$ zfs clone test/a@snap test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME            USED  AVAIL  REFER  MOUNTPOINT
test/a           41K   457G    21K  /test/a
test/a@snap      20K      -    22K  -
test/a.prime       0   457G    22K  /test/a.prime
$
```

AEROSPACE

Demo — nc — 60×17

```
$ zfs list -t snapshot
NAME           USED  AVAIL  REFER  MOUNTPOINT
test@snap       20K      -    24K  -
test/a@snap     20K      -    22K  -
test/b@snap       0      -    21K  -
$ zfs clone test/a@snap test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME            USED  AVAIL  REFER  MOUNTPOINT
test/a           41K   457G    21K  /test/a
test/a@snap      20K      -    22K  -
test/a.prime       0   457G    22K  /test/a.prime
$ ls /test/a.prime
```

AEROSPACE

Demo — nc — 60×17

```
$ zfs list -t snapshot
NAME           USED  AVAIL  REFER  MOUNTPOINT
test@snap       20K      -    24K  -
test/a@snap     20K      -    22K  -
test/b@snap       0      -    21K  -
$ zfs clone test/a@snap test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME            USED  AVAIL  REFER  MOUNTPOINT
test/a           41K   457G    21K  /test/a
test/a@snap      20K      -    22K  -
test/a.prime       0   457G    22K  /test/a.prime
$ ls /test/a.prime
1         2         3
$
```

AEROSPACE

Demo — nc — 60×17

```
$ zfs list -t snapshot
NAME           USED  AVAIL  REFER  MOUNTPOINT
test@snap       20K      -    24K  -
test/a@snap     20K      -    22K  -
test/b@snap       0      -    21K  -
$ zfs clone test/a@snap test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME            USED  AVAIL  REFER  MOUNTPOINT
test/a           41K   457G    21K  /test/a
test/a@snap      20K      -    22K  -
test/a.prime       0   457G    22K  /test/a.prime
$ ls /test/a.prime
1        2        3
$ zfs promote test/a.prime
```

AEROSPACE

Demo — nc — 60×17

```
$ zfs list -t snapshot
NAME           USED  AVAIL  REFER  MOUNTPOINT
test@snap       20K      -    24K  -
test/a@snap     20K      -    22K  -
test/b@snap       0      -    21K  -
$ zfs clone test/a@snap test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME            USED  AVAIL  REFER  MOUNTPOINT
test/a           41K   457G    21K  /test/a
test/a@snap      20K      -    22K  -
test/a.prime       0   457G    22K  /test/a.prime
$ ls /test/a.prime
1       2       3
$ zfs promote test/a.prime
$
```

AEROSPACE

Demo — nc — 60×17

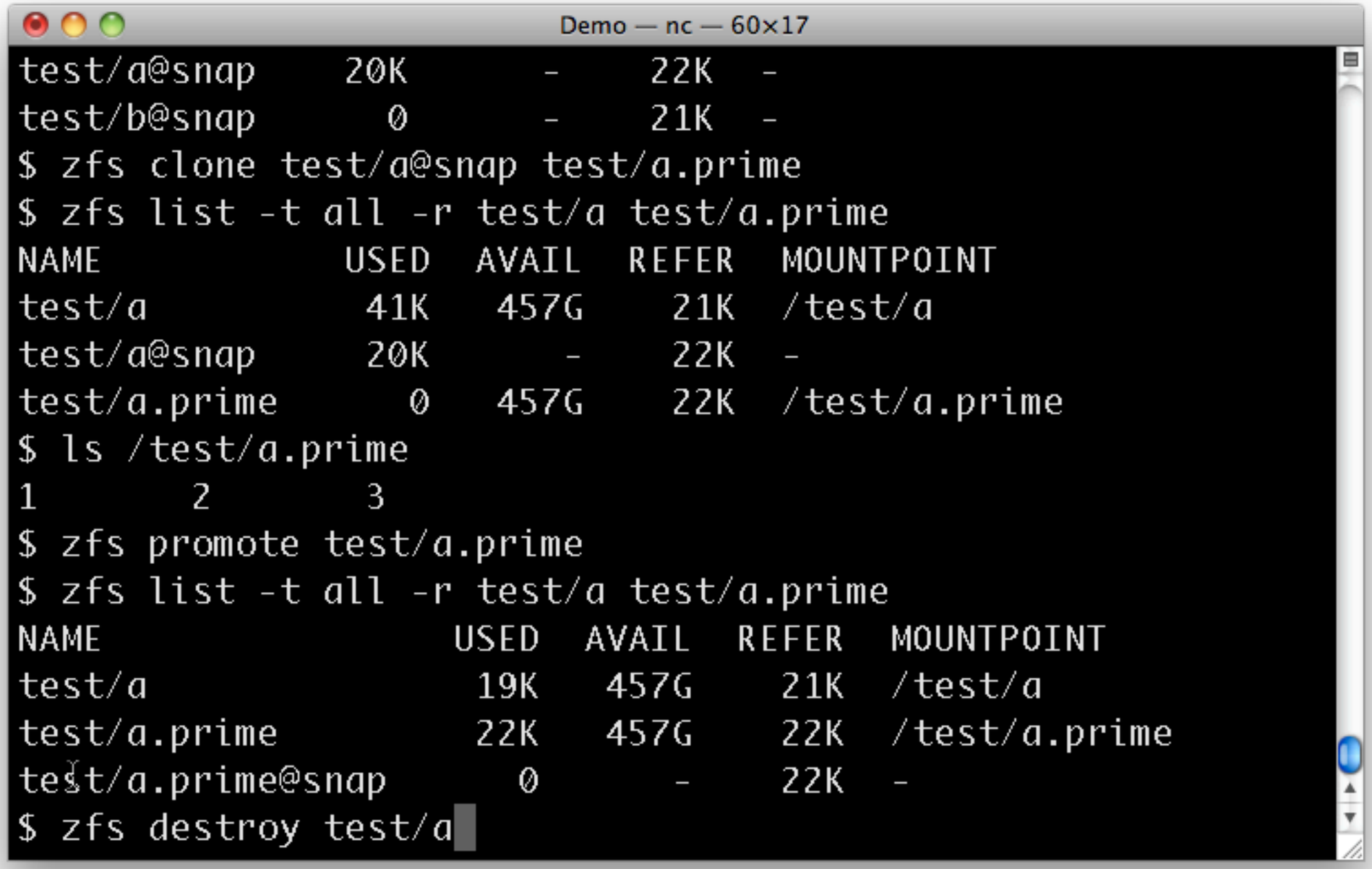
```
$ zfs list -t snapshot
NAME           USED  AVAIL  REFER  MOUNTPOINT
test@snap       20K      -    24K  -
test/a@snap     20K      -    22K  -
test/b@snap       0      -    21K  -
$ zfs clone test/a@snap test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME           USED  AVAIL  REFER  MOUNTPOINT
test/a          41K   457G    21K  /test/a
test/a@snap     20K      -    22K  -
test/a.prime      0   457G    22K  /test/a.prime
$ ls /test/a.prime
1       2       3
$ zfs promote test/a.prime
$ zfs list -t all -r test/a test/a.prime
```

AEROSPACE

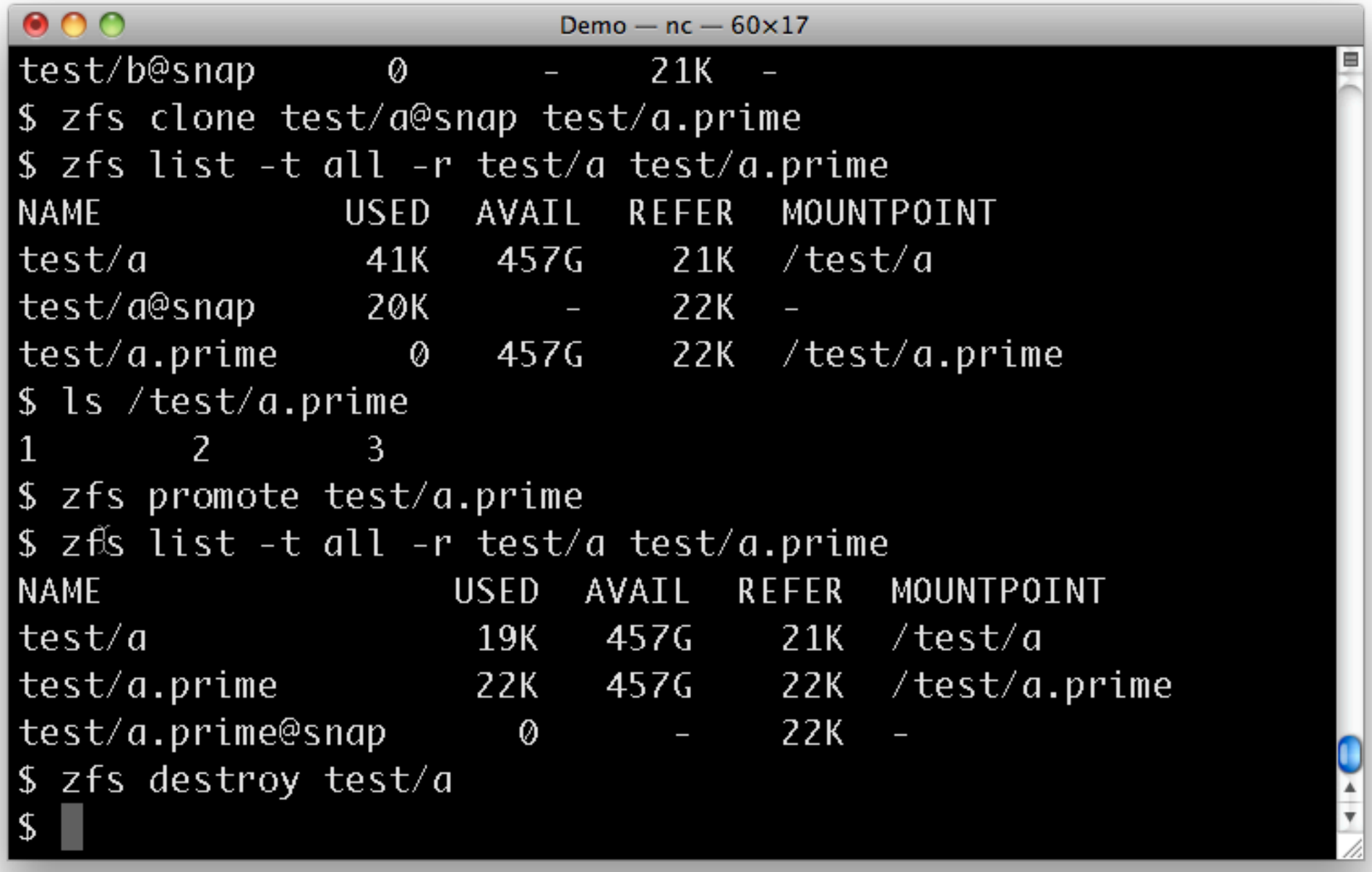
Demo — nc — 60×17

```
test/a@snap    20K      -    22K  -
test/b@snap      0      -    21K  -
$ zfs clone test/a@snap test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME            USED  AVAIL  REFER  MOUNTPOINT
test/a           41K   457G    21K  /test/a
test/a@snap      20K      -    22K  -
test/a.prime       0   457G    22K  /test/a.prime
$ ls /test/a.prime
1       2       3
$ zfs promote test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME                  USED  AVAIL  REFER  MOUNTPOINT
test/a                 19K   457G    21K  /test/a
test/a.prime           22K   457G    22K  /test/a.prime
test/a.prime@snap        0      -    22K  -
$
```

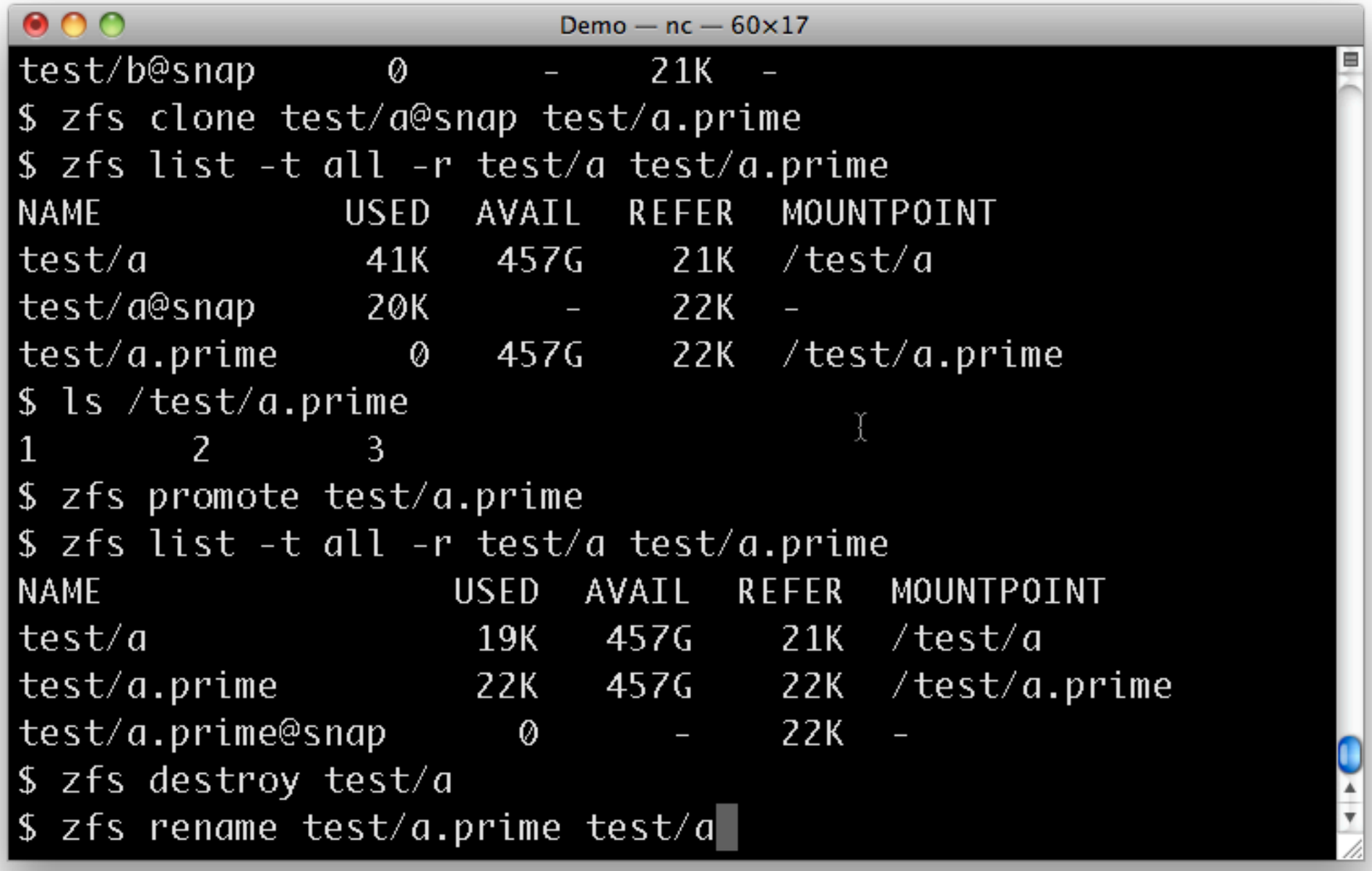
AEROSPACE


Demo — nc — 60×17
test/a@snap 20K - 22K -
test/b@snap 0 - 21K -
$ zfs clone test/a@snap test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME USED AVAIL REFER MOUNTPOINT
test/a 41K 457G 21K /test/a
test/a@snap 20K - 22K -
test/a.prime 0 457G 22K /test/a.prime
$ ls /test/a.prime
1 2 3
$ zfs promote test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME USED AVAIL REFER MOUNTPOINT
test/a 19K 457G 21K /test/a
test/a.prime 22K 457G 22K /test/a.prime
test/a.prime@snap 0 - 22K -
$ zfs destroy test/a

AEROSPACE


Demo — nc — 60×17
test/b@snap 0 - 21K -
$ zfs clone test/a@snap test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME USED AVAIL REFER MOUNTPOINT
test/a 41K 457G 21K /test/a
test/a@snap 20K - 22K -
test/a.prime 0 457G 22K /test/a.prime
$ ls /test/a.prime
1 2 3
$ zfs promote test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME USED AVAIL REFER MOUNTPOINT
test/a 19K 457G 21K /test/a
test/a.prime 22K 457G 22K /test/a.prime
test/a.prime@snap 0 - 22K -
$ zfs destroy test/a
$

AEROSPACE


Demo — nc — 60×17
test/b@snap 0 - 21K -
$ zfs clone test/a@snap test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME USED AVAIL REFER MOUNTPOINT
test/a 41K 457G 21K /test/a
test/a@snap 20K - 22K -
test/a.prime 0 457G 22K /test/a.prime
$ ls /test/a.prime
1 2 3
$ zfs promote test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME USED AVAIL REFER MOUNTPOINT
test/a 19K 457G 21K /test/a
test/a.prime 22K 457G 22K /test/a.prime
test/a.prime@snap 0 - 22K -
$ zfs destroy test/a
$ zfs rename test/a.prime test/a

AEROSPACE

Demo — nc — 60×17

```
$ zfs clone test/a@snap test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME             USED  AVAIL  REFER  MOUNTPOINT
test/a            41K   457G    21K  /test/a
test/a@snap       20K      -    22K  -
test/a.prime       0   457G    22K  /test/a.prime
$ ls /test/a.prime
1       2       3
$ zfs promote test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME                  USED  AVAIL  REFER  MOUNTPOINT
test/a                 19K   457G    21K  /test/a
test/a.prime           22K   457G    22K  /test/a.prime
test/a.prime@snap        0      -    22K  -
$ zfs destroy test/a
$ zfs rename test/a.prime test/a
$
```

AEROSPACE

Demo — nc — 60×17

```
$ zfs clone test/a@snap test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME             USED  AVAIL  REFER  MOUNTPOINT
test/a            41K   457G    21K  /test/a
test/a@snap       20K      -    22K  -
test/a.prime       0   457G    22K  /test/a.prime
$ ls /test/a.prime
1       2       3
$ zfs promote test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME                  USED  AVAIL  REFER  MOUNTPOINT
test/a                 19K   457G    21K  /test/a
test/a.prime           22K   457G    22K  /test/a.prime
test/a.prime@snap        0      -    22K  -
$ zfs destroy test/a
$ zfs rename test/a.prime test/a
$ zfs list
```

AEROSPACE

Demo — nc — 60×17

```
test/a.prime       0   457G    22K  /test/a.prime
$ ls /test/a.prime
1       2       3
$ zfs promote test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME                 USED  AVAIL  REFER  MOUNTPOINT
test/a                19K   457G    21K  /test/a
test/a.prime          22K   457G    22K  /test/a.prime
test/a.prime@snap       0      -    22K  -
$ zfs destroy test/a
$ zfs rename test/a.prime test/a
$ zfs list
NAME      USED  AVAIL  REFER  MOUNTPOINT
test      239K   457G    24K  /test
test/a     38K   457G    22K  /test/a
test/b     21K   457G    21K  /test/b
$
```

AEROSPACE

Demo — nc — 60×17

```
test/a.prime       0   457G    22K  /test/a.prime
$ ls /test/a.prime
1       2       3
$ zfs promote test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME                 USED  AVAIL  REFER  MOUNTPOINT
test/a                19K   457G    21K  /test/a
test/a.prime          22K   457G    22K  /test/a.prime
test/a.prime@snap       0      -    22K  -
$ zfs destroy test/a
$ zfs rename test/a.prime test/a
$ zfs list
NAME      USED  AVAIL  REFER  MOUNTPOINT
test      239K   457G    24K  /test
test/a     38K   457G    22K  /test/a
test/b     21K   457G    21K  /test/b
$ ls /test/a
```

AEROSPACE

Demo — nc — 60×17

```
1        2        3
$ zfs promote test/a.prime
$ zfs list -t all -r test/a test/a.prime
NAME                 USED  AVAIL  REFER  MOUNTPOINT
test/a                19K   457G    21K  /test/a
test/a.prime          22K   457G    22K  /test/a.prime
test/a.prime@snap       0      -    22K  -
$ zfs destroy test/a
$ zfs rename test/a.prime test/a
$ zfs list
NAME      USED  AVAIL  REFER  MOUNTPOINT
test      239K   457G    24K  /test
test/a     38K   457G    22K  /test/a
test/b     21K   457G    21K  /test/b
$ ls /test/a
1        2        3
$
```

AEROSPACE

Solving problems with ZFS

AEROSPACE

Monday, September 26, 2011

Now that I have introduced some ZFS basics, it's time to move on to some of the problems we have addressed with ZFS.

History of Aerospace Adoption of ZFS

- NAS filers (Solaris)
- Mirror server
- Aerosource
- General servers, root FS

AEROSPACE

Monday, September 26, 2011

First a brief history of our adoption of ZFS.
...
I will talk about the middle two

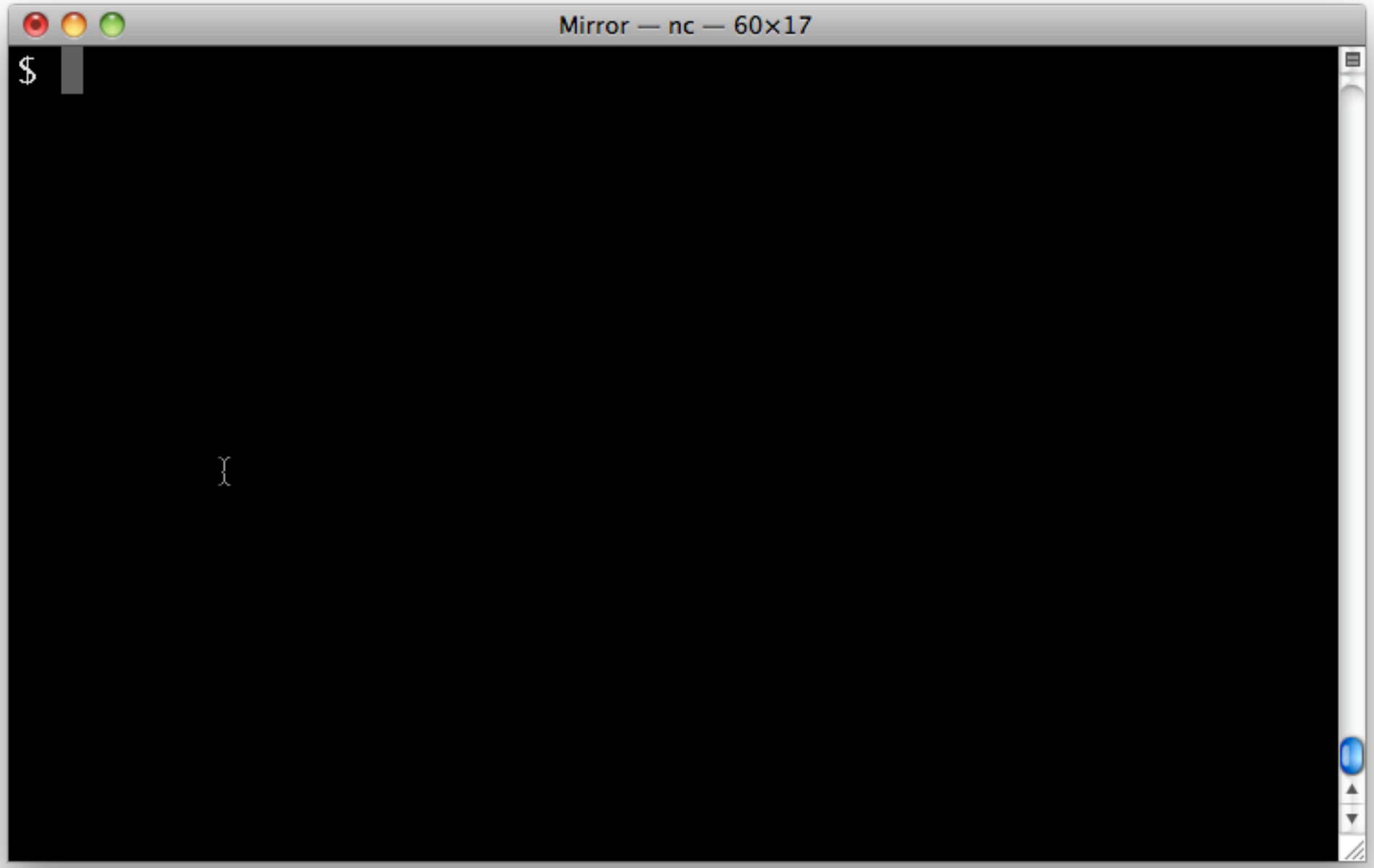
Mirroring with ZFS

AEROSPACE

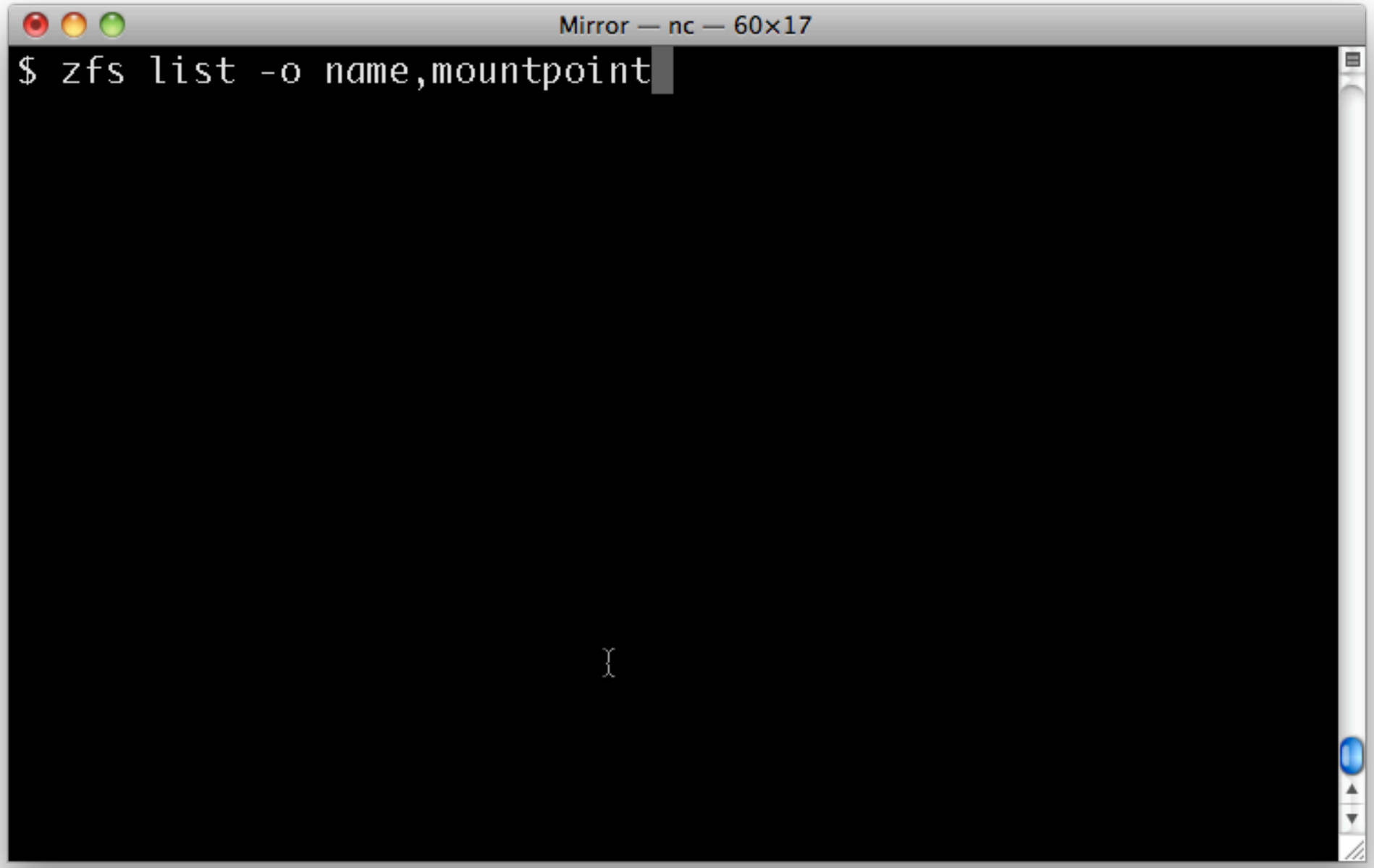
ZFS management model and cost: excellent fit for mirrors

AEROSPACE

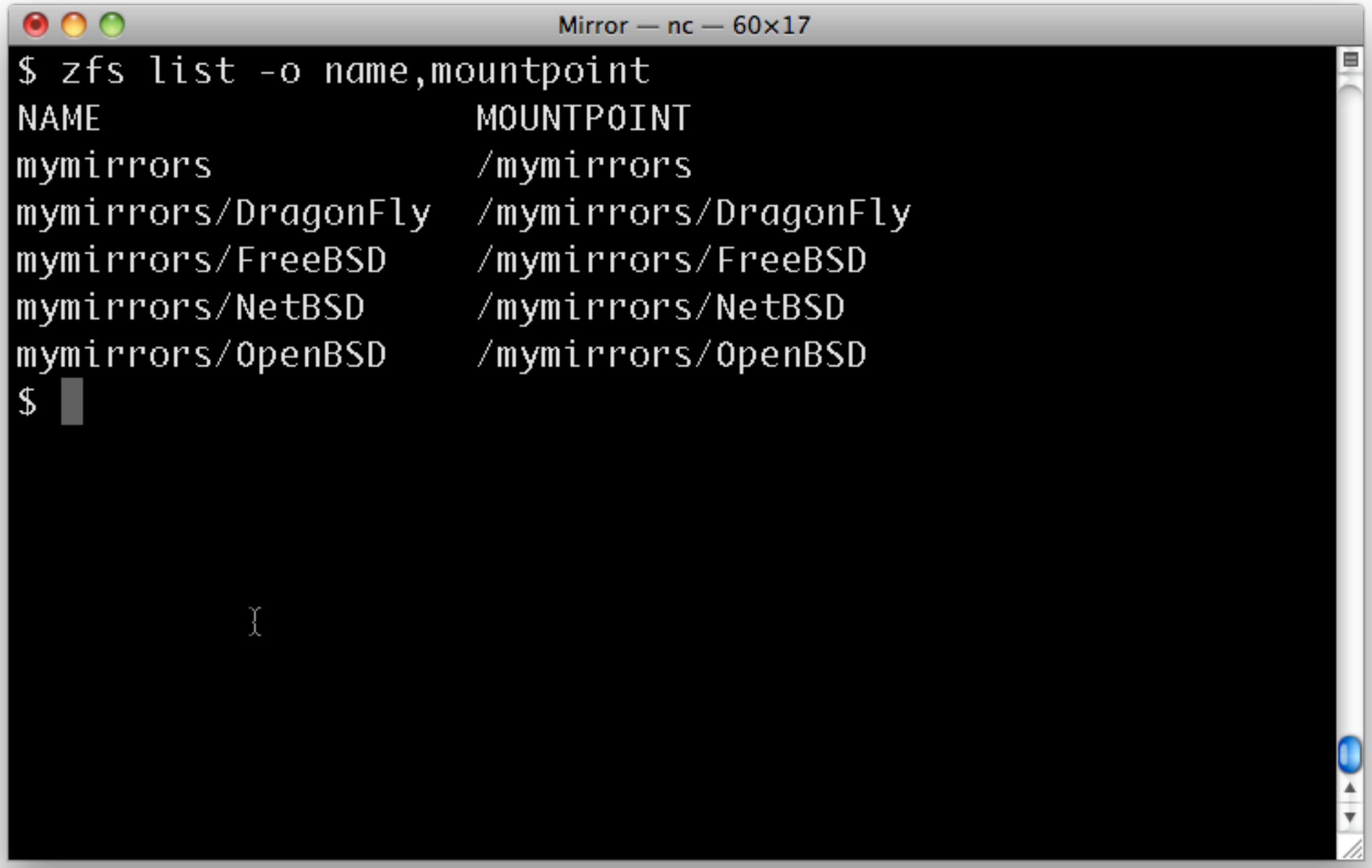
One big pool of storage
No need to buy raid controllers
Fast reboots



AEROSPACE



AEROSPACE



AEROSPACE

Monday, September 26, 2011

You could do a simple mirror layout like this.
It would give you some basic ZFS benefits including easy addition of storage.

Addressing mirror consistency with ZFS

AEROSPACE

Catalog and data files out of sync

- When you rsync a mirror you may get catalogs before the files they reference
- Arguably the problem is naive mirror software like rsync
  - Workarounds
    - Rsync --delete-after and --delay-updates
    - Use an rsync, test, repeat if needed loop

AEROSPACE

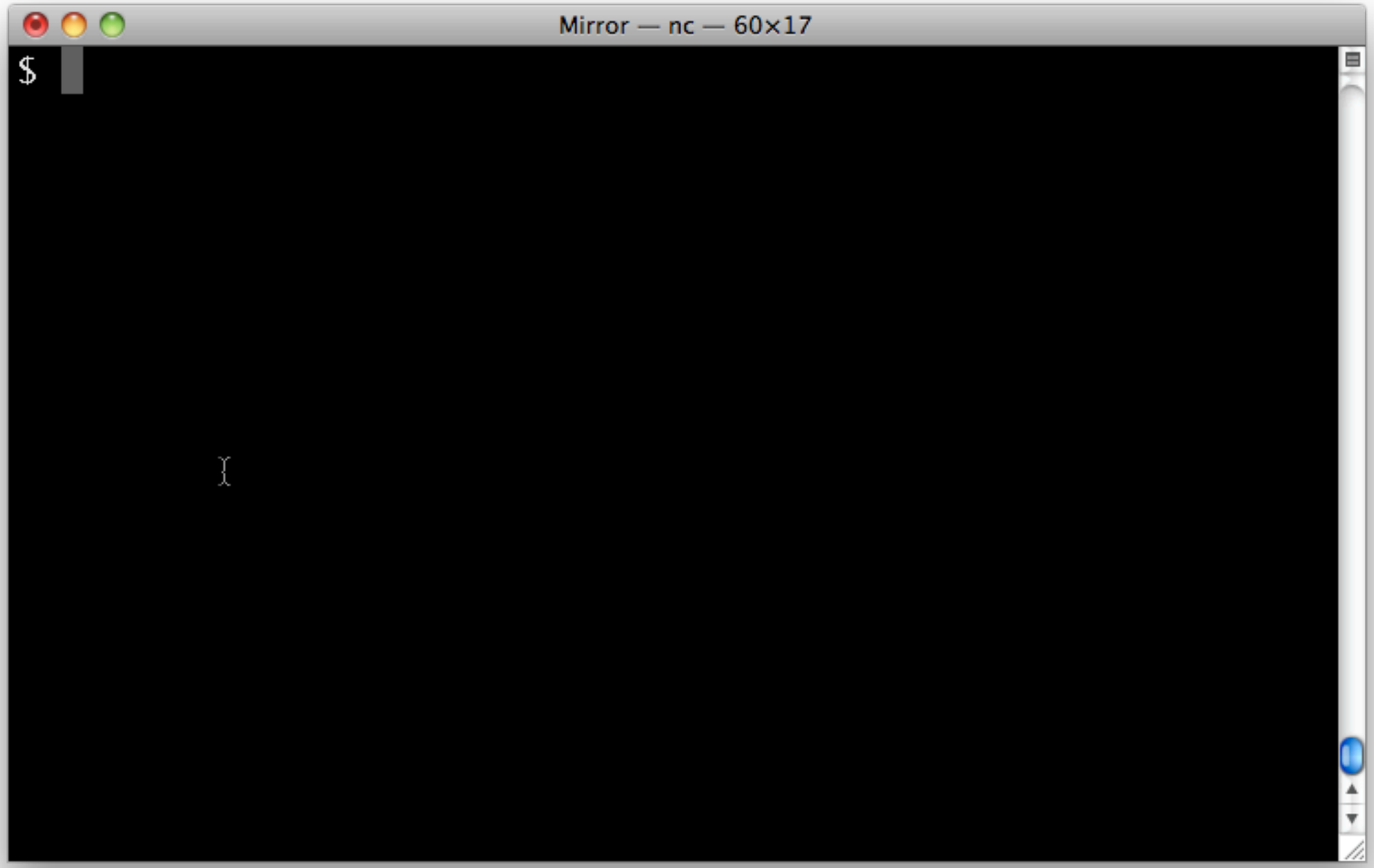
Monday, September 26, 2011

Some times you can get the catalogs days early when doing a large mirror update. Workarounds shrink, but do not eliminate the window

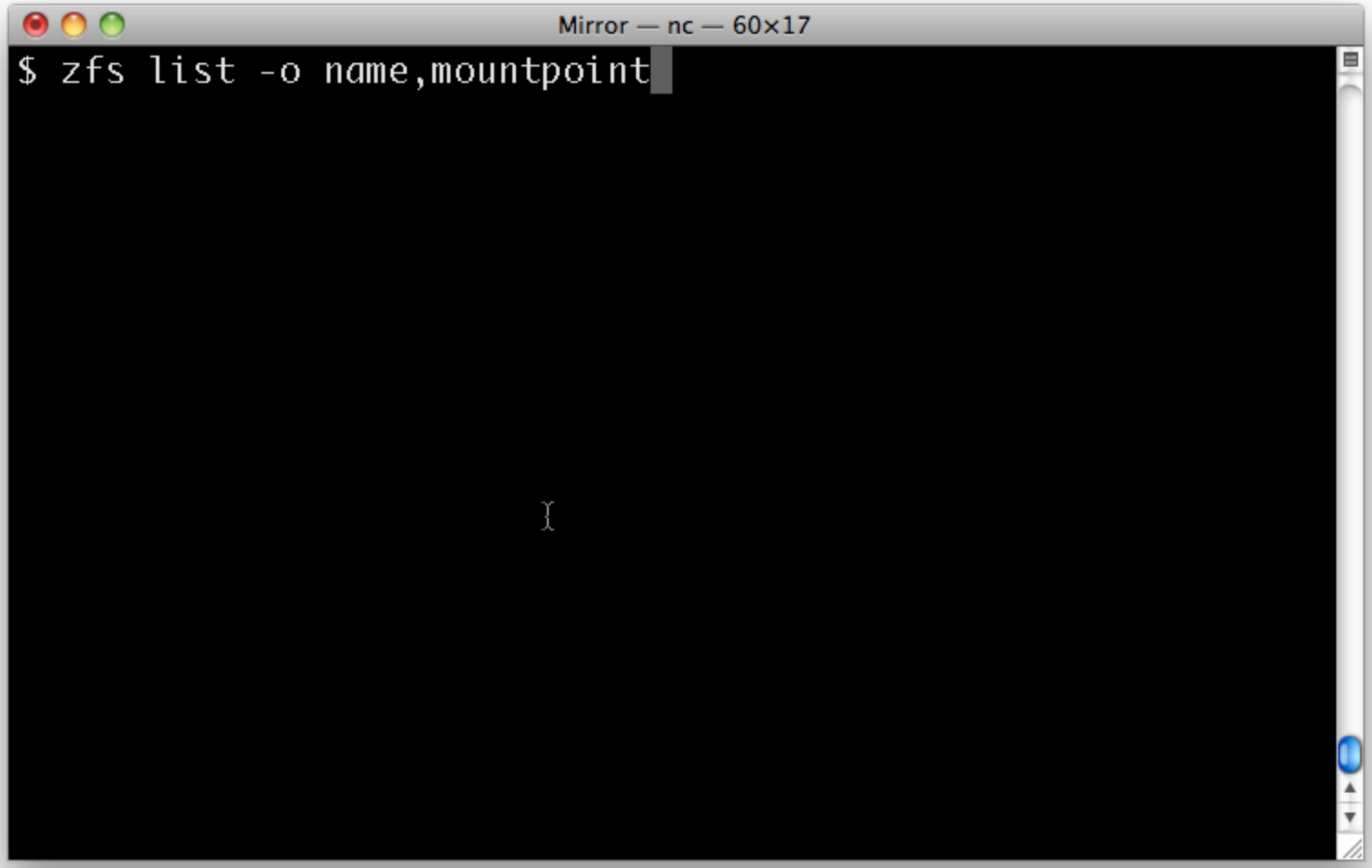
ZFS cloned snapshots

AEROSPACE

An alternative solution is ... and updating the snapshot



AEROSPACE



AEROSPACE

Mirror — nc — 60×17

```
$ zfs list -o name,mountpoint
NAME                            MOUNTPOINT
sites                           /sites
sites/DragonFly-2011-02-26      /sites/DragonFly-2011-02-26
sites/FreeBSD-2011-02-26        /sites/FreeBSD-2011-02-26
sites/NetBSD-2011-02-26         /sites/NetBSD-2011-02-26
sites/OpenBSD-2011-02-26        /sites/OpenBSD-2011-02-26
$
```

AEROSPACE

Mirror — nc — 60×17

```
$ zfs list -o name,mountpoint
NAME                              MOUNTPOINT
sites                             /sites
sites/DragonFly-2011-02-26        /sites/DragonFly-2011-02-26
sites/FreeBSD-2011-02-26          /sites/FreeBSD-2011-02-26
sites/NetBSD-2011-02-26           /sites/NetBSD-2011-02-26
sites/OpenBSD-2011-02-26          /sites/OpenBSD-2011-02-26
$ ls -l /sites/FreeBSD
```

AEROSPACE

Mirror — nc — 60×17

```
$ zfs list -o name,mountpoint
NAME                            MOUNTPOINT
sites                           /sites
sites/DragonFly-2011-02-26      /sites/DragonFly-2011-02-26
sites/FreeBSD-2011-02-26        /sites/FreeBSD-2011-02-26
sites/NetBSD-2011-02-26         /sites/NetBSD-2011-02-26
sites/OpenBSD-2011-02-26        /sites/OpenBSD-2011-02-26
$ ls -l /sites/FreeBSD
lrwxr-xr-x  1 root  wheel  18 Feb 28 00:24 /sites/FreeBSD ->
 FreeBSD-2011-02-26
$
```

AEROSPACE

Mirror — nc — 60×17

```
$ zfs list -o name,mountpoint
NAME                            MOUNTPOINT
sites                           /sites
sites/DragonFly-2011-02-26      /sites/DragonFly-2011-02-26
sites/FreeBSD-2011-02-26        /sites/FreeBSD-2011-02-26
sites/NetBSD-2011-02-26         /sites/NetBSD-2011-02-26
sites/OpenBSD-2011-02-26        /sites/OpenBSD-2011-02-26
$ ls -l /sites/FreeBSD
lrwxr-xr-x  1 root  wheel  18 Feb 28 00:24 /sites/FreeBSD ->
 FreeBSD-2011-02-26
$ zfs snapshot sites/FreeBSD-2011-02-26@updating
```

AEROSPACE

Mirror — nc — 60×17

```
$ zfs list -o name,mountpoint
NAME                            MOUNTPOINT
sites                           /sites
sites/DragonFly-2011-02-26      /sites/DragonFly-2011-02-26
sites/FreeBSD-2011-02-26        /sites/FreeBSD-2011-02-26
sites/NetBSD-2011-02-26         /sites/NetBSD-2011-02-26
sites/OpenBSD-2011-02-26        /sites/OpenBSD-2011-02-26
$ ls -l /sites/FreeBSD
lrwxr-xr-x  1 root  wheel  18 Feb 28 00:24 /sites/FreeBSD ->
 FreeBSD-2011-02-26
$ zfs snapshot sites/FreeBSD-2011-02-26@updating
$
```

AEROSPACE

Mirror — nc — 60×17

```
$ zfs list -o name,mountpoint
NAME                            MOUNTPOINT
sites                           /sites
sites/DragonFly-2011-02-26      /sites/DragonFly-2011-02-26
sites/FreeBSD-2011-02-26        /sites/FreeBSD-2011-02-26
sites/NetBSD-2011-02-26         /sites/NetBSD-2011-02-26
sites/OpenBSD-2011-02-26        /sites/OpenBSD-2011-02-26
$ ls -l /sites/FreeBSD
lrwxr-xr-x  1 root  wheel  18 Feb 28 00:24 /sites/FreeBSD ->
 FreeBSD-2011-02-26
$ zfs snapshot sites/FreeBSD-2011-02-26@updating
$ zfs clone sites/FreeBSD-2011-02-26@updating sites/FreeBSD-
2011-02-27
```

AEROSPACE

Mirror — nc — 60×17

```
$ zfs list -o name,mountpoint
NAME                            MOUNTPOINT
sites                           /sites
sites/DragonFly-2011-02-26      /sites/DragonFly-2011-02-26
sites/FreeBSD-2011-02-26        /sites/FreeBSD-2011-02-26
sites/NetBSD-2011-02-26         /sites/NetBSD-2011-02-26
sites/OpenBSD-2011-02-26        /sites/OpenBSD-2011-02-26
$ ls -l /sites/FreeBSD
lrwxr-xr-x  1 root  wheel  18 Feb 28 00:24 /sites/FreeBSD ->
 FreeBSD-2011-02-26
$ zfs snapshot sites/FreeBSD-2011-02-26@updating
$ zfs clone sites/FreeBSD-2011-02-26@updating sites/FreeBSD-
2011-02-27
$
```

AEROSPACE

Update the FreeBSD-2011-02-27 filesystem until consistent

AEROSPACE

Mirror — nc — 60×17

```
$ zfs list -o name,mountpoint
NAME                            MOUNTPOINT
sites                           /sites
sites/DragonFly-2011-02-26      /sites/DragonFly-2011-02-26
sites/FreeBSD-2011-02-26        /sites/FreeBSD-2011-02-26
sites/NetBSD-2011-02-26         /sites/NetBSD-2011-02-26
sites/OpenBSD-2011-02-26        /sites/OpenBSD-2011-02-26
$ ls -l /sites/FreeBSD
lrwxr-xr-x  1 root  wheel  18 Feb 28 00:24 /sites/FreeBSD ->
 FreeBSD-2011-02-26
$ zfs snapshot sites/FreeBSD-2011-02-26@updating
$ zfs clone sites/FreeBSD-2011-02-26@updating sites/FreeBSD-
2011-02-27
$
$ zfs promote sites/FreeBSD-2011-02-27
```

AEROSPACE

Mirror — nc — 60×17

```
$ zfs list -o name,mountpoint
NAME                            MOUNTPOINT
sites                           /sites
sites/DragonFly-2011-02-26      /sites/DragonFly-2011-02-26
sites/FreeBSD-2011-02-26        /sites/FreeBSD-2011-02-26
sites/NetBSD-2011-02-26         /sites/NetBSD-2011-02-26
sites/OpenBSD-2011-02-26        /sites/OpenBSD-2011-02-26
$ ls -l /sites/FreeBSD
lrwxr-xr-x  1 root  wheel  18 Feb 28 00:24 /sites/FreeBSD ->
 FreeBSD-2011-02-26
$ zfs snapshot sites/FreeBSD-2011-02-26@updating
$ zfs clone sites/FreeBSD-2011-02-26@updating sites/FreeBSD-
2011-02-27
$
$ zfs promote sites/FreeBSD-2011-02-27
$
```

AEROSPACE

Mirror — nc — 60×17

```
$ zfs list -o name,mountpoint
NAME                            MOUNTPOINT
sites                           /sites
sites/DragonFly-2011-02-26      /sites/DragonFly-2011-02-26
sites/FreeBSD-2011-02-26        /sites/FreeBSD-2011-02-26
sites/NetBSD-2011-02-26         /sites/NetBSD-2011-02-26
sites/OpenBSD-2011-02-26        /sites/OpenBSD-2011-02-26
$ ls -l /sites/FreeBSD
lrwxr-xr-x  1 root  wheel  18 Feb 28 00:24 /sites/FreeBSD ->
 FreeBSD-2011-02-26
$ zfs snapshot sites/FreeBSD-2011-02-26@updating
$ zfs clone sites/FreeBSD-2011-02-26@updating sites/FreeBSD-
2011-02-27
$
$ zfs promote sites/FreeBSD-2011-02-27
$ zfs list -o name,mountpoint -t all
```

Mirror — nc — 60×17

```
 FreeBSD-2011-02-26
$ zfs snapshot sites/FreeBSD-2011-02-26@updating
$ zfs clone sites/FreeBSD-2011-02-26@updating sites/FreeBSD-
2011-02-27
$
$ zfs promote sites/FreeBSD-2011-02-27
$ zfs list -o name,mountpoint -t all
NAME                                  MOUNTPOINT
sites                                 /sites
sites/DragonFly-2011-02-26            /sites/DragonFly-2011-02-
26
sites/FreeBSD-2011-02-26              /sites/FreeBSD-2011-02-26
sites/FreeBSD-2011-02-27              /sites/FreeBSD-2011-02-27
sites/FreeBSD-2011-02-27@updating     -
sites/NetBSD-2011-02-26               /sites/NetBSD-2011-02-26
sites/OpenBSD-2011-02-26              /sites/OpenBSD-2011-02-26
$
```

AEROSPACE

Mirror — nc — 60×17

```
 FreeBSD-2011-02-26
$ zfs snapshot sites/FreeBSD-2011-02-26@updating
$ zfs clone sites/FreeBSD-2011-02-26@updating sites/FreeBSD-
2011-02-27
$
$ zfs promote sites/FreeBSD-2011-02-27
$ zfs list -o name,mountpoint -t all
NAME                                  MOUNTPOINT
sites                                 /sites
sites/DragonFly-2011-02-26            /sites/DragonFly-2011-02-
26
sites/FreeBSD-2011-02-26              /sites/FreeBSD-2011-02-26
sites/FreeBSD-2011-02-27              /sites/FreeBSD-2011-02-27
sites/FreeBSD-2011-02-27@updating     -
sites/NetBSD-2011-02-26               /sites/NetBSD-2011-02-26
sites/OpenBSD-2011-02-26              /sites/OpenBSD-2011-02-26
$ ln -sf FreeBSD-2011-02-27 FreeBSD
```

AEROSPACE

Mirror — nc — 60×17

```
$ zfs snapshot sites/FreeBSD-2011-02-26@updating
$ zfs clone sites/FreeBSD-2011-02-26@updating sites/FreeBSD-
2011-02-27
$
$ zfs promote sites/FreeBSD-2011-02-27
$ zfs list -o name,mountpoint -t all
NAME                                  MOUNTPOINT
sites                                 /sites
sites/DragonFly-2011-02-26            /sites/DragonFly-2011-02-
26
sites/FreeBSD-2011-02-26              /sites/FreeBSD-2011-02-26
sites/FreeBSD-2011-02-27              /sites/FreeBSD-2011-02-27
sites/FreeBSD-2011-02-27@updating     -
sites/NetBSD-2011-02-26               /sites/NetBSD-2011-02-26
sites/OpenBSD-2011-02-26              /sites/OpenBSD-2011-02-26
$ ln -sf FreeBSD-2011-02-27 FreeBSD
$
```

AEROSPACE

Mirror — nc — 60×17

```
$ zfs snapshot sites/FreeBSD-2011-02-26@updating
$ zfs clone sites/FreeBSD-2011-02-26@updating sites/FreeBSD-
2011-02-27
$
$ zfs promote sites/FreeBSD-2011-02-27
$ zfs list -o name,mountpoint -t all
NAME                                  MOUNTPOINT
sites                                 /sites
sites/DragonFly-2011-02-26            /sites/DragonFly-2011-02-
26
sites/FreeBSD-2011-02-26              /sites/FreeBSD-2011-02-26
sites/FreeBSD-2011-02-27              /sites/FreeBSD-2011-02-27
sites/FreeBSD-2011-02-27@updating     -
sites/NetBSD-2011-02-26               /sites/NetBSD-2011-02-26
sites/OpenBSD-2011-02-26              /sites/OpenBSD-2011-02-26
$ ln -sf FreeBSD-2011-02-27 FreeBSD
$ zfs destroy sites/FreeBSD-2011-02-26
```

AEROSPACE

Mirror — nc — 60×17

```
$ zfs clone sites/FreeBSD-2011-02-26@updating sites/FreeBSD-
2011-02-27
$
$ zfs promote sites/FreeBSD-2011-02-27
$ zfs list -o name,mountpoint -t all
NAME                                    MOUNTPOINT
sites                                   /sites
sites/DragonFly-2011-02-26              /sites/DragonFly-2011-02-
26
sites/FreeBSD-2011-02-26                /sites/FreeBSD-2011-02-26
sites/FreeBSD-2011-02-27                /sites/FreeBSD-2011-02-27
sites/FreeBSD-2011-02-27@updating       -
sites/NetBSD-2011-02-26                 /sites/NetBSD-2011-02-26
sites/OpenBSD-2011-02-26                /sites/OpenBSD-2011-02-26
$ ln -sf FreeBSD-2011-02-27 FreeBSD
$ zfs destroy sites/FreeBSD-2011-02-26
$
```

AEROSPACE

Monday, September 26, 2011

This destroy may fail if the file system is busy.

Mirror — nc — 60×17

```
$ zfs clone sites/FreeBSD-2011-02-26@updating sites/FreeBSD-
2011-02-27
$
$ zfs promote sites/FreeBSD-2011-02-27
$ zfs list -o name,mountpoint -t all
NAME                                  MOUNTPOINT
sites                                 /sites
sites/DragonFly-2011-02-26            /sites/DragonFly-2011-02-
26
sites/FreeBSD-2011-02-26              /sites/FreeBSD-2011-02-26
sites/FreeBSD-2011-02-27              /sites/FreeBSD-2011-02-27
sites/FreeBSD-2011-02-27@updating     -
sites/NetBSD-2011-02-26               /sites/NetBSD-2011-02-26
sites/OpenBSD-2011-02-26              /sites/OpenBSD-2011-02-26
$ ln -sf FreeBSD-2011-02-27 FreeBSD
$ zfs destroy sites/FreeBSD-2011-02-26
$ zfs destroy sites/FreeBSD-2011-02-27@updating
```

AEROSPACE

This will also fail in that case so a script needs to handle this.

Mirror — nc — 60×17

```
2011-02-27
$
$ zfs promote sites/FreeBSD-2011-02-27
$ zfs list -o name,mountpoint -t all
NAME                                MOUNTPOINT
sites                               /sites
sites/DragonFly-2011-02-26          /sites/DragonFly-2011-02-
26
sites/FreeBSD-2011-02-26            /sites/FreeBSD-2011-02-26
sites/FreeBSD-2011-02-27            /sites/FreeBSD-2011-02-27
sites/FreeBSD-2011-02-27@updating   -
sites/NetBSD-2011-02-26             /sites/NetBSD-2011-02-26
sites/OpenBSD-2011-02-26            /sites/OpenBSD-2011-02-26
$ ln -sf FreeBSD-2011-02-27 FreeBSD
$ zfs destroy sites/FreeBSD-2011-02-26
$ zfs destroy sites/FreeBSD-2011-02-27@updating
$
```

AEROSPACE

Not currently mirroring to ZFS due to unstable hardware

AEROSPACE

Aerosource and ZFS

AEROSPACE

Intro to Aerosource and ARANDA

AEROSPACE

What is Aerosouce?

AEROSPACE

Aerosource

AEROSPACE
Assuring Space Mission Success

Login | Help/Guide | About Trac | My Notifications | Contact Us | Preferences

Start Page | Index | History | Watch Page

Wiki
- Project List
- Project Library
- Request Project
- Request Account
- Join Project
- Timeline
- Roadmap
- Browse Source
- View Tickets
- New Ticket
- Search
- Commit Summary
- Blog

Last modified 6 months ago

Aerosource: Enterprise Source Software at Aerospace

Aerosource is a project management environment for Aerospace that allows the free sharing of ideas and source code within the Aerospace Corporation. Aerosource is built using the Trac project management software and utilizes the Subversion version control system for managing source code.

Getting Started

Create a new project

- Everything you need to know about Subversion
- Everything there is to know about Trac
- **Using your Blog**
- **Using Tags**
- AerosourceLinks contains links to tutorials and other helpful documentation.

- ProjectList contains all current Aerosource projects.
- VideoTutorials
- Try it out in our **Sandbox**
- Questions? **Contact us** via email at `aerosource-system@lists.aero.org`

Admin Links

- Aerosource Administrators

Request A Project

project-name Go

Commit Summary

24 Hour Commit Summary

31 Day Commit Summary

12 Month Commit Summary

4 Year Commit Summary

Project Growth

Sort Projects By:

Newest
Most Active
Most Recent Changes
Most Participants
Alphabetic

Download in other formats:
Plain Text | WikiToPdf

AEROSPACE

Monday, September 26, 2011

Our internal SourceForge.
Collaboration platform.
Open to all employees.
Based on Trac and Subversion.
Lots of custom management infrastructure.

ARANDA: Aerosource Restricted And NDA

AEROSPACE

Monday, September 26, 2011

Projects where pieces are subject to legal or security restrictions.
Not supported at all for many years, added recently to support continuity of operations and provide uniform access to tools.

About 300 Projects

AEROSPACE

90GB of Project Data

AEROSPACE

More on ZFS Properties

- Properties are attached to filesystems and volumes
- Property names containing ':' are reserved for user defined properties
- No other structure exists for property names

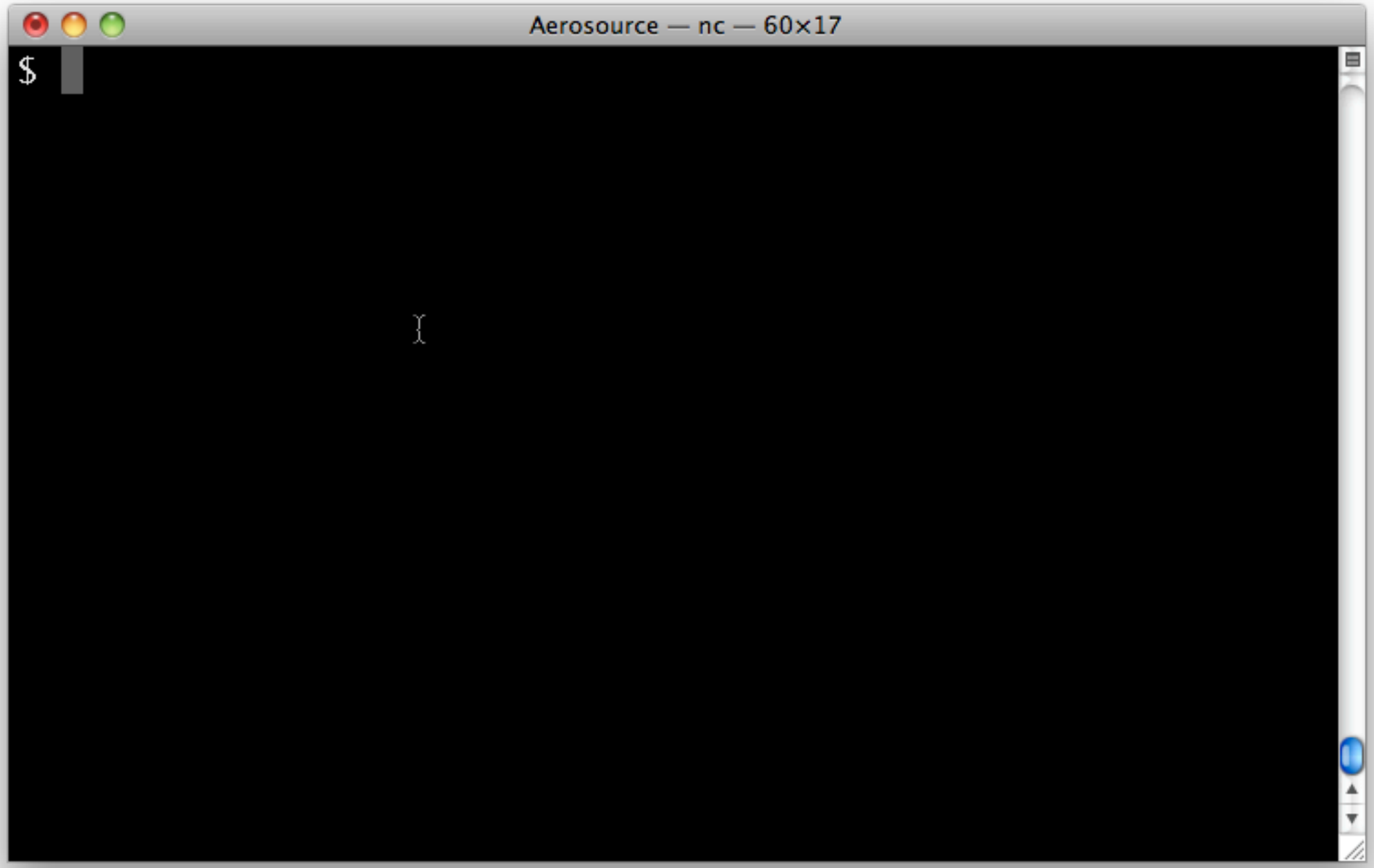
AEROSPACE

Problem:
Avoiding conflicting meanings for ZFS properties between scripts

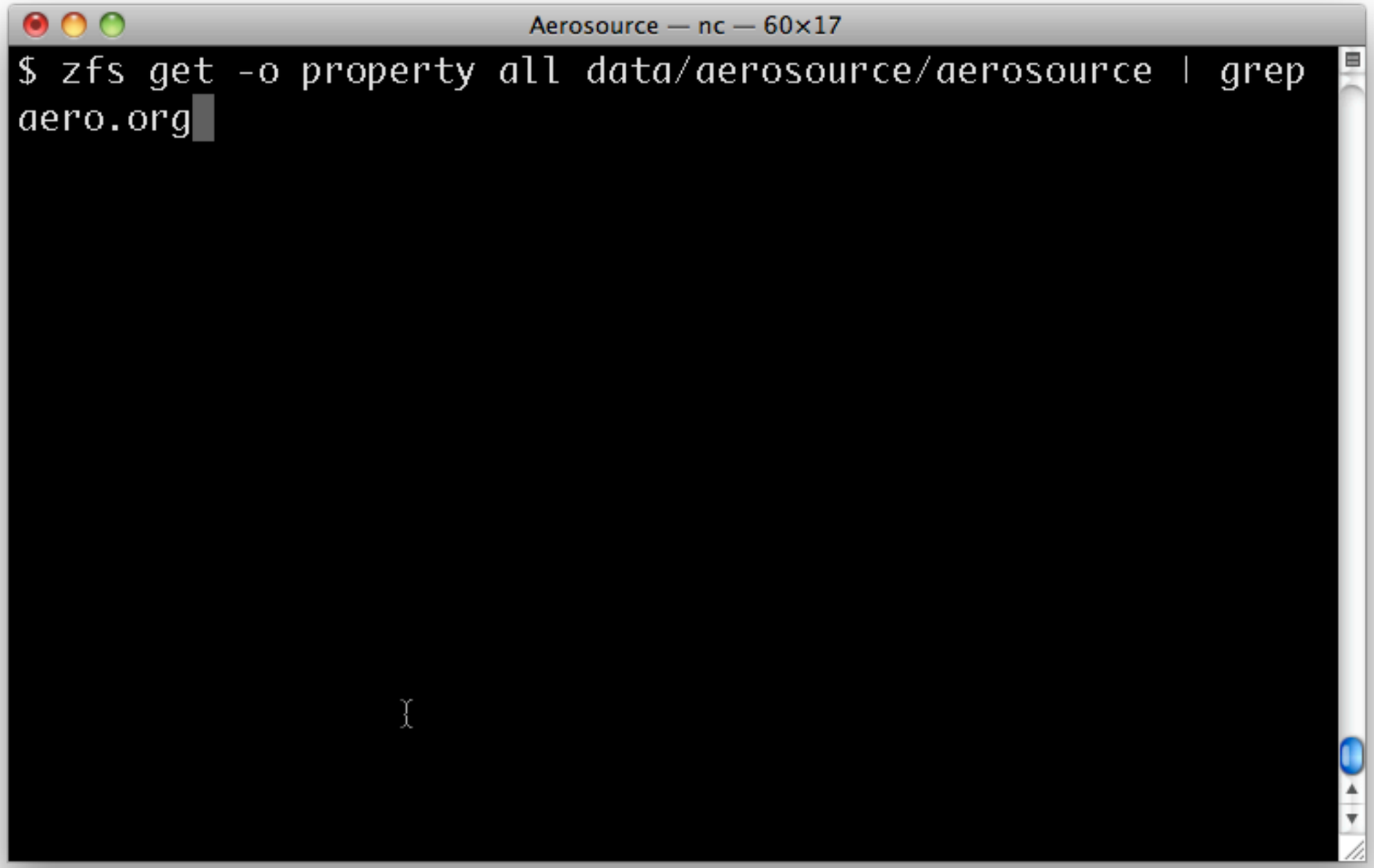
AEROSPACE

Solution:
Prepend domain name to property names

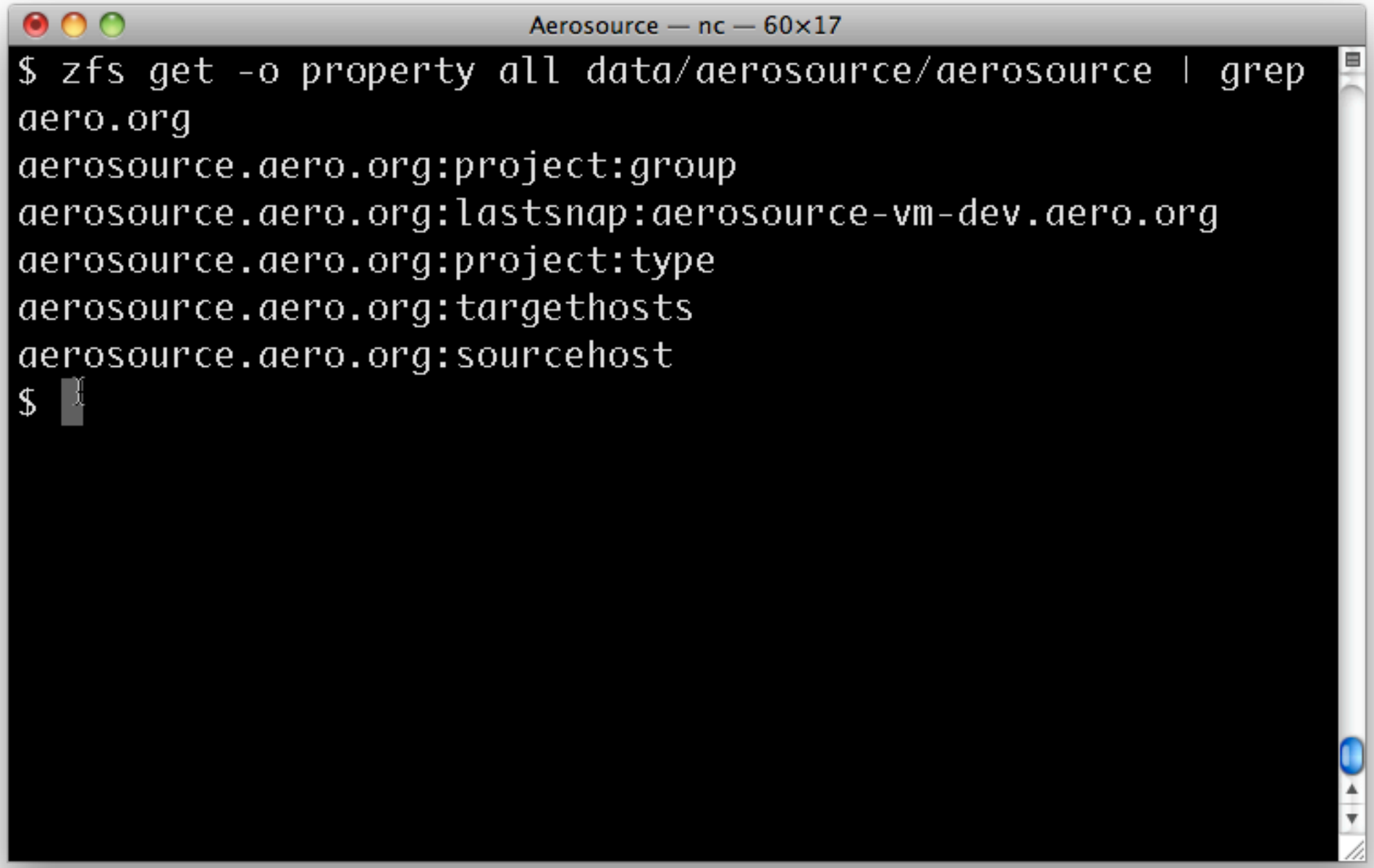
AEROSPACE



AEROSPACE


Aerosource — nc — 60×17
$ zfs get -o property all data/aerosource/aerosource | grep
aero.org

AEROSPACE


Aerosource — nc — 60×17
$ zfs get -o property all data/aerosource/aerosource | grep aero.org
aerosource.aero.org:project:group
aerosource.aero.org:lastsnap:aerosource-vm-dev.aero.org
aerosource.aero.org:project:type
aerosource.aero.org:targethosts
aerosource.aero.org:sourcehost
$

AEROSPACE

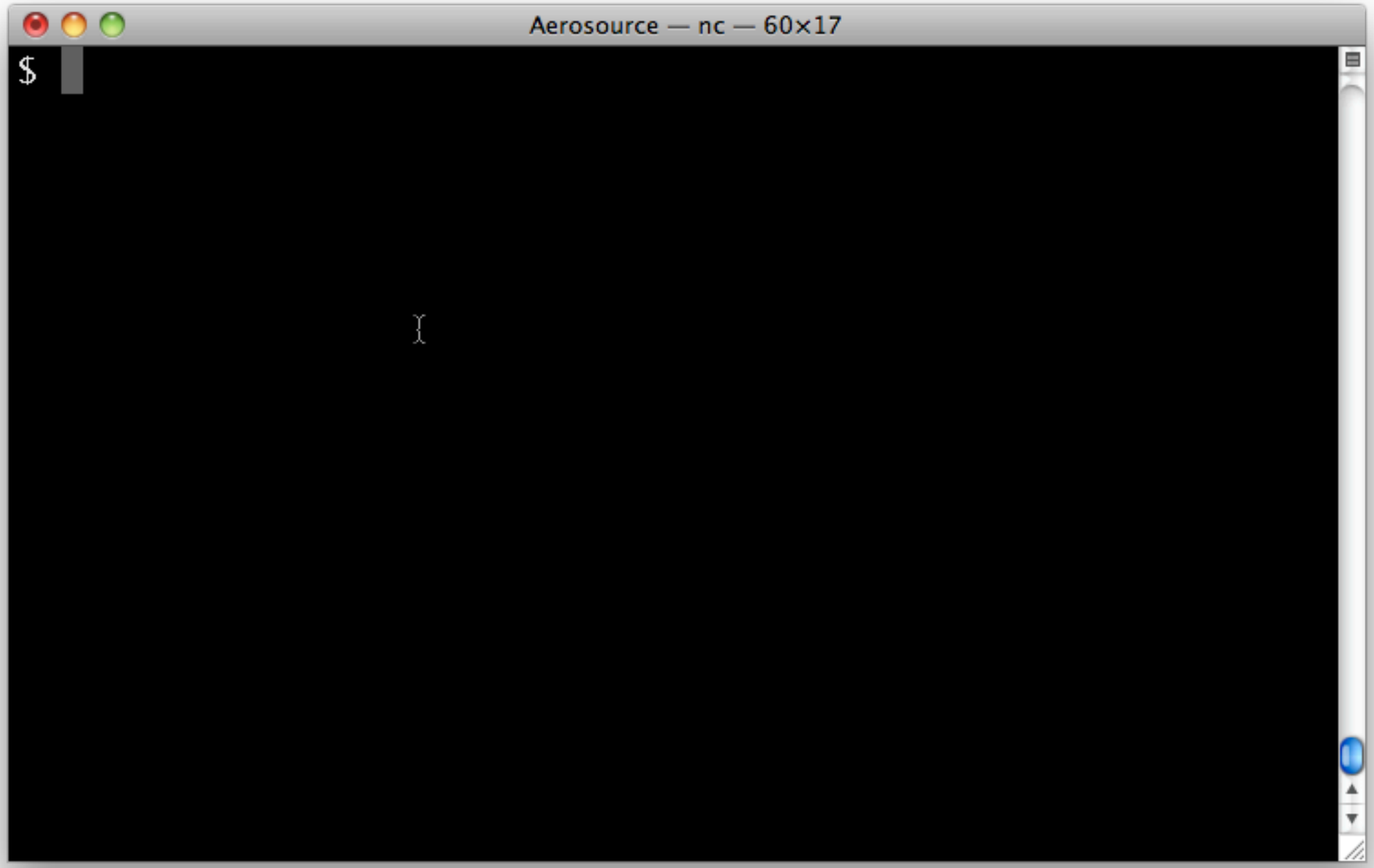
ZFS Properties for project meta-data

AEROSPACE

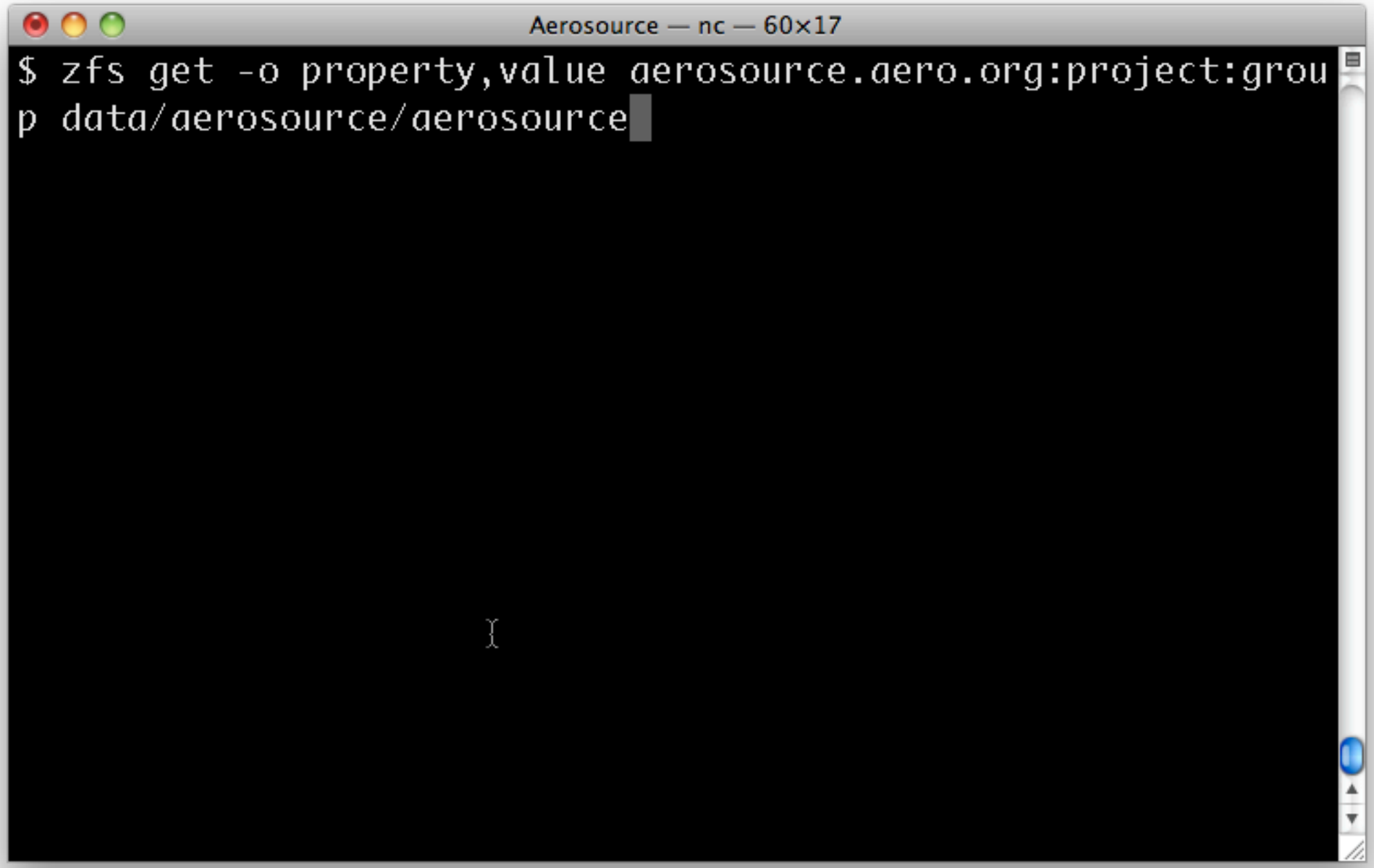
What to store?

- Project user and group
  - Only store the group (same as user)
- Project access control (Apache configuration)
  - Public, Auth Required, Private Group

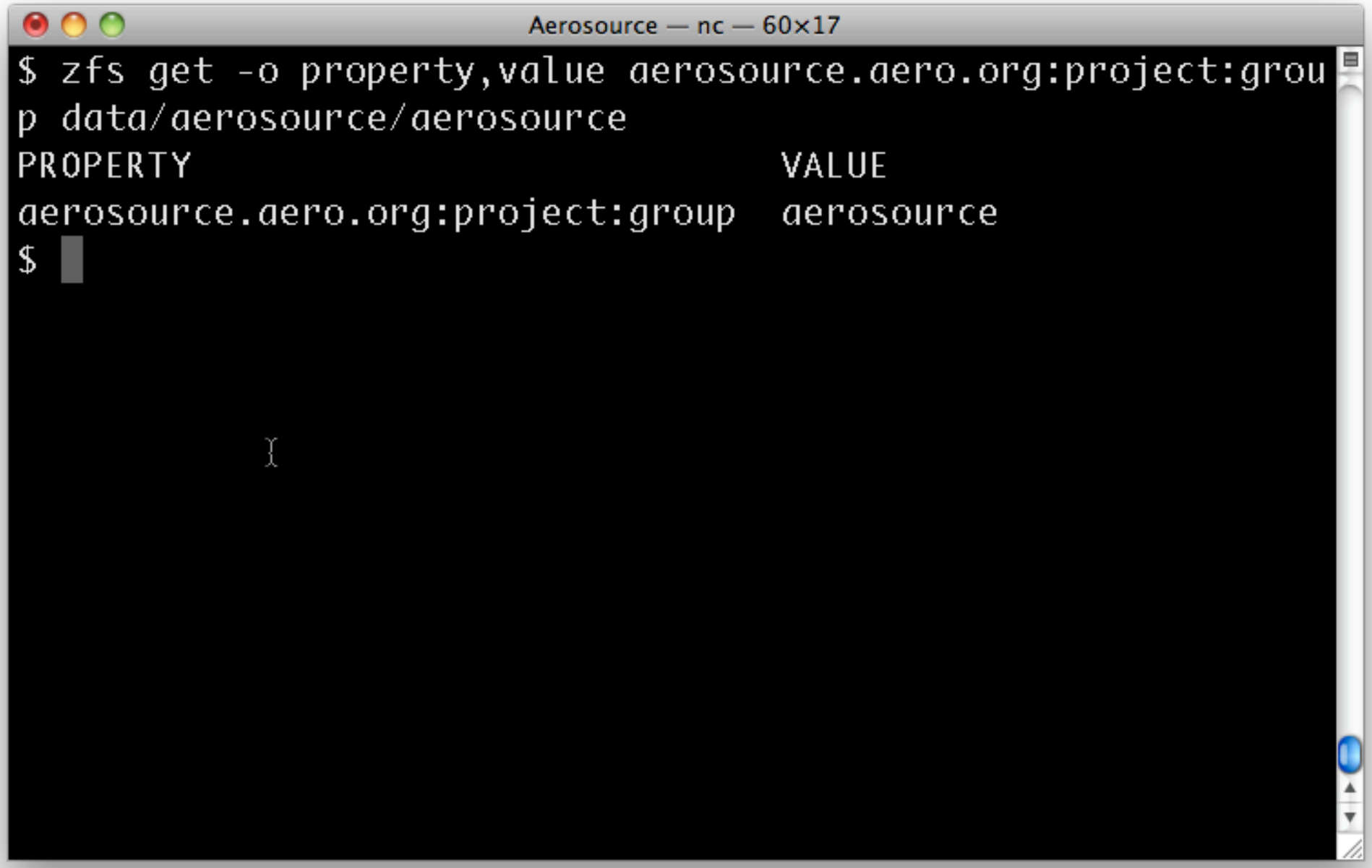
AEROSPACE



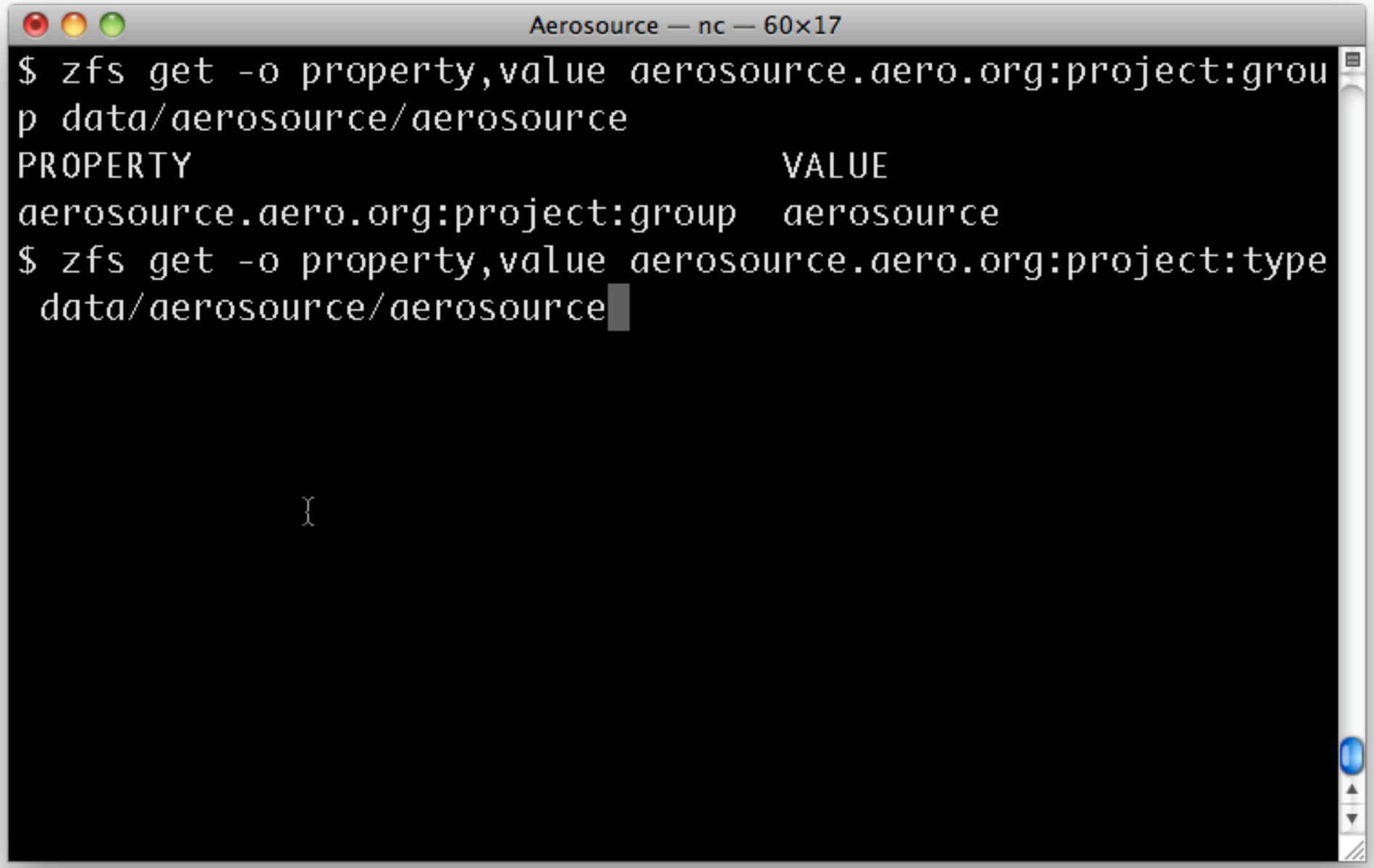
AEROSPACE


Aerosource — nc — 60×17
$ zfs get -o property,value aerosource.aero.org:project:grou
p data/aerosource/aerosource

AEROSPACE


Aerosource — nc — 60×17
$ zfs get -o property,value aerosource.aero.org:project:grou
p data/aerosource/aerosource
PROPERTY VALUE
aerosource.aero.org:project:group aerosource
$

AEROSPACE


Aerosource — nc — 60×17
$ zfs get -o property,value aerosource.aero.org:project:grou
p data/aerosource/aerosource
PROPERTY VALUE
aerosource.aero.org:project:group aerosource
$ zfs get -o property,value aerosource.aero.org:project:type
data/aerosource/aerosource

AEROSPACE

Aerosource — nc — 60×17

```
$ zfs get -o property,value aerosource.aero.org:project:grou
p data/aerosource/aerosource
PROPERTY                              VALUE
aerosource.aero.org:project:group     aerosource
$ zfs get -o property,value aerosource.aero.org:project:type
 data/aerosource/aerosource
PROPERTY                             VALUE
aerosource.aero.org:project:type     OldPublic
$
```

AEROSPACE

Properties drive scripts that create Apache config files

AEROSPACE

previously we had files containing lists of projects for each type of project.

Project Storage Replication

AEROSPACE

Differing Replication Needs

- Off-site warm spare
  - Alternate datacenter on East coast
  - Full replication
  - Efficient on slow WAN links
- Development server
  - Smaller system
  - Developers not authorized to see all project contents
  - Avoid using excessive disk space

Long term goal: multiple source hosts

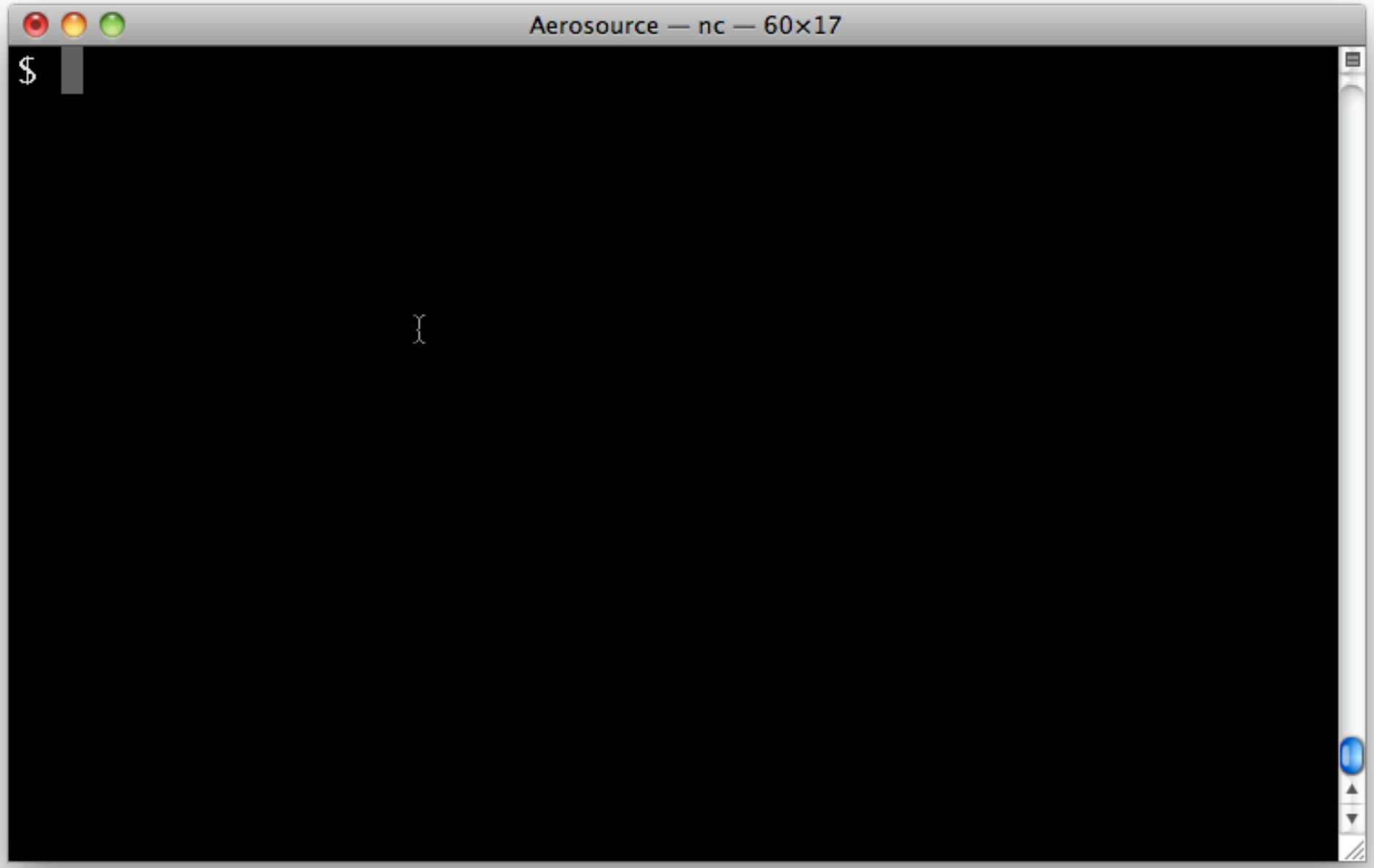
AEROSPACE

Additionally, we'd eventually like to support having projects have a home server so both servers are active and serving the projects that are at home on them.

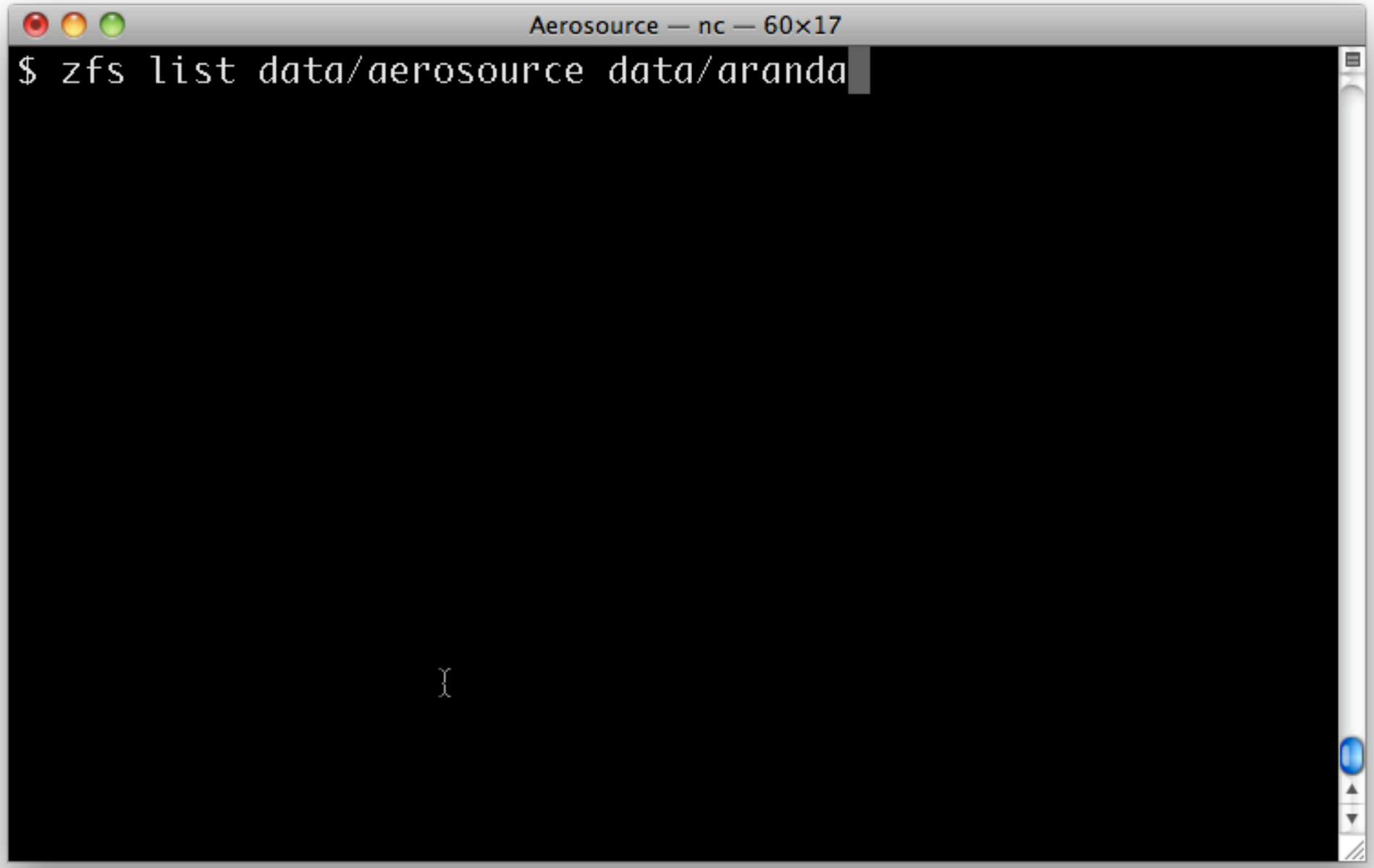
Want only one replication script

Aerosource storage layout

AEROSPACE



AEROSPACE



AEROSPACE

Aerosource — nc — 60×17

```
$ zfs list data/aerosource data/aranda
NAME                 USED  AVAIL  REFER  MOUNTPOINT
data/aerosource     71.6G  1.06T   841K  /aerosource
data/aranda         19.3G  1.06T    60K  /aranda
$
```

AEROSPACE

Properties supporting replication

- aerosource.aero.org:sourcehost => aerosource-west.aero.org
- aerosource.aero.org:targethost => aerosource-vm-dev.aero.org:data
- aerosource.aero.org:lastsnap:*targethost* => rep_1298838010

AEROSPACE

Monday, September 26, 2011

sourcehost: home system, always aerosource-west today
targethost: list of hosts and zfs datasets to send data to
lastsnap:targethost name of the last snapshot sent to a given host

Replication Cases

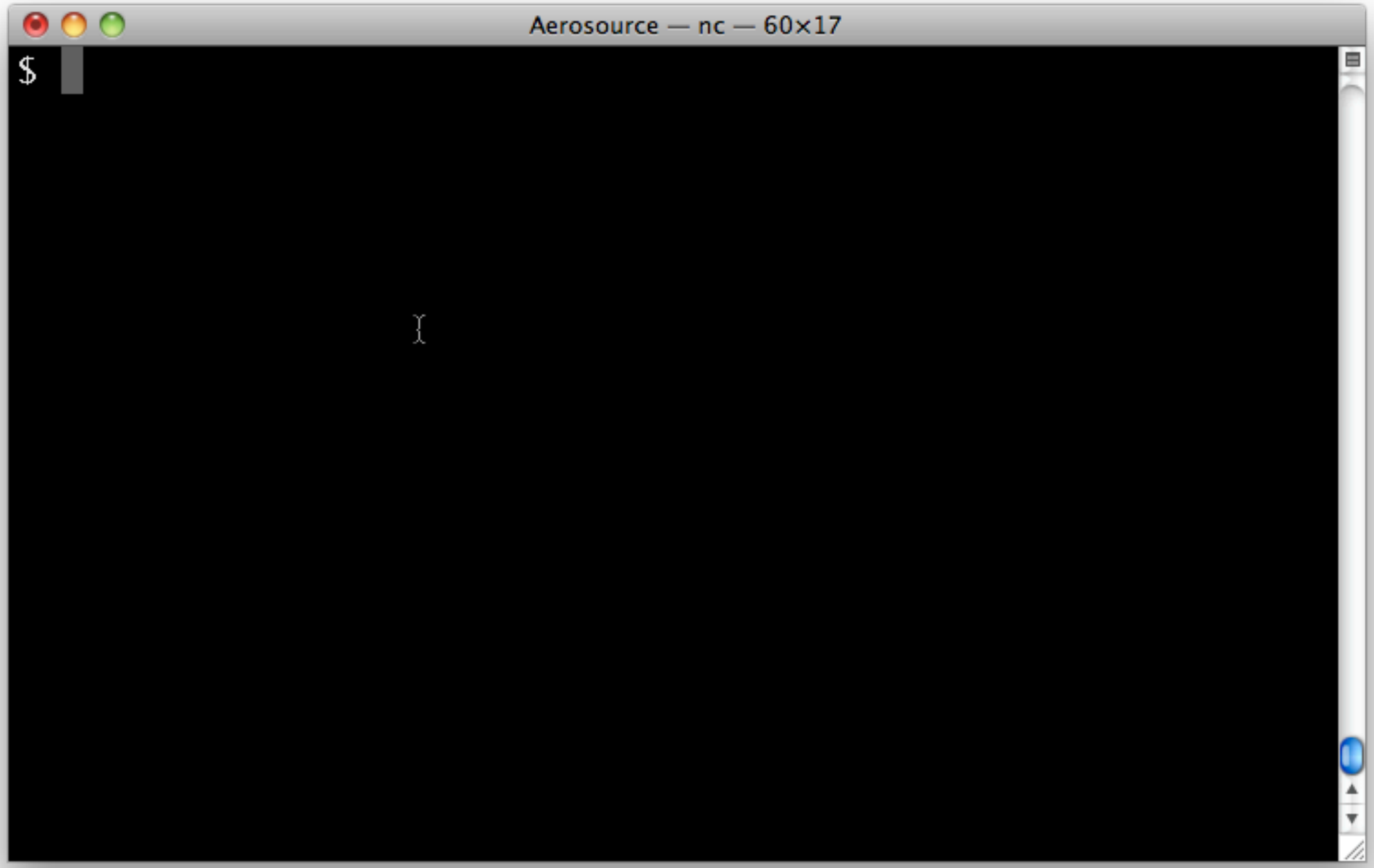
- Bootstrap: Project not transferred before
- Incremental: Transfer changes since last transferred snapshot

AEROSPACE

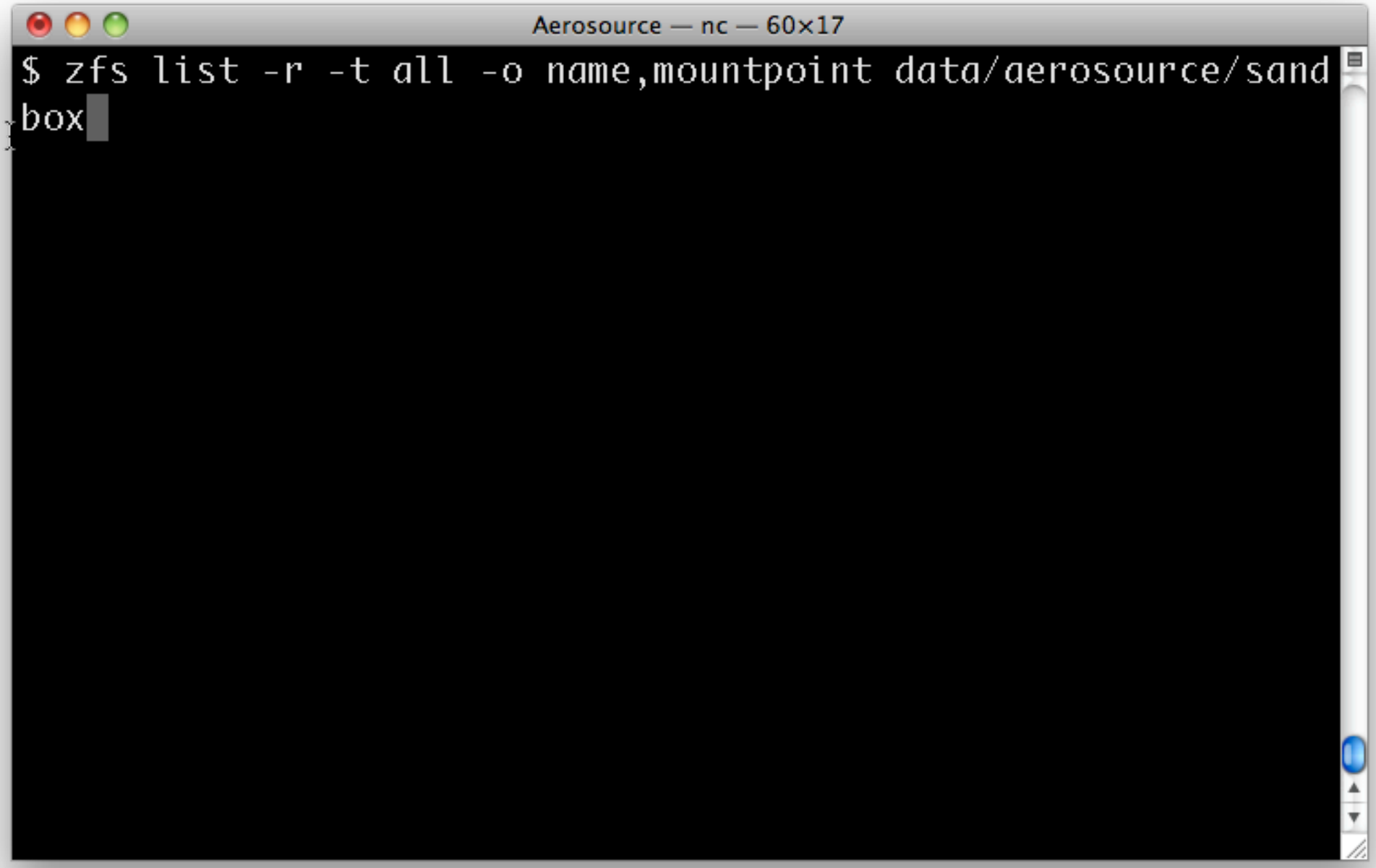
Case: Bootstrap

AEROSPACE

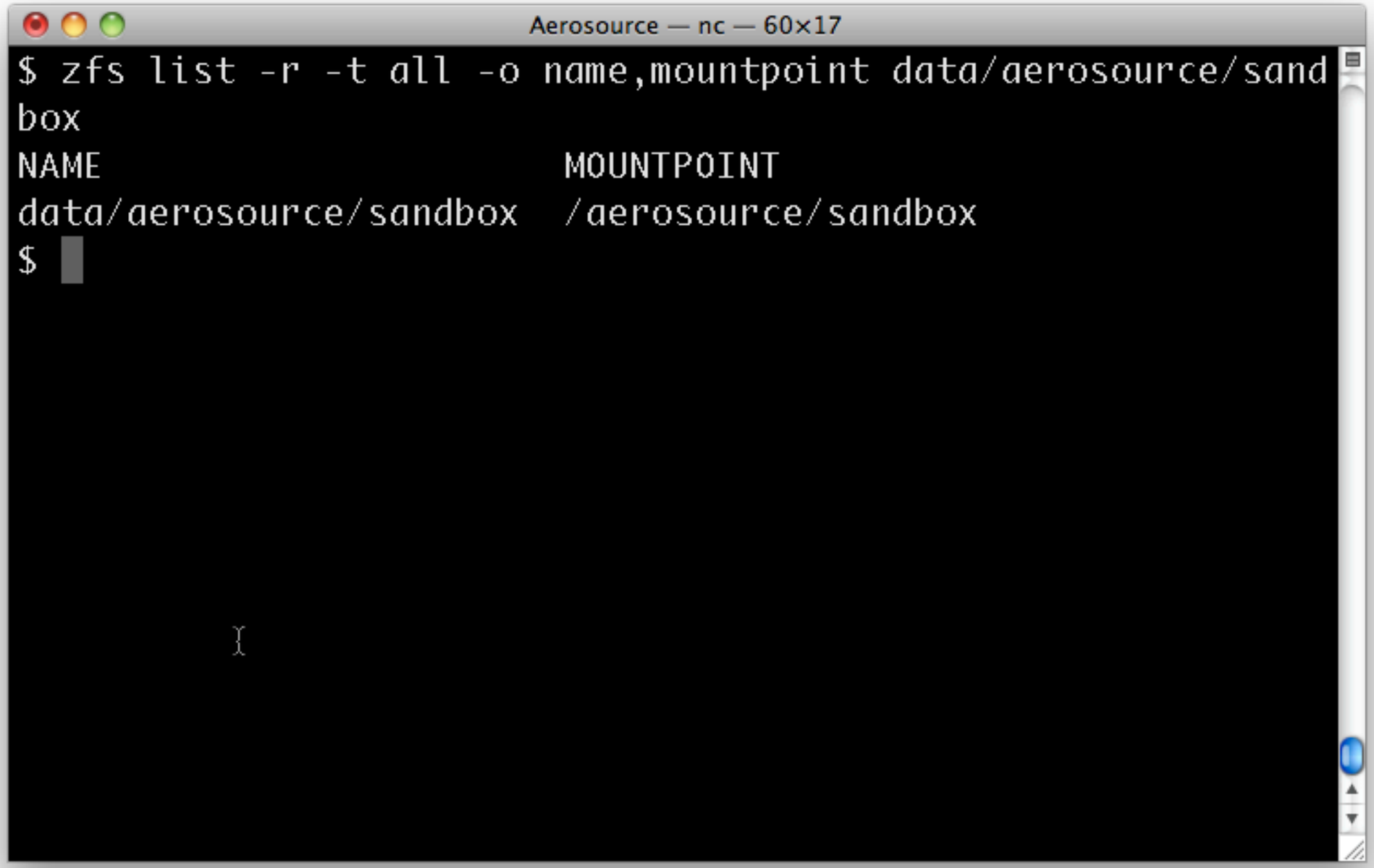
Bootstrap is the easiest case
Just make a snapshot and send it.
We have an option to destroy the file system on the other end if one exists.



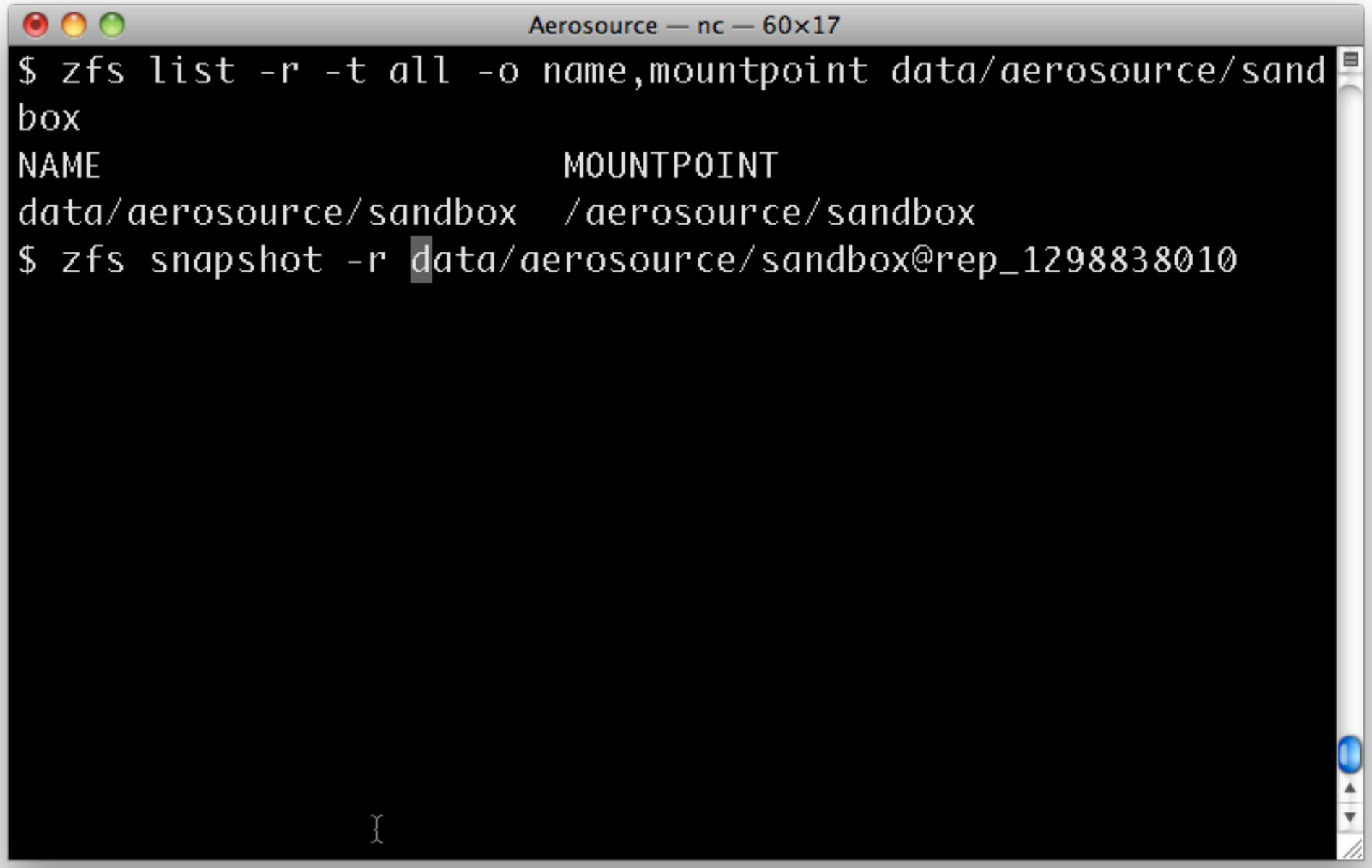
AEROSPACE


Aerosource — nc — 60×17
$ zfs list -r -t all -o name,mountpoint data/aerosource/sandbox

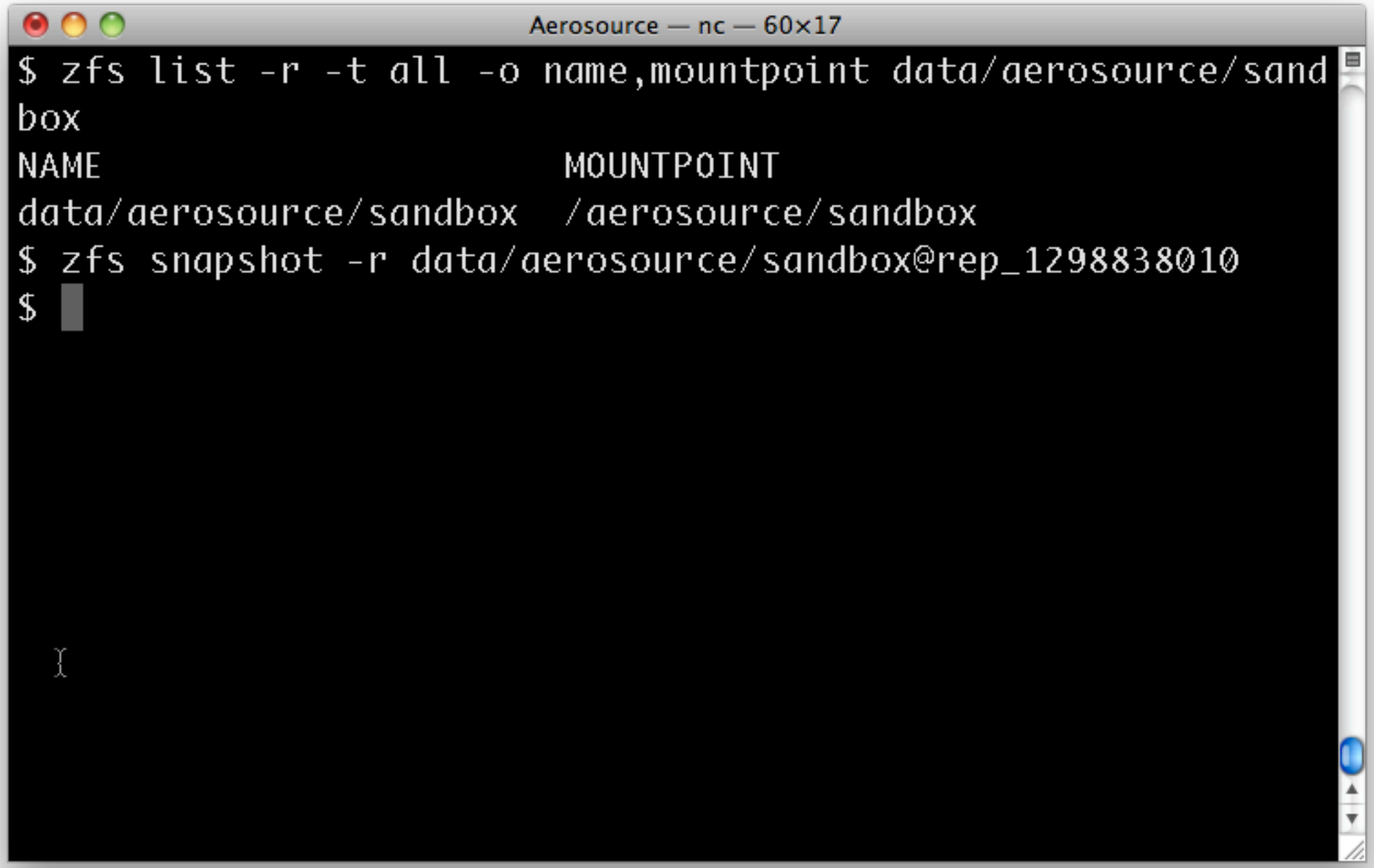
AEROSPACE


Aerosource — nc — 60×17
$ zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME   MOUNTPOINT
data/aerosource/sandbox   /aerosource/sandbox
$

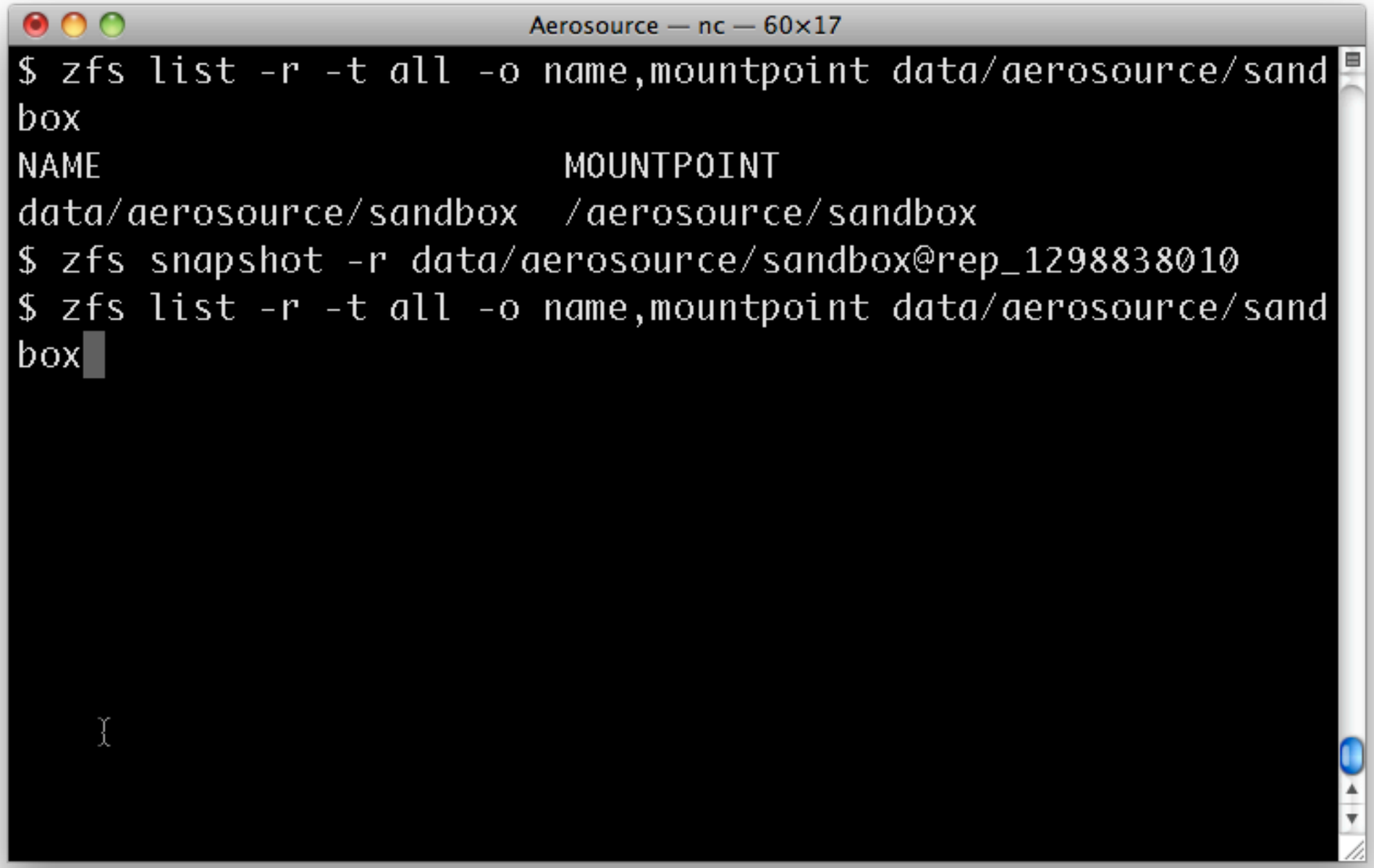
AEROSPACE


Aerosource — nc — 60×17
$ zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME   MOUNTPOINT
data/aerosource/sandbox   /aerosource/sandbox
$ zfs snapshot -r data/aerosource/sandbox@rep_1298838010

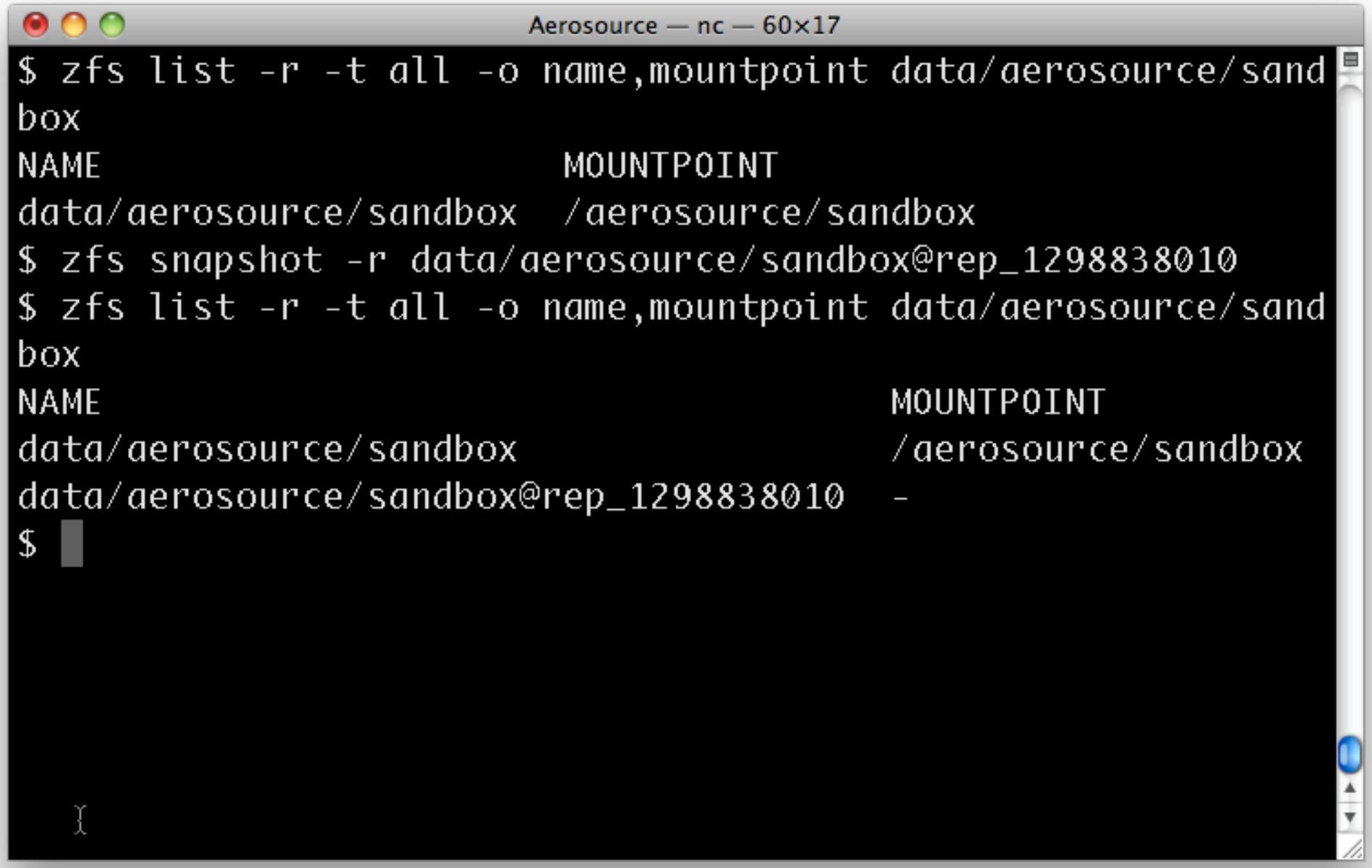
AEROSPACE


Aerosource — nc — 60×17
$ zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME   MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
$ zfs snapshot -r data/aerosource/sandbox@rep_1298838010
$

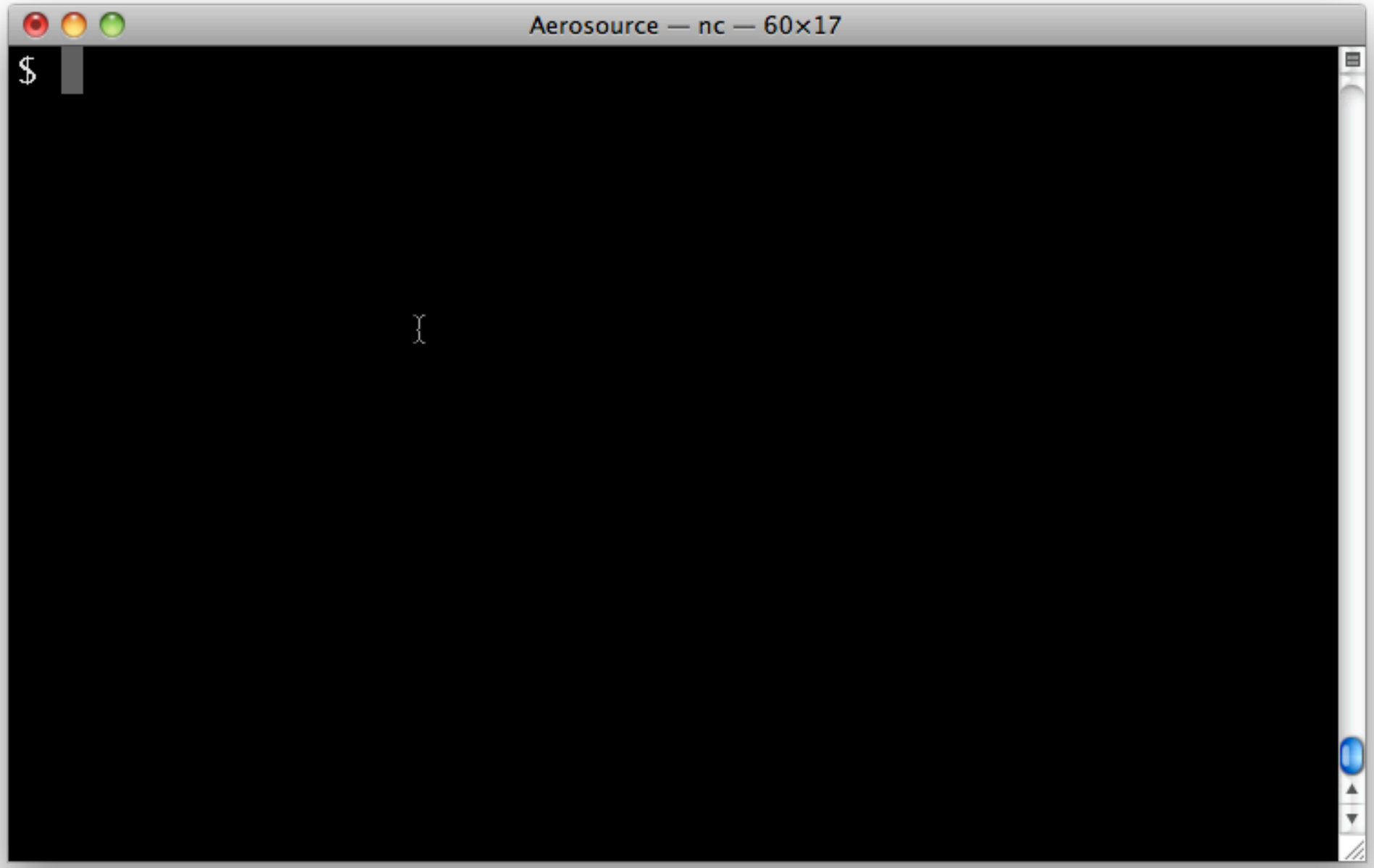
AEROSPACE


Aerosource — nc — 60×17
$ zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME   MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
$ zfs snapshot -r data/aerosource/sandbox@rep_1298838010
$ zfs list -r -t all -o name,mountpoint data/aerosource/sandbox

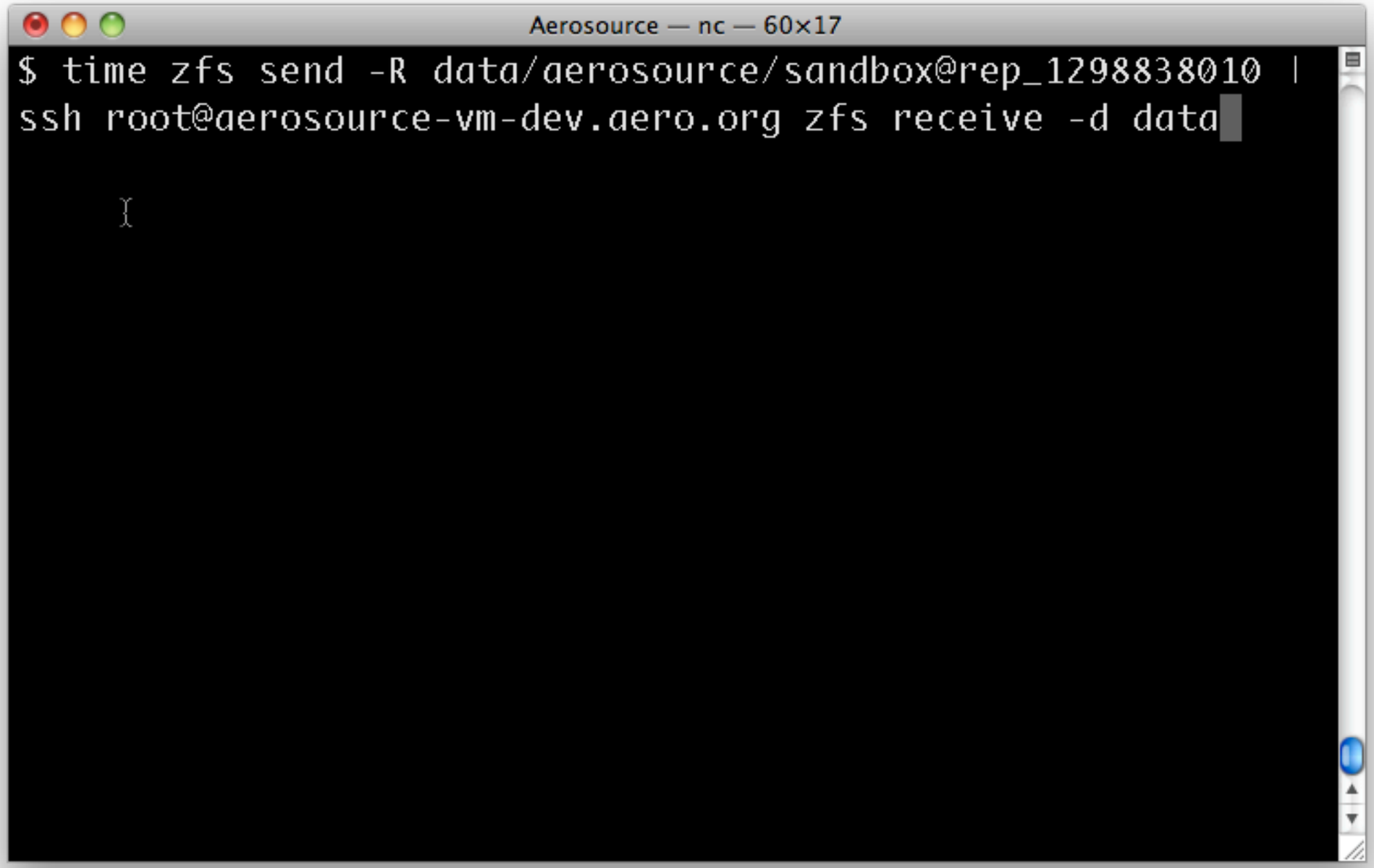
AEROSPACE


Aerosource — nc — 60×17
$ zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
$ zfs snapshot -r data/aerosource/sandbox@rep_1298838010
$ zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010 -
$

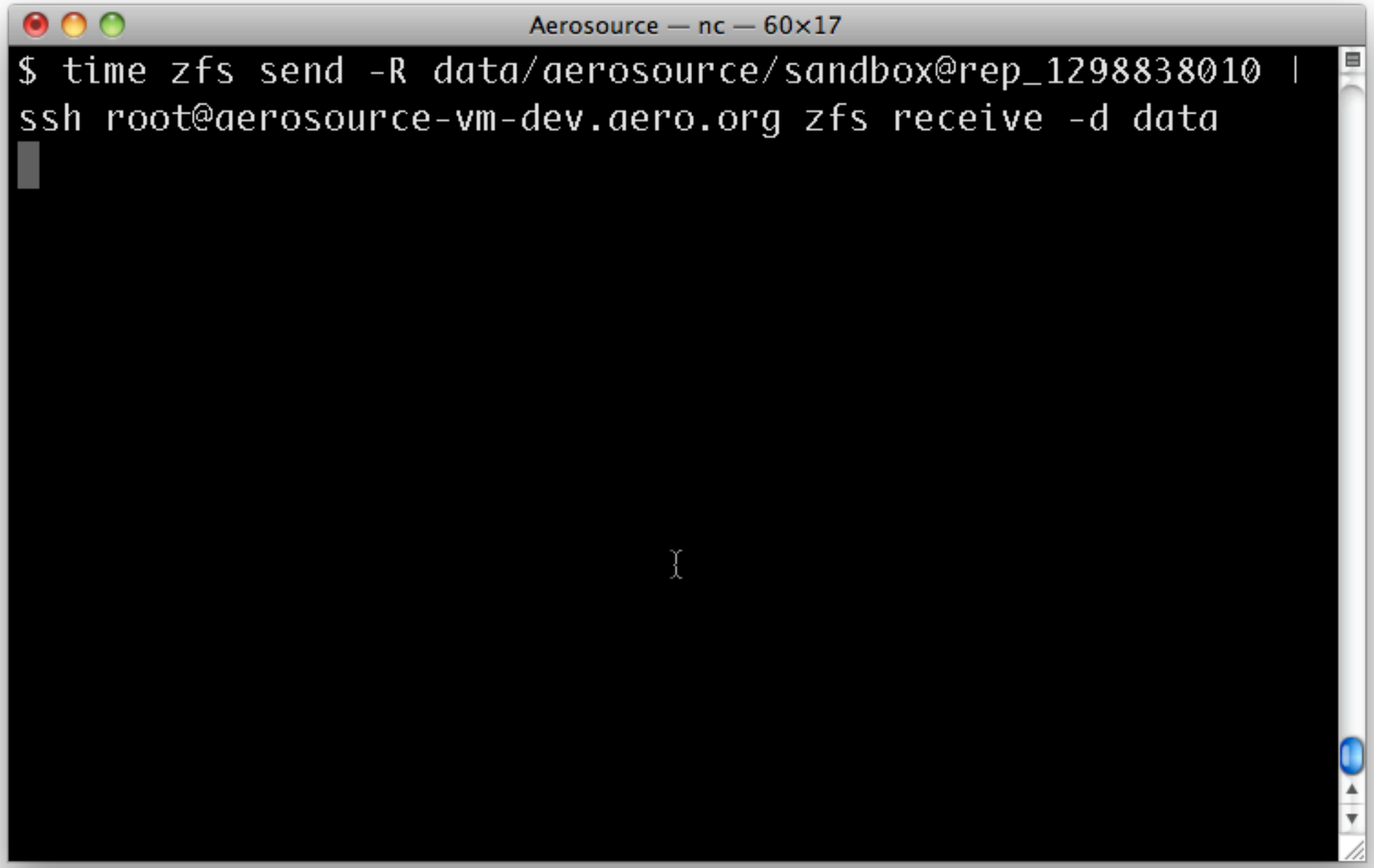
AEROSPACE



AEROSPACE


Aerosource — nc — 60×17
$ time zfs send -R data/aerosource/sandbox@rep_1298838010 |
ssh root@aerosource-vm-dev.aero.org zfs receive -d data

AEROSPACE


Aerosource — nc — 60×17
$ time zfs send -R data/aerosource/sandbox@rep_1298838010 |
ssh root@aerosource-vm-dev.aero.org zfs receive -d data

AEROSPACE

Aerosource — nc — 60×17

```
$ time zfs send -R data/aerosource/sandbox@rep_1298838010 |
ssh root@aerosource-vm-dev.aero.org zfs receive -d data

real    4m56.640s
user    0m56.602s
sys     0m16.322s
$
```

AEROSPACE

Aerosource — nc — 60×17

```
$ time zfs send -R data/aerosource/sandbox@rep_1298838010 |
ssh root@aerosource-vm-dev.aero.org zfs receive -d data

real    4m56.640s
user    0m56.602s
sys     0m16.322s
$ ssh root@aerosource-vm-dev.aero.org zfs list -r -t all -o
name,mountpoint data/aerosource/sandbox
```

AEROSPACE

Aerosource — nc — 60×17

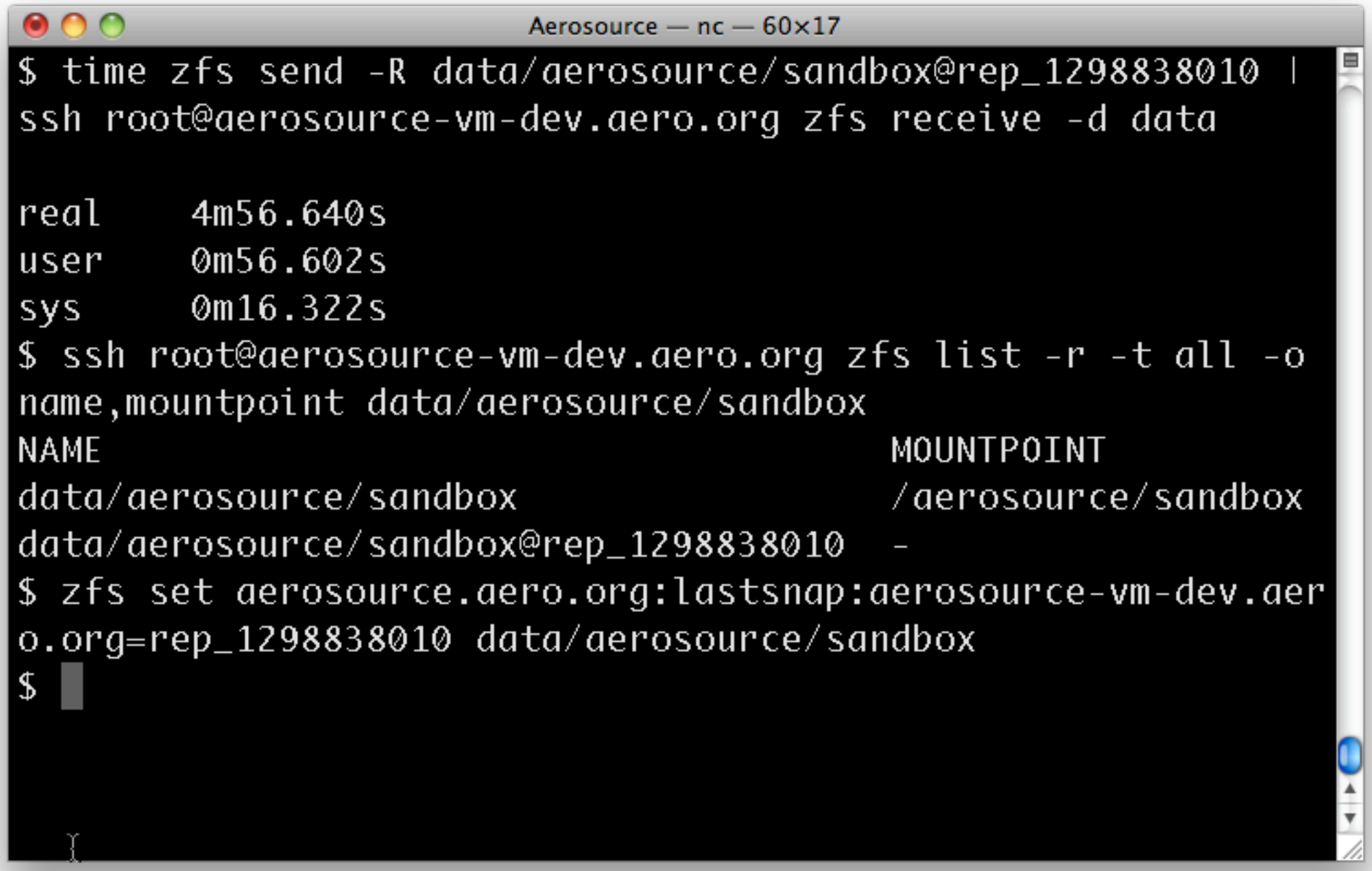
```
$ time zfs send -R data/aerosource/sandbox@rep_1298838010 |
ssh root@aerosource-vm-dev.aero.org zfs receive -d data

real    4m56.640s
user    0m56.602s
sys     0m16.322s
$ ssh root@aerosource-vm-dev.aero.org zfs list -r -t all -o
name,mountpoint data/aerosource/sandbox
NAME                                      MOUNTPOINT
data/aerosource/sandbox                   /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010    -
$
```

AEROSPACE

Aerosource — nc — 60×17

```
$ time zfs send -R data/aerosource/sandbox@rep_1298838010 |
ssh root@aerosource-vm-dev.aero.org zfs receive -d data

real    4m56.640s
user    0m56.602s
sys     0m16.322s
$ ssh root@aerosource-vm-dev.aero.org zfs list -r -t all -o
name,mountpoint data/aerosource/sandbox
NAME                                      MOUNTPOINT
data/aerosource/sandbox                   /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010    -
$ zfs set aerosource.aero.org:lastsnap:aerosource-vm-dev.aer
o.org=rep_1298838010 data/aerosource/sandbox
```

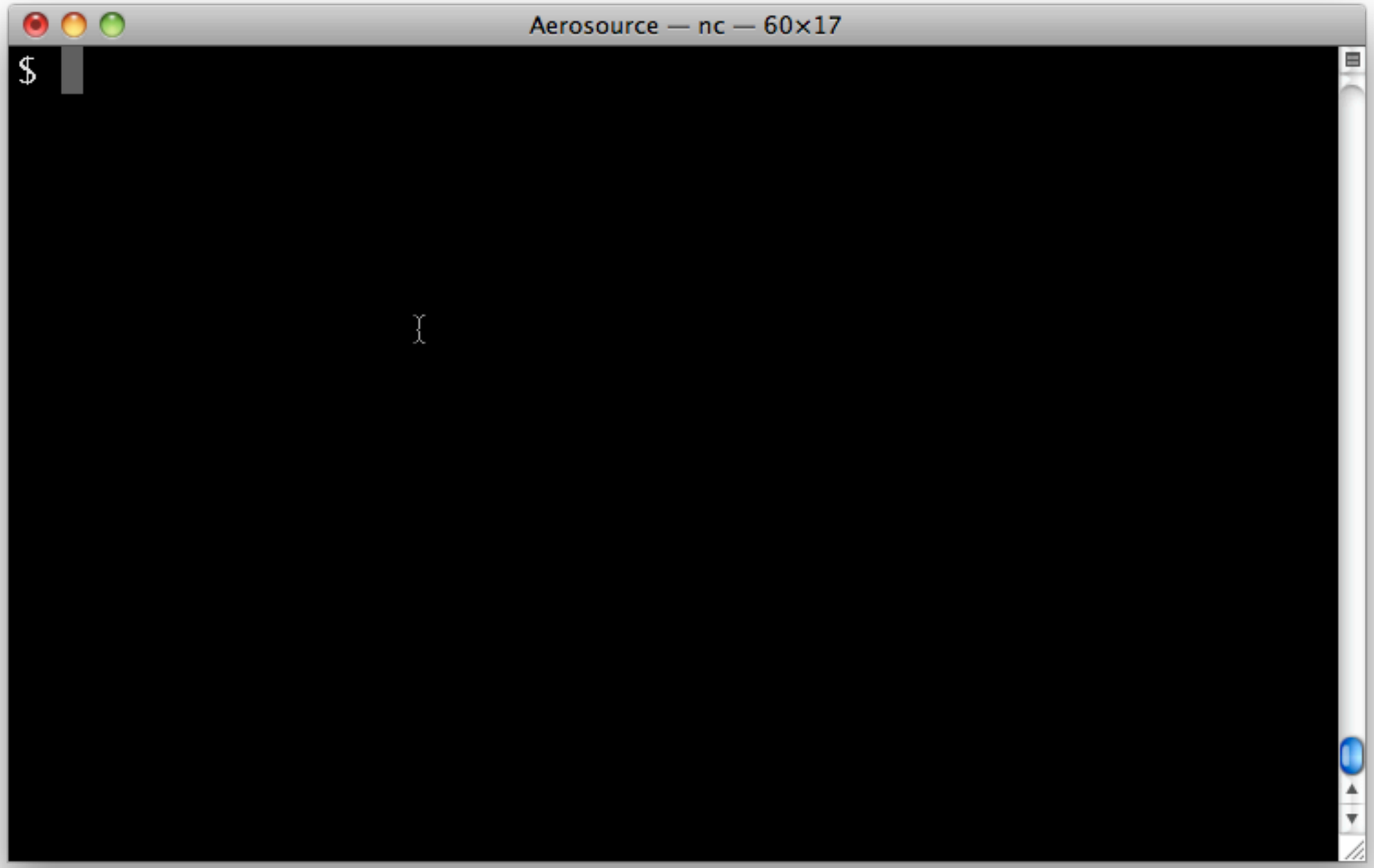
AEROSPACE



AEROSPACE

that is that

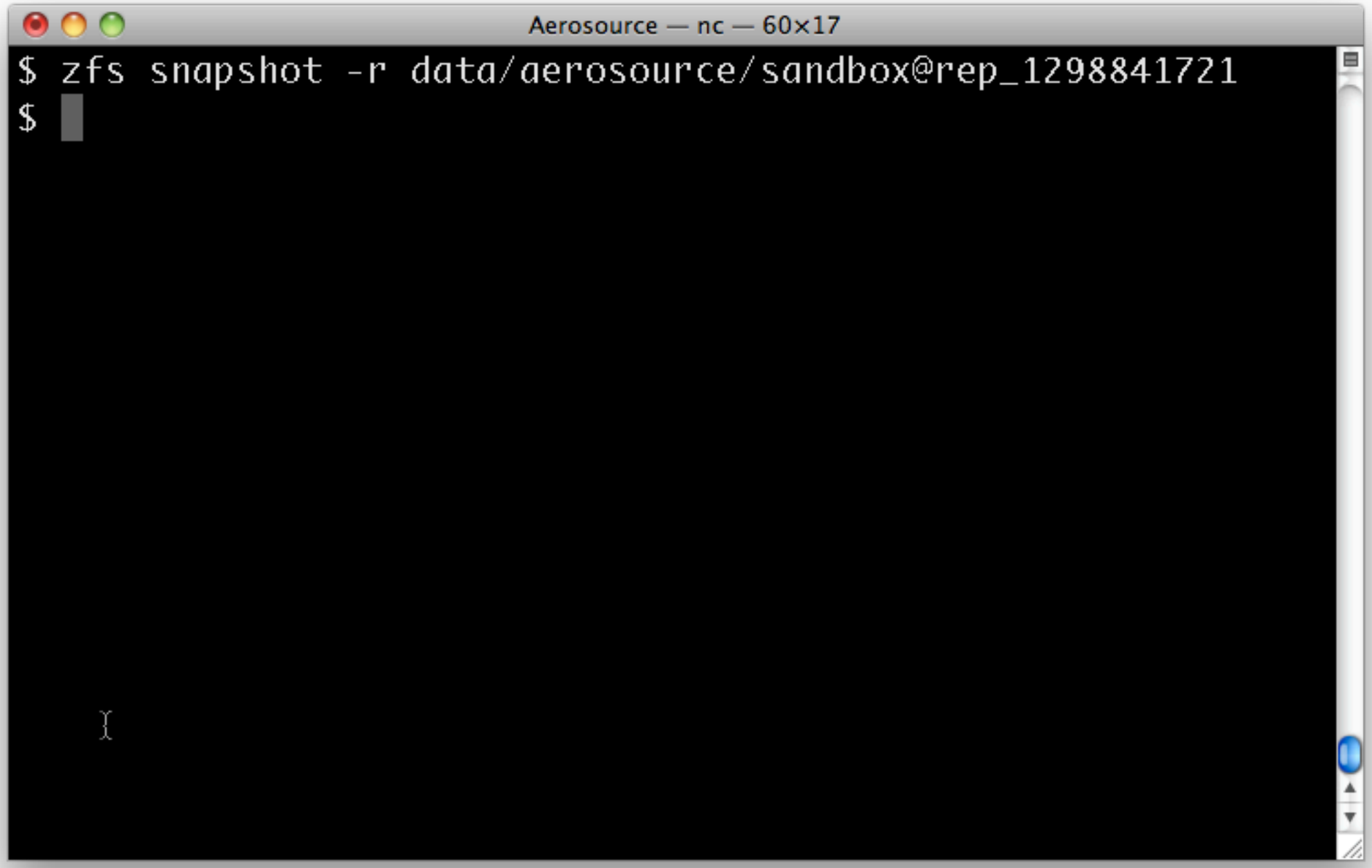
Case: Incremental



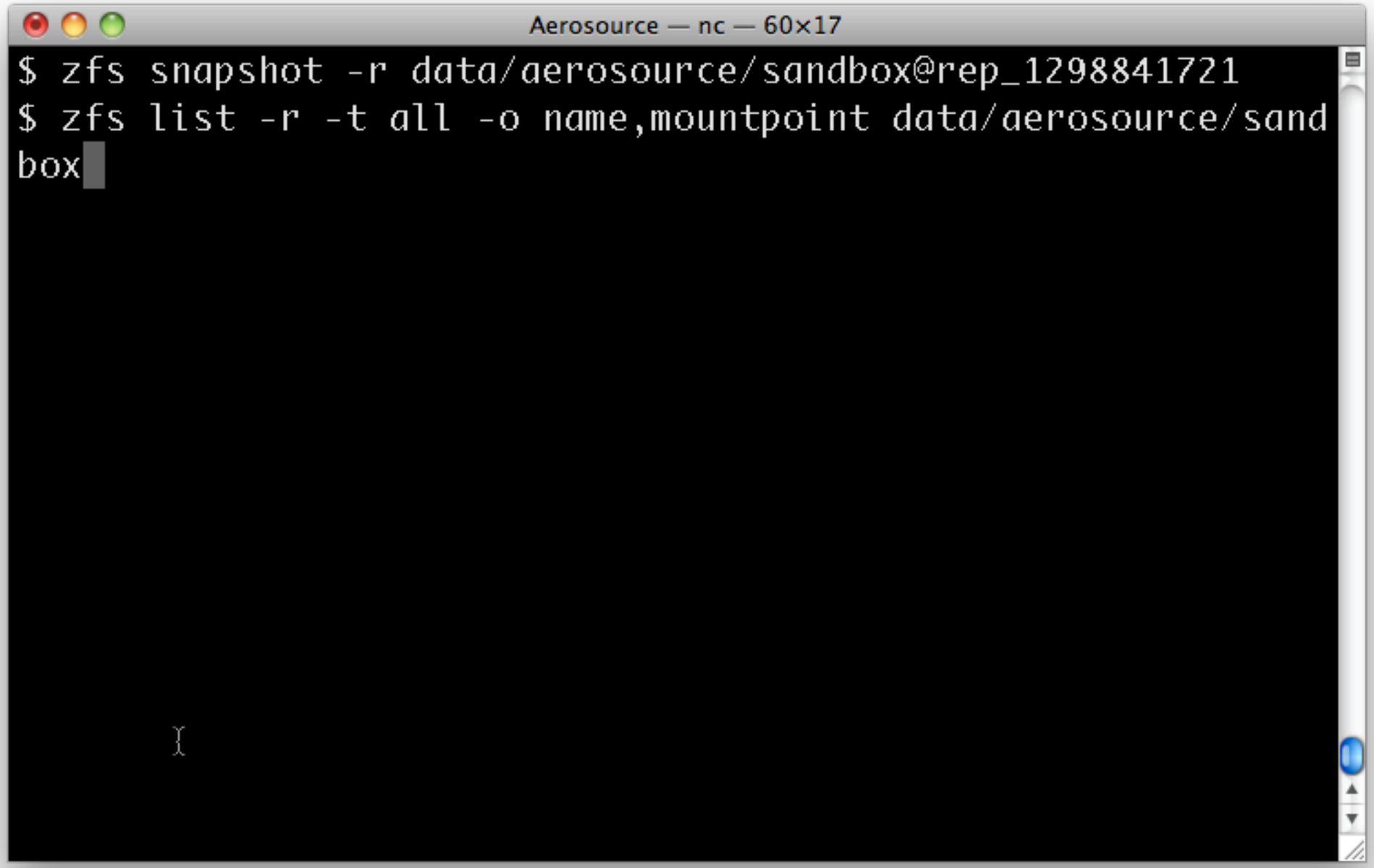
AEROSPACE


Aerosource — nc — 60×17
$ zfs snapshot -r data/aerosource/sandbox@rep_1298841721

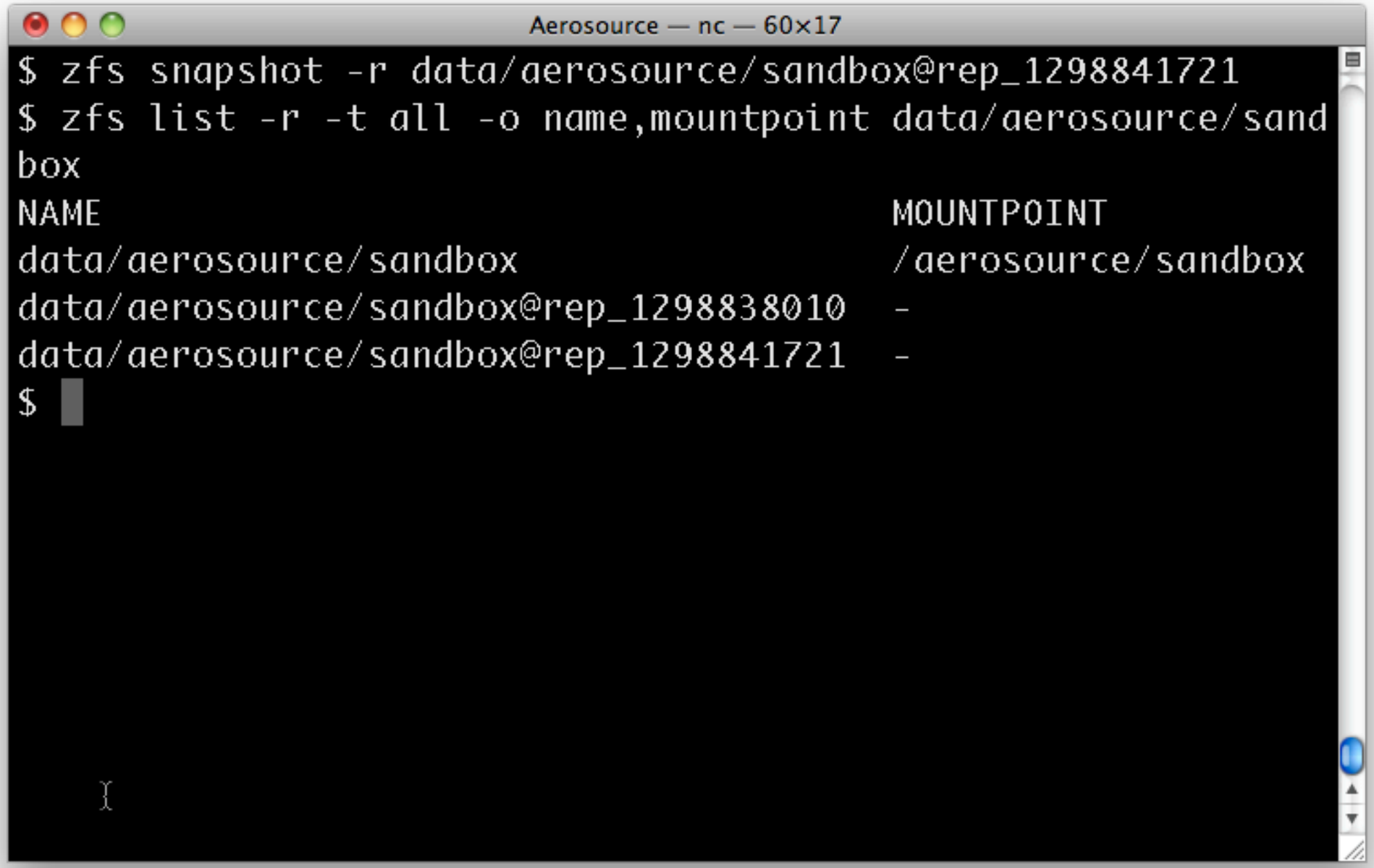
AEROSPACE


Aerosource — nc — 60×17
$ zfs snapshot -r data/aerosource/sandbox@rep_1298841721
$

AEROSPACE


Aerosource — nc — 60×17
$ zfs snapshot -r data/aerosource/sandbox@rep_1298841721
$ zfs list -r -t all -o name,mountpoint data/aerosource/sand
box

AEROSPACE

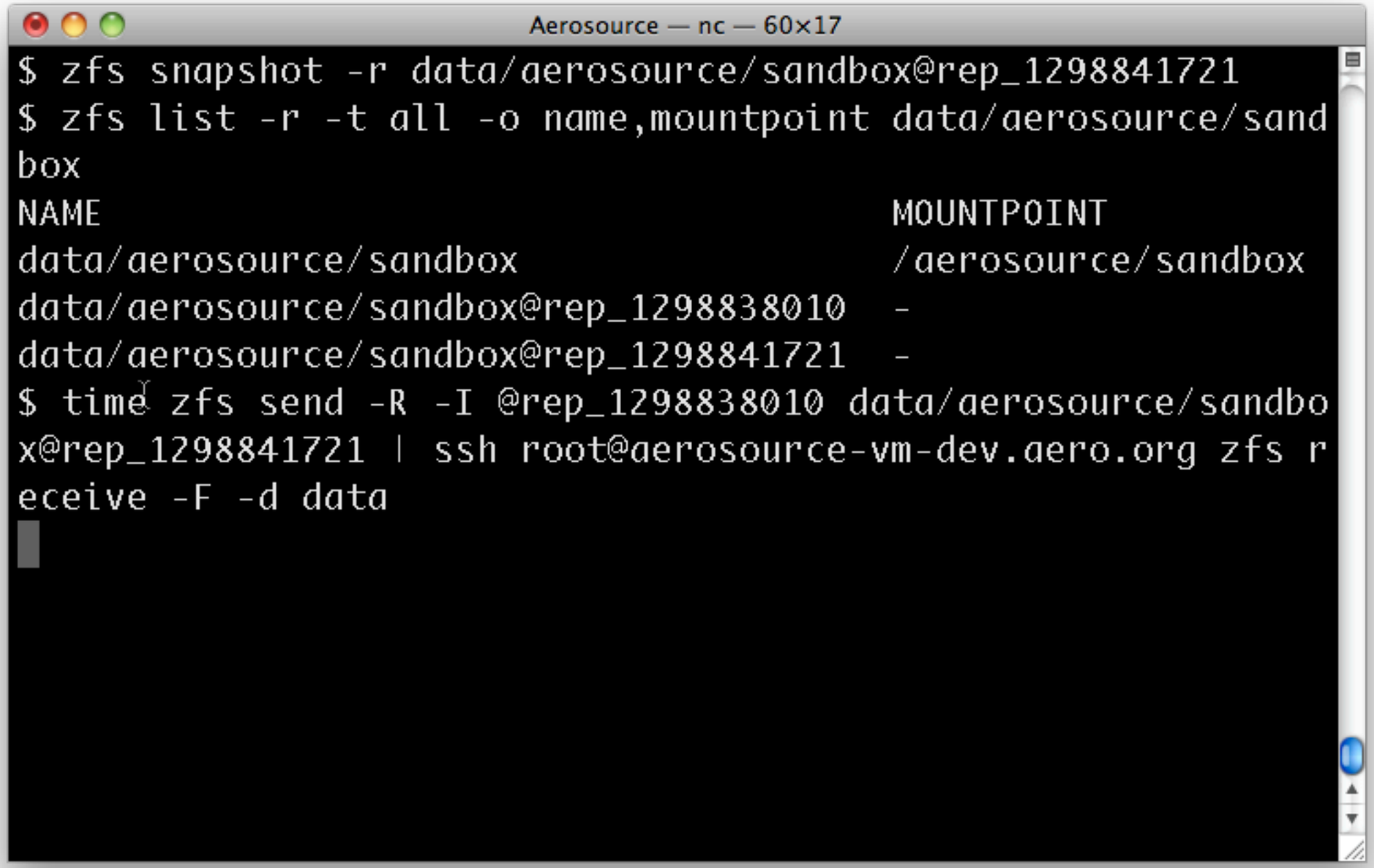

Aerosource — nc — 60×17
$ zfs snapshot -r data/aerosource/sandbox@rep_1298841721
$ zfs list -r -t all -o name,mountpoint data/aerosource/sand
box
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010 -
data/aerosource/sandbox@rep_1298841721 -
$

AEROSPACE

Aerosource — nc — 60×17

```
$ zfs snapshot -r data/aerosource/sandbox@rep_1298841721
$ zfs list -r -t all -o name,mountpoint data/aerosource/sand
box
NAME                                          MOUNTPOINT
data/aerosource/sandbox                       /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010        -
data/aerosource/sandbox@rep_1298841721        -
$ time zfs send -R -I @rep_1298838010 data/aerosource/sandbo
x@rep_1298841721 | ssh root@aerosource-vm-dev.aero.org zfs r
eceive -F -d data
```

AEROSPACE


Aerosource — nc — 60×17
$ zfs snapshot -r data/aerosource/sandbox@rep_1298841721
$ zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010 -
data/aerosource/sandbox@rep_1298841721 -
$ time zfs send -R -I @rep_1298838010 data/aerosource/sandbox@rep_1298841721 | ssh root@aerosource-vm-dev.aero.org zfs receive -F -d data

Aerosource — nc — 60×17

```
$ zfs snapshot -r data/aerosource/sandbox@rep_1298841721
$ zfs list -r -t all -o name,mountpoint data/aerosource/sand
box
NAME                                          MOUNTPOINT
data/aerosource/sandbox                       /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010        -
data/aerosource/sandbox@rep_1298841721        -
$ time zfs send -R -I @rep_1298838010 data/aerosource/sandbo
x@rep_1298841721 | ssh root@aerosource-vm-dev.aero.org zfs r
eceive -F -d data

real    0m7.468s
user    0m0.021s
sys     0m0.039s
$
```

AEROSPACE

Aerosource — nc — 60×17

```
$ zfs snapshot -r data/aerosource/sandbox@rep_1298841721
$ zfs list -r -t all -o name,mountpoint data/aerosource/sand
box
NAME                                          MOUNTPOINT
data/aerosource/sandbox                       /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010        -
data/aerosource/sandbox@rep_1298841721        -
$ time zfs send -R -I @rep_1298838010 data/aerosource/sandbo
x@rep_1298841721 | ssh root@aerosource-vm-dev.aero.org zfs r
eceive -F -d data

real    0m7.468s
user    0m0.021s
sys     0m0.039s
$ zfs set aerosource.aero.org:lastsnap:aerosource-vm-dev.aer
o.org=rep_1298841721 data/aerosource/sandbox
```

AEROSPACE

Aerosource — nc — 60×17

```
$ zfs snapshot -r data/aerosource/sandbox@rep_1298841721
$ zfs list -r -t all -o name,mountpoint data/aerosource/sand
box
NAME                                          MOUNTPOINT
data/aerosource/sandbox                       /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010        -
data/aerosource/sandbox@rep_1298841721        -
$ time zfs send -R -I @rep_1298838010 data/aerosource/sandbo
x@rep_1298841721 | ssh root@aerosource-vm-dev.aero.org zfs r
eceive -F -d data

real    0m7.468s
user    0m0.021s
sys     0m0.039s
$ zfs set aerosource.aero.org:lastsnap:aerosource-vm-dev.aer
o.org=rep_1298841721 data/aerosource/sandbox
$
```

AEROSPACE

Aerosource — nc — 60×17

```
$ zfs snapshot -r data/aerosource/sandbox@rep_1298841721
$ zfs list -r -t all -o name,mountpoint data/aerosource/sand
box
NAME                                          MOUNTPOINT
data/aerosource/sandbox                       /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010        -
data/aerosource/sandbox@rep_1298841721        -
$ time zfs send -R -I @rep_1298838010 data/aerosource/sandbo
x@rep_1298841721 | ssh root@aerosource-vm-dev.aero.org zfs r
eceive -F -d data

real    0m7.468s
user    0m0.021s
sys     0m0.039s
$ zfs set aerosource.aero.org:lastsnap:aerosource-vm-dev.aer
o.org=rep_1298841721 data/aerosource/sandbox
$ zfs destroy -r data/aerosource/sandbox@rep_1298838010
```

AEROSPACE

Aerosource — nc — 60×17

```
$ zfs list -r -t all -o name,mountpoint data/aerosource/sand
box
NAME                                          MOUNTPOINT
data/aerosource/sandbox                       /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010        -
data/aerosource/sandbox@rep_1298841721        -
$ time zfs send -R -I @rep_1298838010 data/aerosource/sandbo
x@rep_1298841721 | ssh root@aerosource-vm-dev.aero.org zfs r
eceive -F -d data

real    0m7.468s
user    0m0.021s
sys     0m0.039s
$ zfs set aerosource.aero.org:lastsnap:aerosource-vm-dev.aer
o.org=rep_1298841721 data/aerosource/sandbox
$ zfs destroy -r data/aerosource/sandbox@rep_1298838010
$
```

AEROSPACE

Monday, September 26, 2011

And that's it for the update.

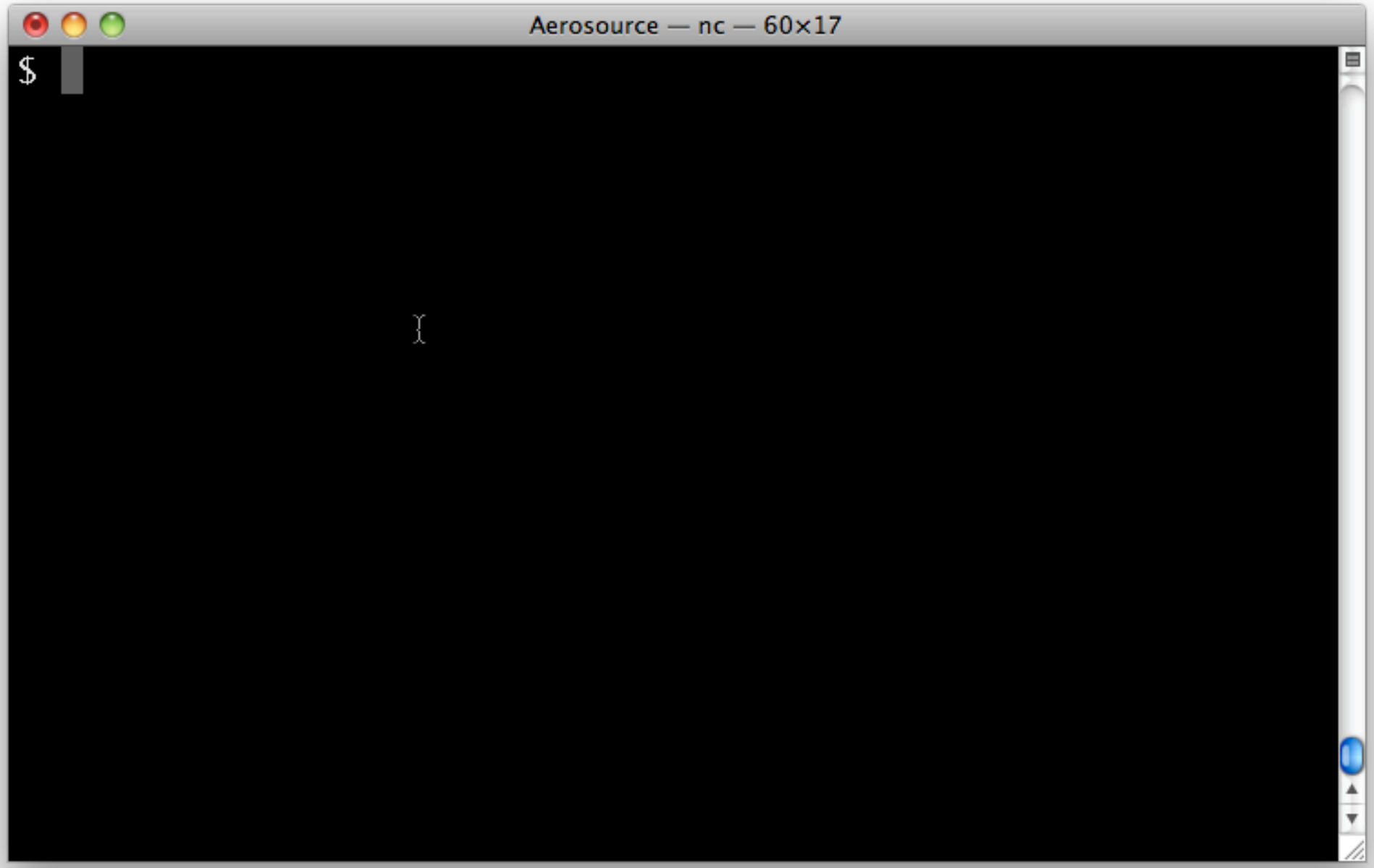
You might have noticed that we didn't remove the snapshot on the remote system. The reason for this is that we don't need to due to our use of the -F option to zfs receive. It removes filesystems and snapshots that were removed on the source side as well as any changes made on the target side since the snapshot specified by -I

Aerosource — nc — 60×17

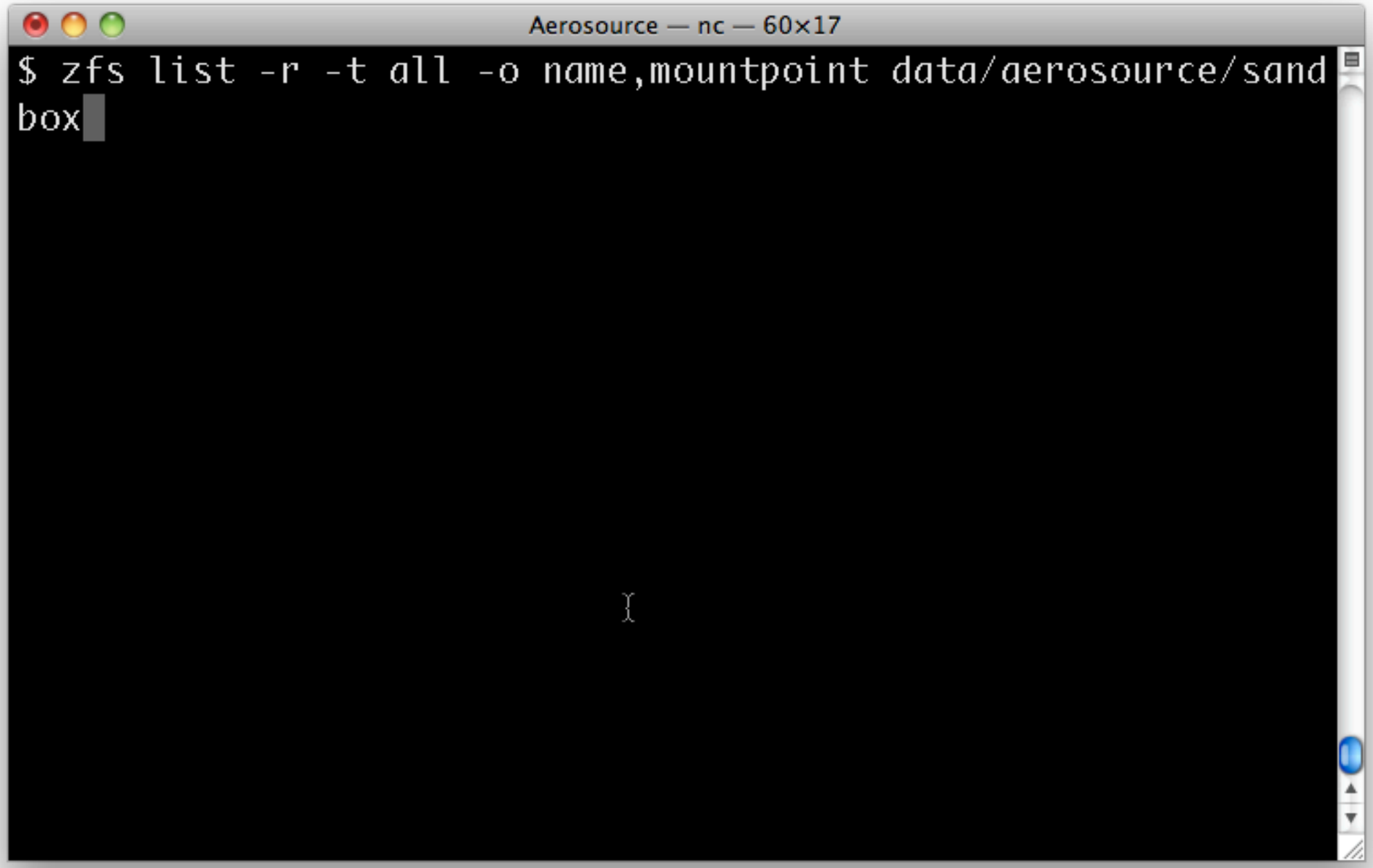
```
$ zfs list -r -t all -o name,mountpoint data/aerosource/sand
box
NAME                                          MOUNTPOINT
data/aerosource/sandbox                       /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010  -
data/aerosource/sandbox@rep_1298841721  -
$ time zfs send -R -I @rep_1298838010 data/aerosource/sandbo
x@rep_1298841721 | ssh root@aerosource-vm-dev.aero.org zfs r
eceive -F -d data

real    0m7.468s
user    0m0.021s
sys     0m0.039s
$ zfs set aerosource.aero.org:lastsnap:aerosource-vm-dev.aer
o.org=rep_1298841721 data/aerosource/sandbox
$ zfs destroy -r data/aerosource/sandbox@rep_1298838010
$ clear
```

AEROSPACE

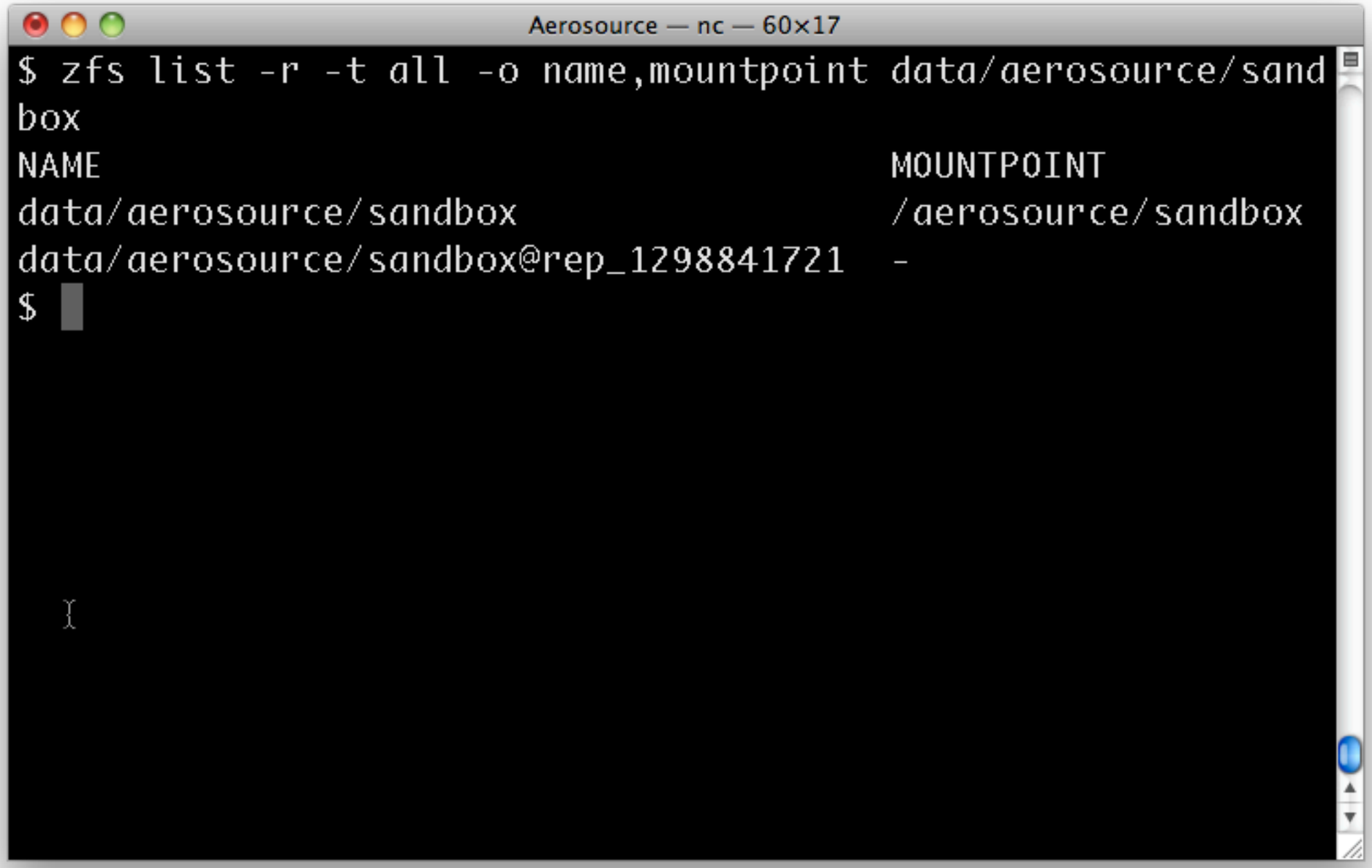
This can be seen in a quick example.



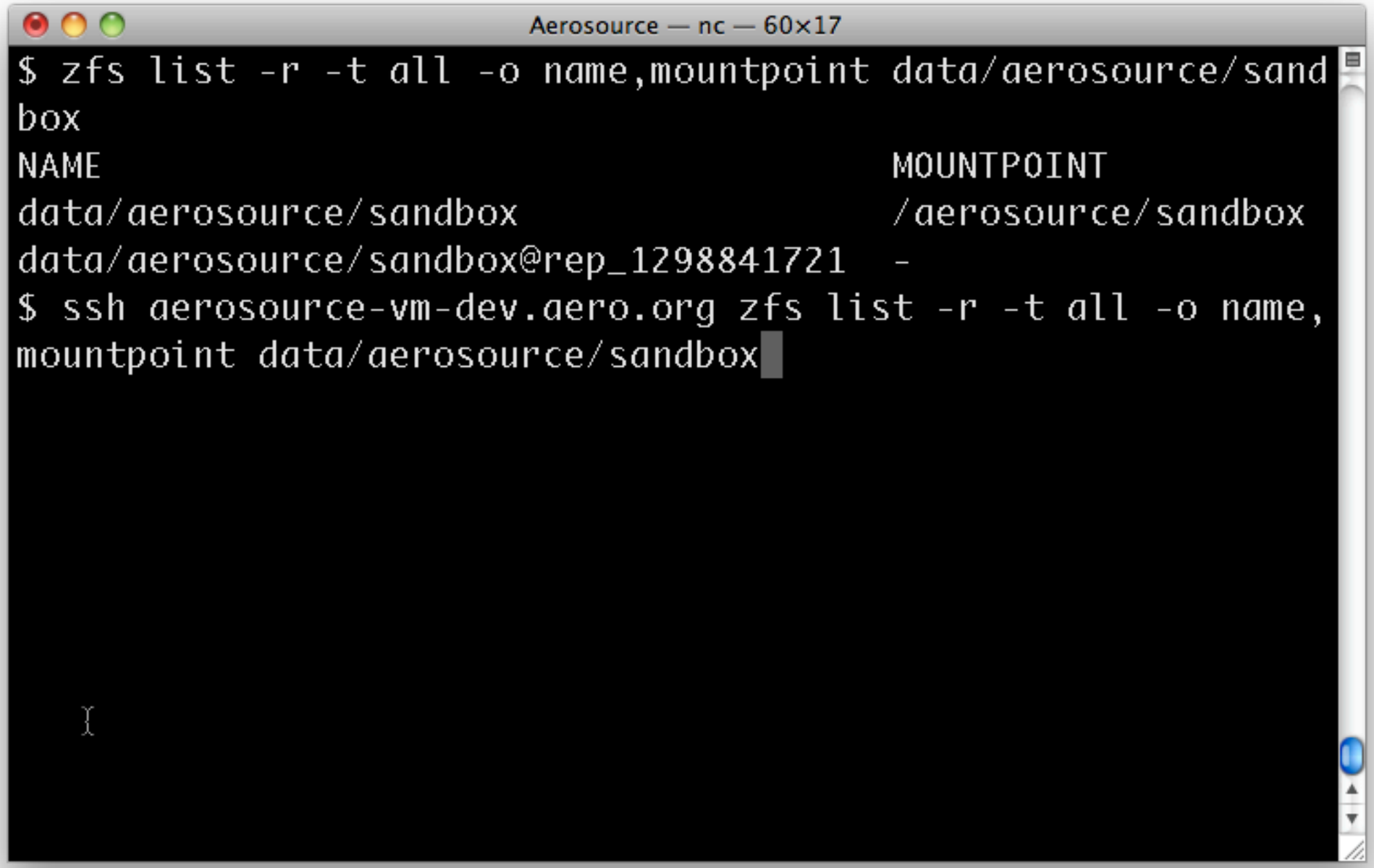
AEROSPACE


Aerosource — nc — 60×17
$ zfs list -r -t all -o name,mountpoint data/aerosource/sandbox

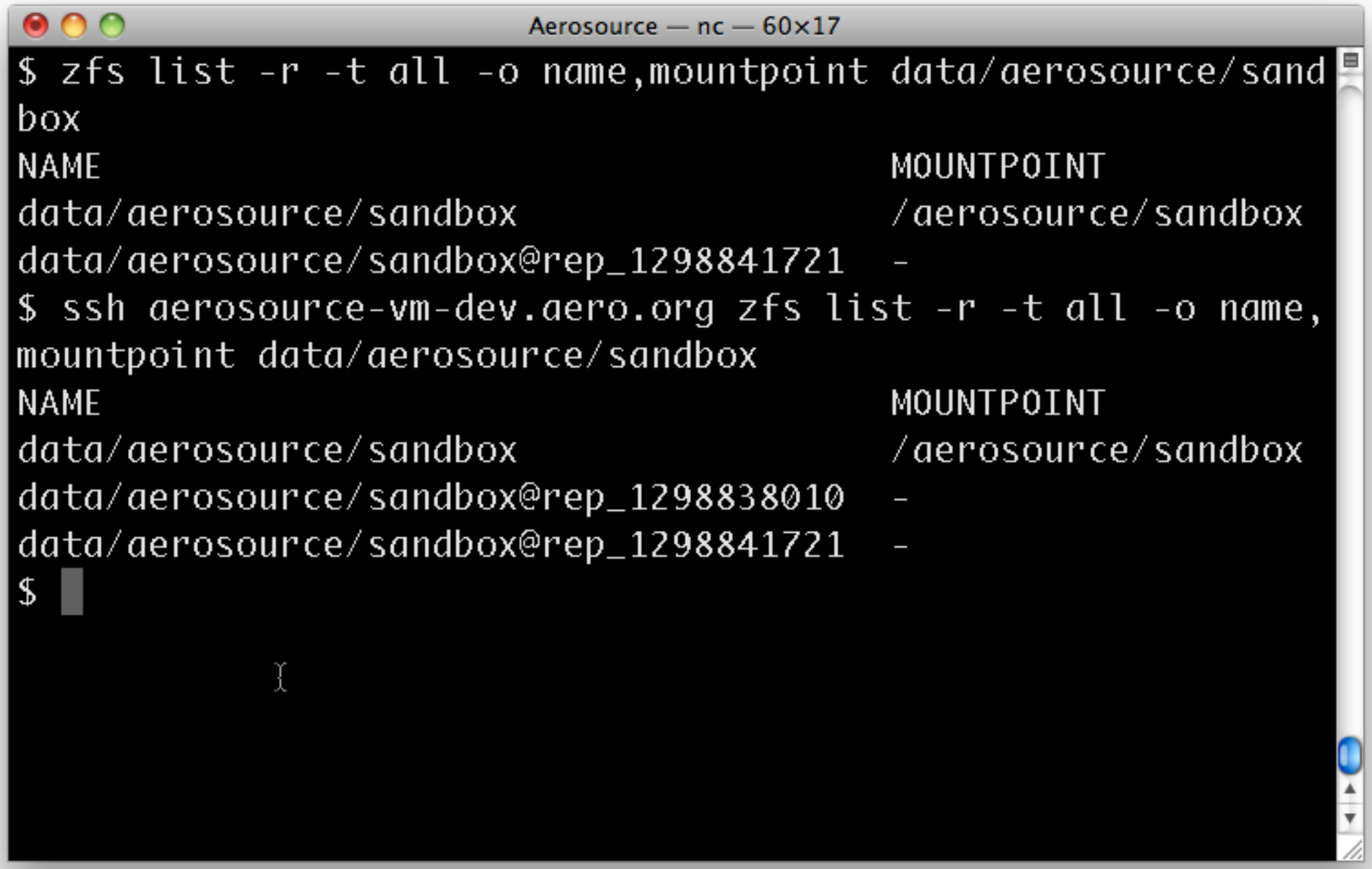
AEROSPACE


Aerosource — nc — 60×17
$ zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298841721 -
$

AEROSPACE


Aerosource — nc — 60×17
$ zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME   MOUNTPOINT
data/aerosource/sandbox   /aerosource/sandbox
data/aerosource/sandbox@rep_1298841721   -
$ ssh aerosource-vm-dev.aero.org zfs list -r -t all -o name,mountpoint data/aerosource/sandbox

AEROSPACE

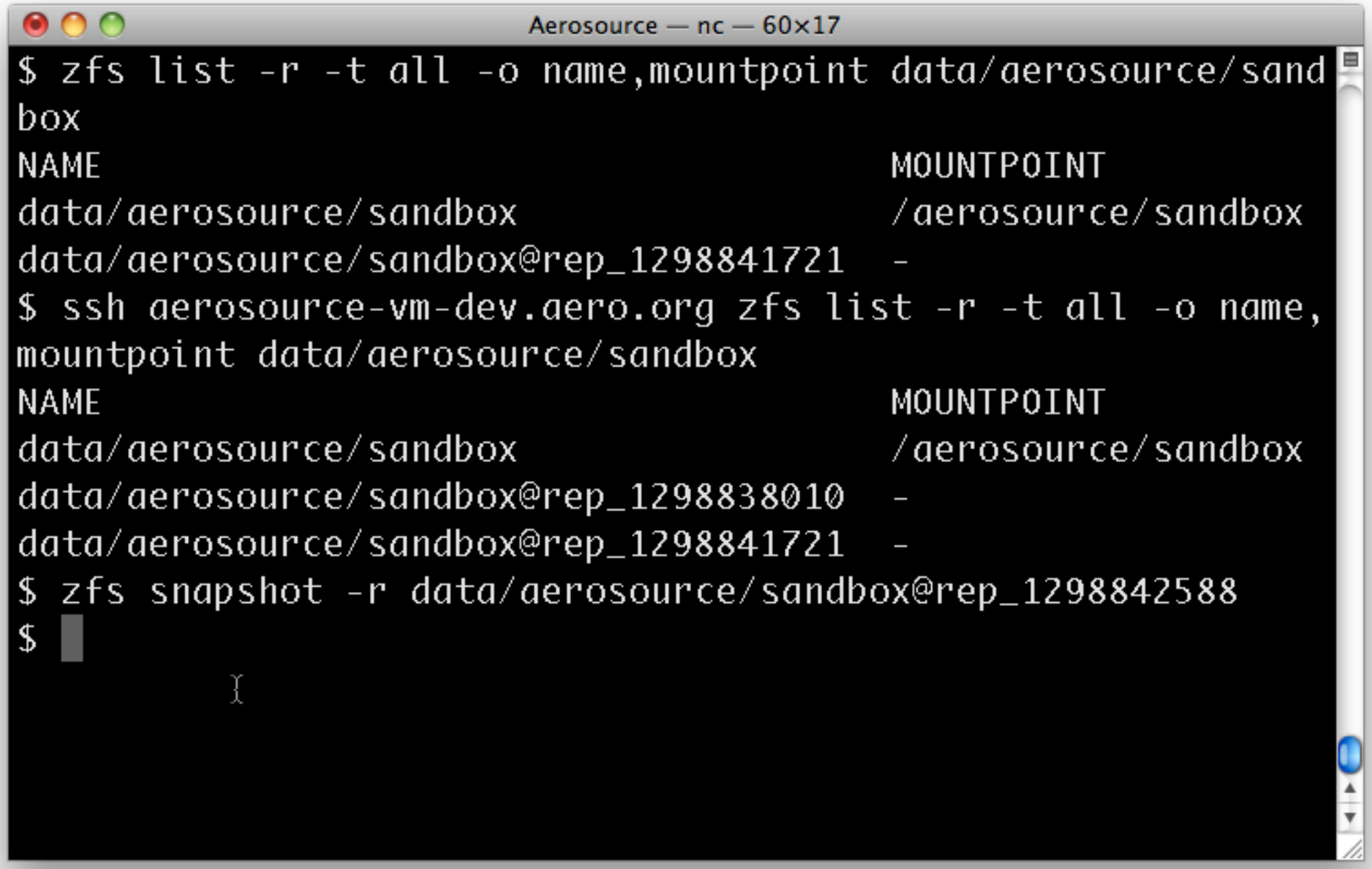

Aerosource — nc — 60×17
$ zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298841721 -
$ ssh aerosource-vm-dev.aero.org zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010 -
data/aerosource/sandbox@rep_1298841721 -
$

AEROSPACE

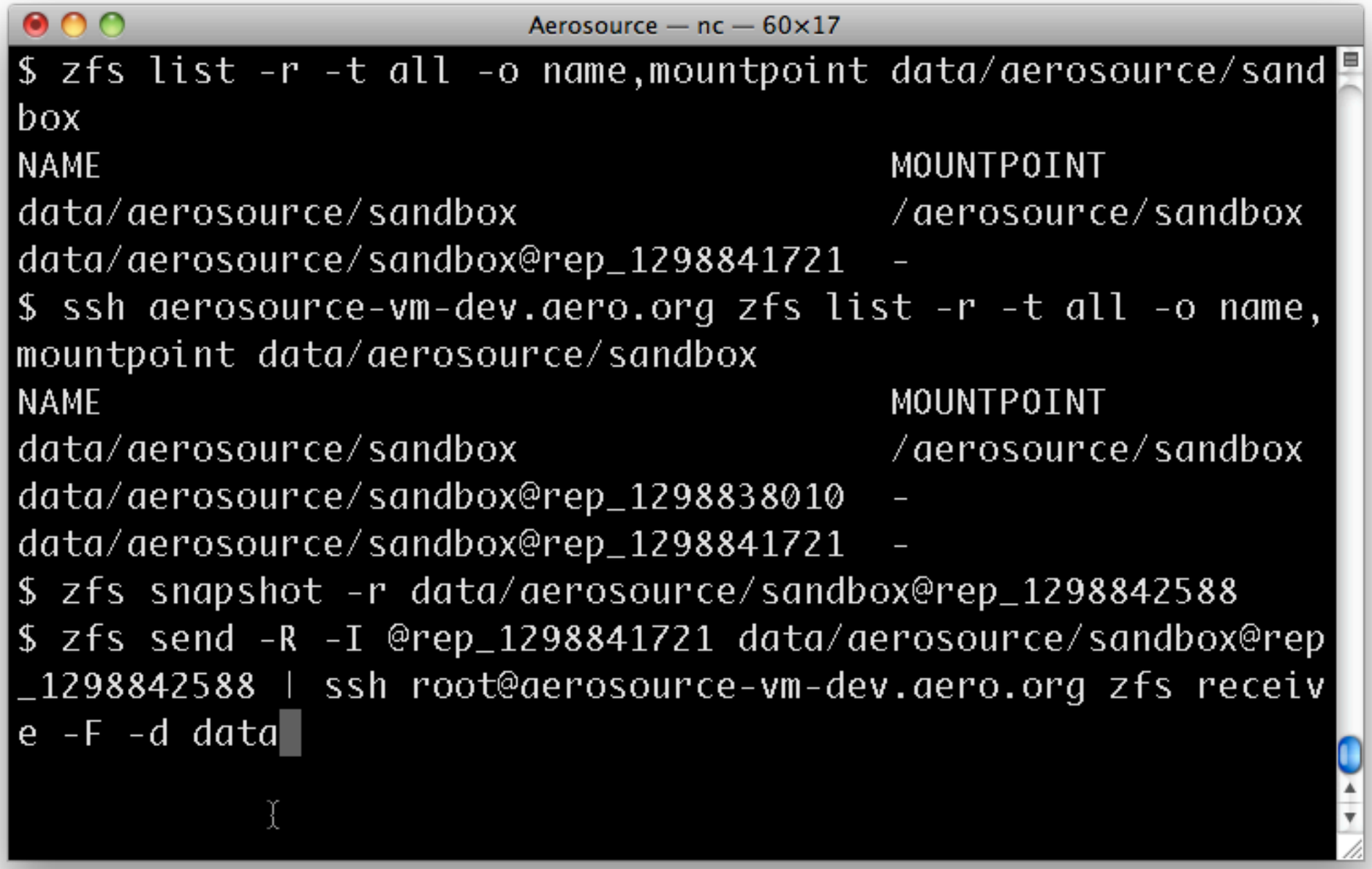
Aerosource — nc — 60×17

```
$ zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME                                      MOUNTPOINT
data/aerosource/sandbox                   /aerosource/sandbox
data/aerosource/sandbox@rep_1298841721    -
$ ssh aerosource-vm-dev.aero.org zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME                                      MOUNTPOINT
data/aerosource/sandbox                   /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010    -
data/aerosource/sandbox@rep_1298841721    -
$ zfs snapshot -r data/aerosource/sandbox@rep_1298842588
```

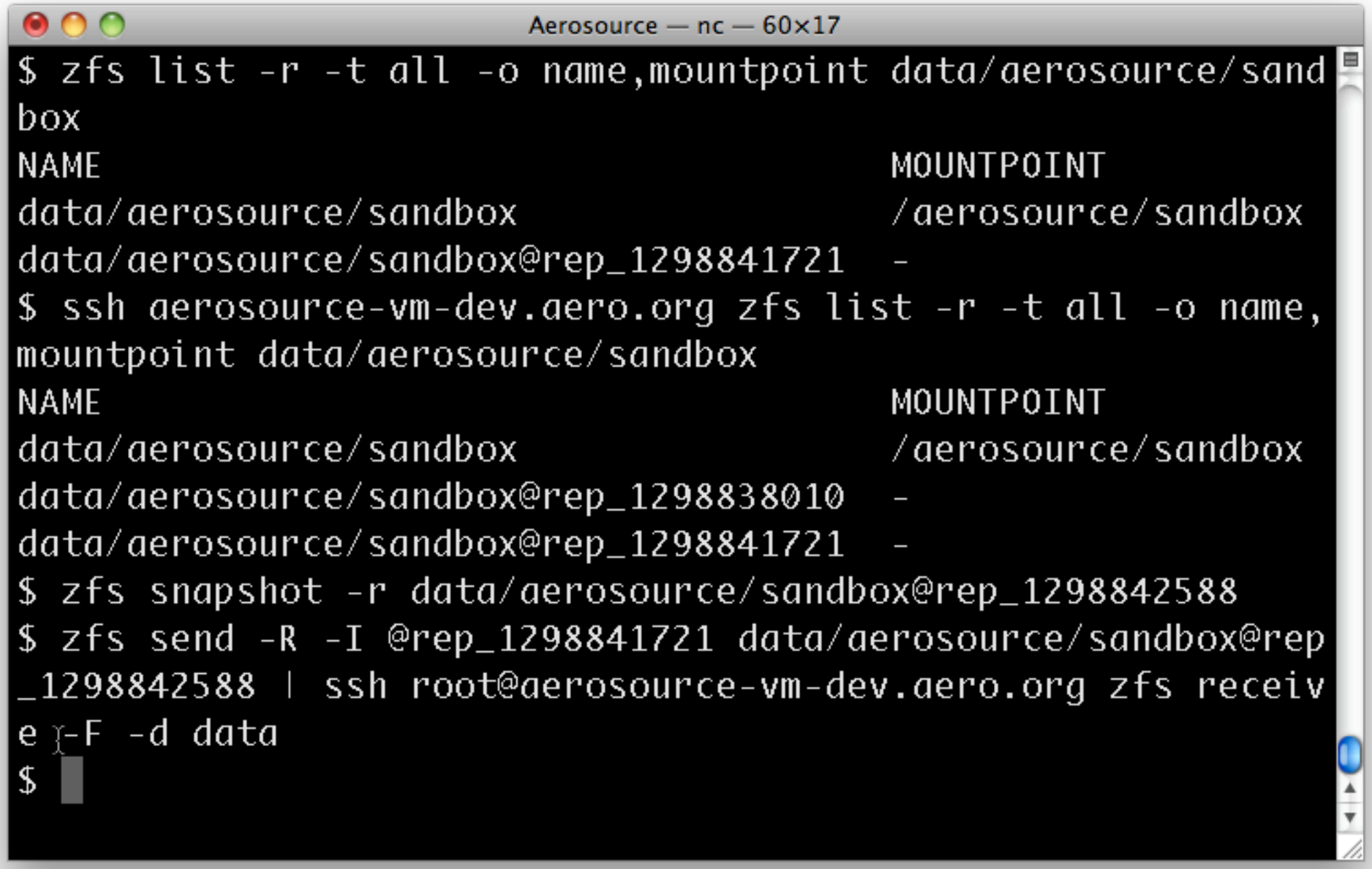
AEROSPACE


Aerosource — nc — 60×17
$ zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298841721 -
$ ssh aerosource-vm-dev.aero.org zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010 -
data/aerosource/sandbox@rep_1298841721 -
$ zfs snapshot -r data/aerosource/sandbox@rep_1298842588
$

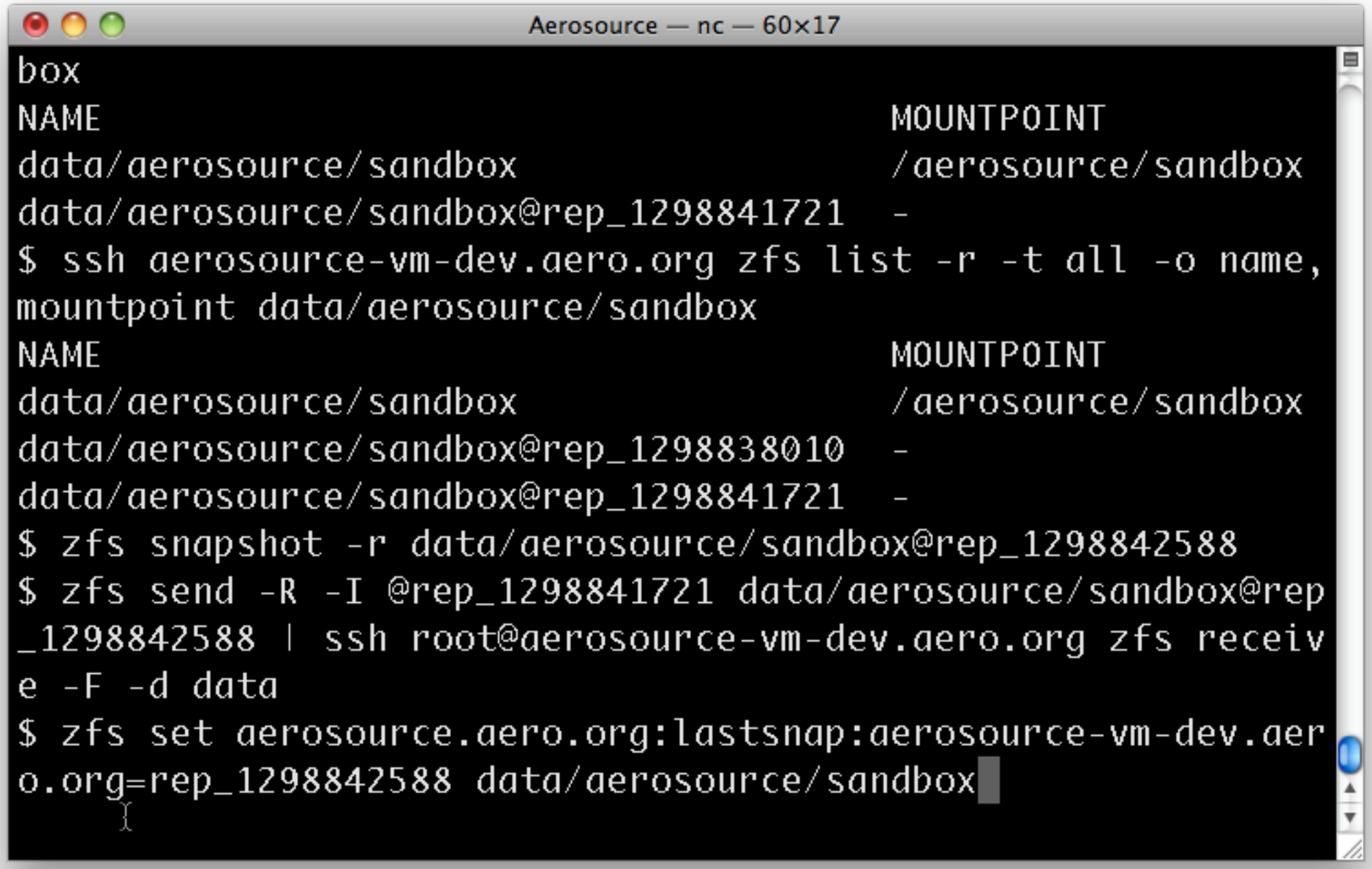
AEROSPACE


Aerosource — nc — 60×17
$ zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298841721 -
$ ssh aerosource-vm-dev.aero.org zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010 -
data/aerosource/sandbox@rep_1298841721 -
$ zfs snapshot -r data/aerosource/sandbox@rep_1298842588
$ zfs send -R -I @rep_1298841721 data/aerosource/sandbox@rep_1298842588 | ssh root@aerosource-vm-dev.aero.org zfs receive -F -d data

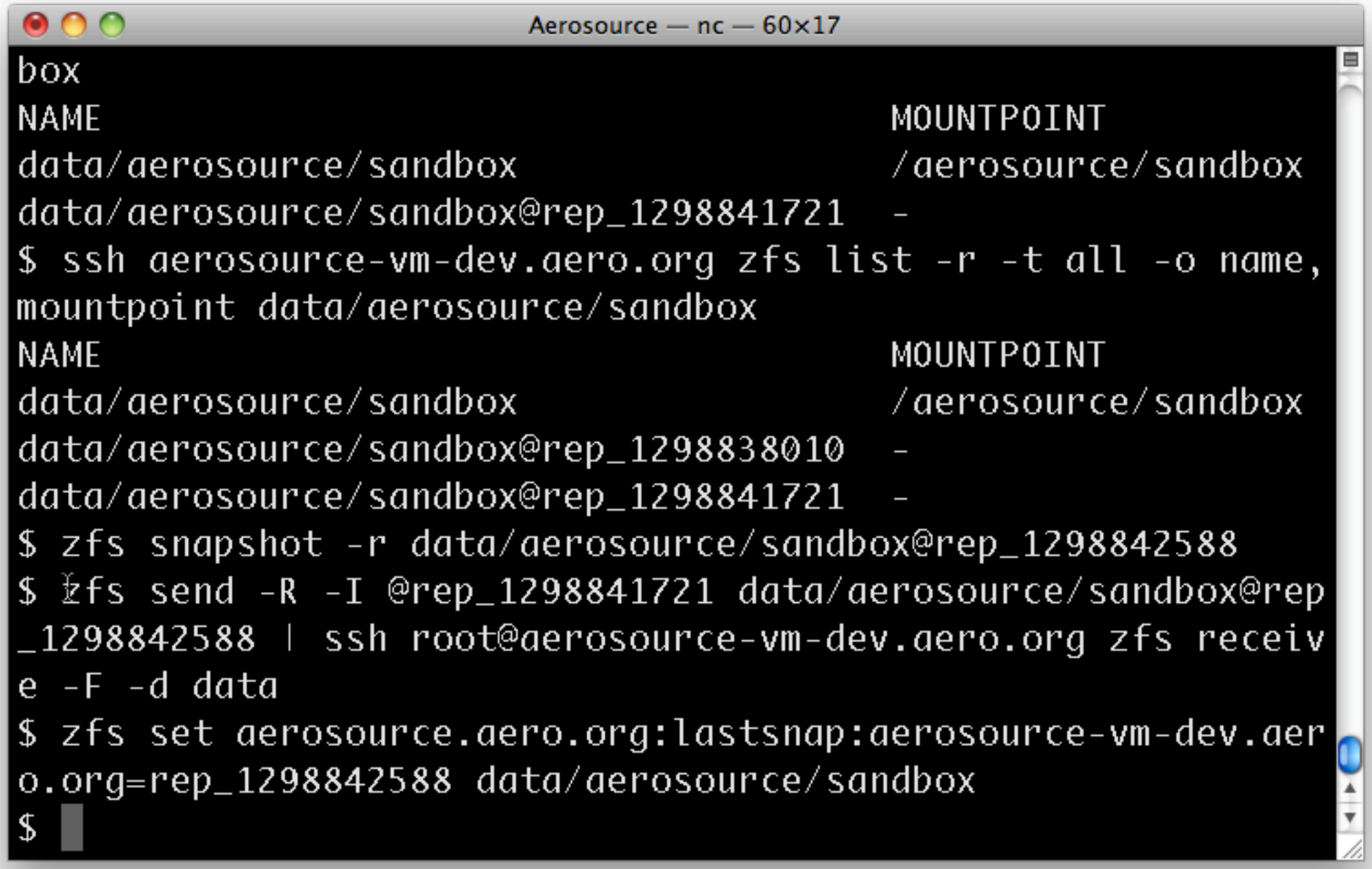
AEROSPACE


Aerosource — nc — 60×17
$ zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298841721 -
$ ssh aerosource-vm-dev.aero.org zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010 -
data/aerosource/sandbox@rep_1298841721 -
$ zfs snapshot -r data/aerosource/sandbox@rep_1298842588
$ zfs send -R -I @rep_1298841721 data/aerosource/sandbox@rep_1298842588 | ssh root@aerosource-vm-dev.aero.org zfs receive -F -d data
$

AEROSPACE


Aerosource — nc — 60×17
box
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298841721 -
$ ssh aerosource-vm-dev.aero.org zfs list -r -t all -o name,
mountpoint data/aerosource/sandbox
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010 -
data/aerosource/sandbox@rep_1298841721 -
$ zfs snapshot -r data/aerosource/sandbox@rep_1298842588
$ zfs send -R -I @rep_1298841721 data/aerosource/sandbox@rep
_1298842588 | ssh root@aerosource-vm-dev.aero.org zfs receiv
e -F -d data
$ zfs set aerosource.aero.org:lastsnap:aerosource-vm-dev.aer
o.org=rep_1298842588 data/aerosource/sandbox

AEROSPACE


Aerosource — nc — 60×17
box
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298841721 -
$ ssh aerosource-vm-dev.aero.org zfs list -r -t all -o name,
mountpoint data/aerosource/sandbox
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010 -
data/aerosource/sandbox@rep_1298841721 -
$ zfs snapshot -r data/aerosource/sandbox@rep_1298842588
$ zfs send -R -I @rep_1298841721 data/aerosource/sandbox@rep
_1298842588 | ssh root@aerosource-vm-dev.aero.org zfs receiv
e -F -d data
$ zfs set aerosource.aero.org:lastsnap:aerosource-vm-dev.aer
o.org=rep_1298842588 data/aerosource/sandbox
$

AEROSPACE

Aerosource — nc — 60×17

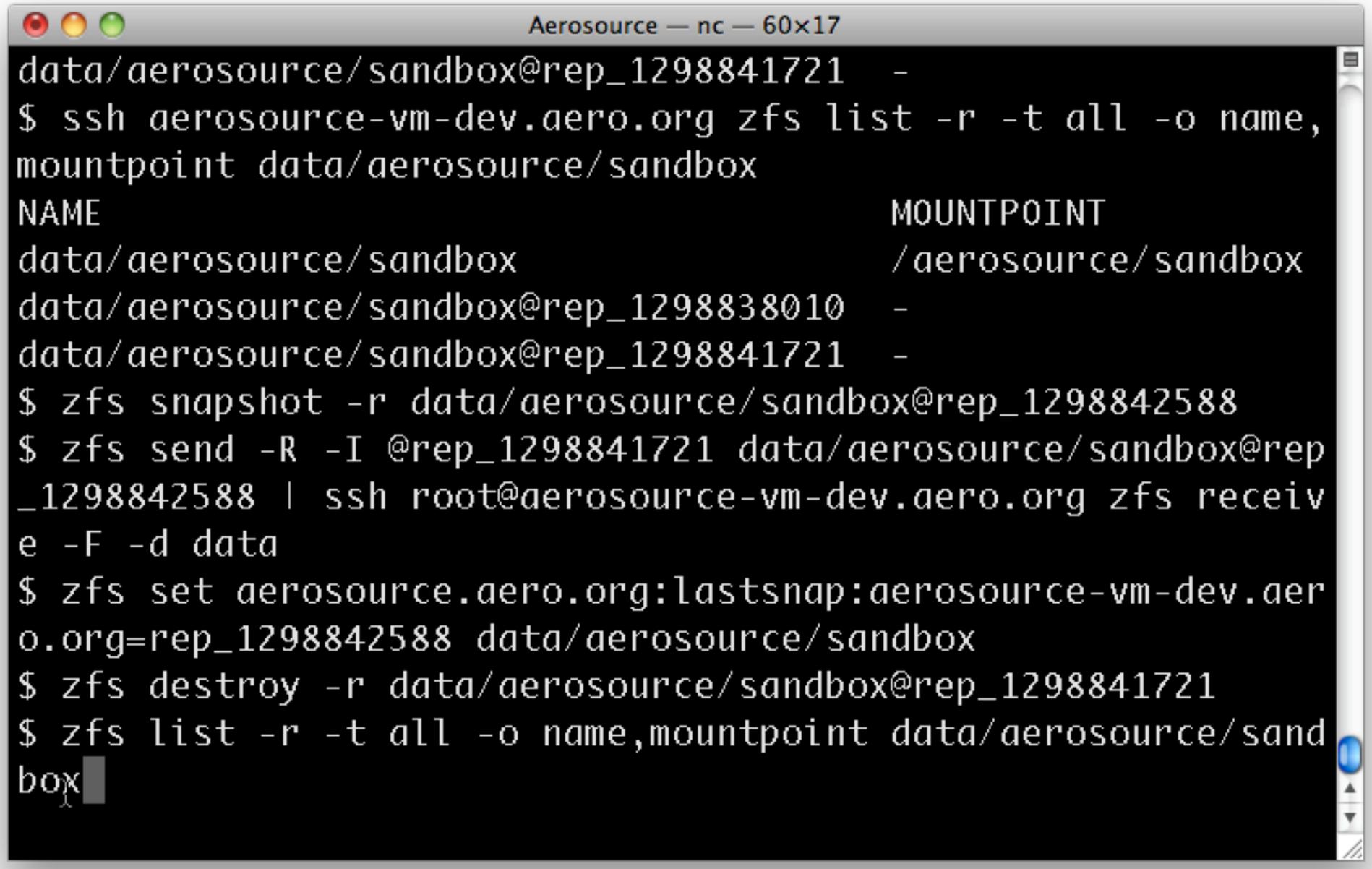
```
data/aerosource/sandbox                         /aerosource/sandbox
data/aerosource/sandbox@rep_1298841721  -
$ ssh aerosource-vm-dev.aero.org zfs list -r -t all -o name,
mountpoint data/aerosource/sandbox
NAME                                            MOUNTPOINT
data/aerosource/sandbox                         /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010  -
data/aerosource/sandbox@rep_1298841721  -
$ zfs snapshot -r data/aerosource/sandbox@rep_1298842588
$ zfs send -R -I @rep_1298841721 data/aerosource/sandbox@rep
_1298842588 | ssh root@aerosource-vm-dev.aero.org zfs receiv
e -F -d data
$ zfs set aerosource.aero.org:lastsnap:aerosource-vm-dev.aer
o.org=rep_1298842588 data/aerosource/sandbox
$ zfs destroy -r data/aerosource/sandbox@rep_1298841721
```

AEROSPACE

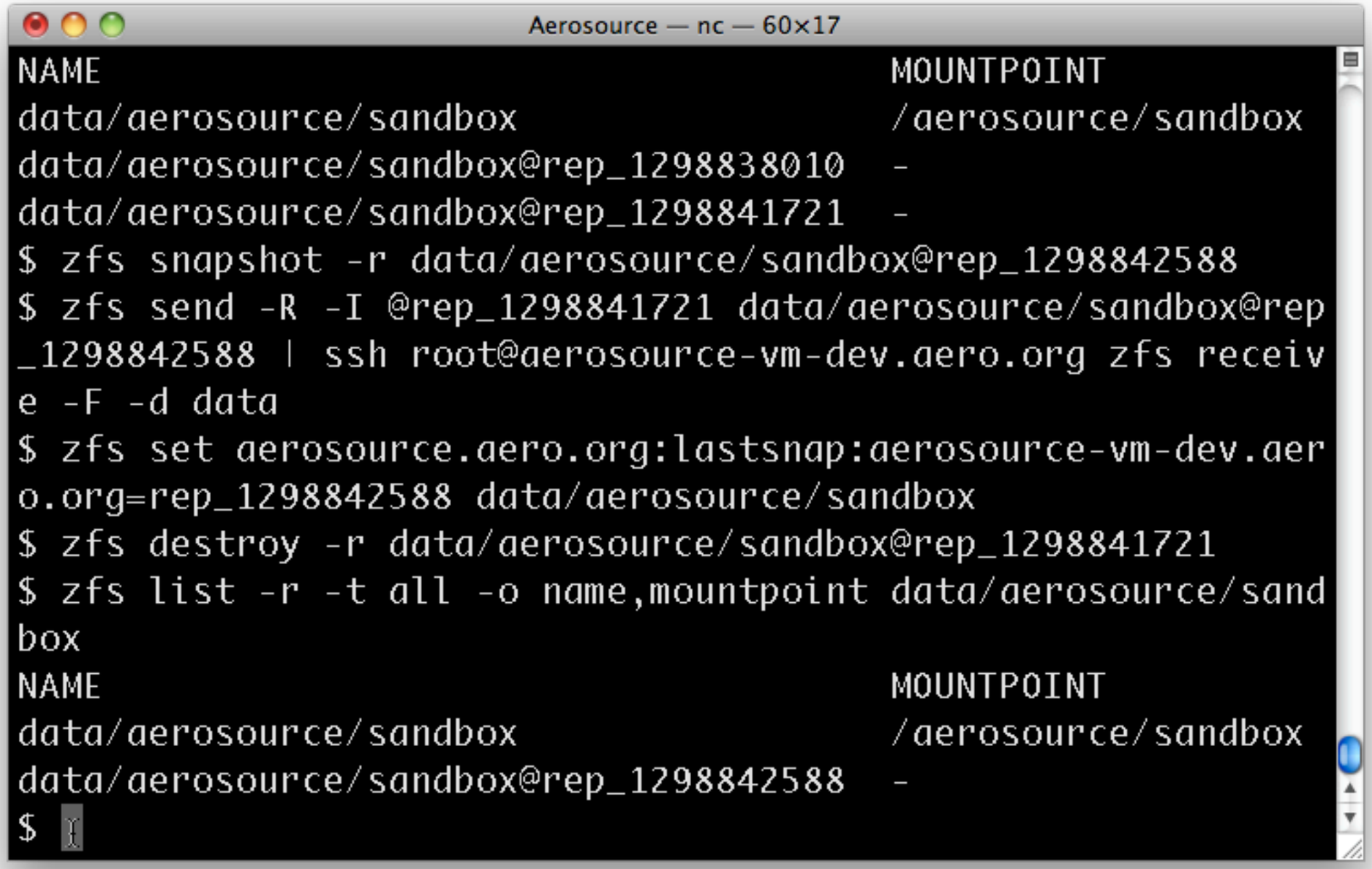
Aerosource — nc — 60×17

```
data/aerosource/sandbox                         /aerosource/sandbox
data/aerosource/sandbox@rep_1298841721  -
$ ssh aerosource-vm-dev.aero.org zfs list -r -t all -o name,
mountpoint data/aerosource/sandbox
NAME                                            MOUNTPOINT
data/aerosource/sandbox                         /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010  -
data/aerosource/sandbox@rep_1298841721  -
$ zfs snapshot -r data/aerosource/sandbox@rep_1298842588
$ zfs send -R -I @rep_1298841721 data/aerosource/sandbox@rep
_1298842588 | ssh root@aerosource-vm-dev.aero.org zfs receiv
e -F -d data
$ zfs set aerosource.aero.org:lastsnap:aerosource-vm-dev.aer
o.org=rep_1298842588 data/aerosource/sandbox
$ zfs destroy -r data/aerosource/sandbox@rep_1298841721
$
```

AEROSPACE


Aerosource — nc — 60×17
data/aerosource/sandbox@rep_1298841721 -
$ ssh aerosource-vm-dev.aero.org zfs list -r -t all -o name,
mountpoint data/aerosource/sandbox
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010 -
data/aerosource/sandbox@rep_1298841721 -
$ zfs snapshot -r data/aerosource/sandbox@rep_1298842588
$ zfs send -R -I @rep_1298841721 data/aerosource/sandbox@rep
_1298842588 | ssh root@aerosource-vm-dev.aero.org zfs receiv
e -F -d data
$ zfs set aerosource.aero.org:lastsnap:aerosource-vm-dev.aer
o.org=rep_1298842588 data/aerosource/sandbox
$ zfs destroy -r data/aerosource/sandbox@rep_1298841721
$ zfs list -r -t all -o name,mountpoint data/aerosource/sand
box

AEROSPACE

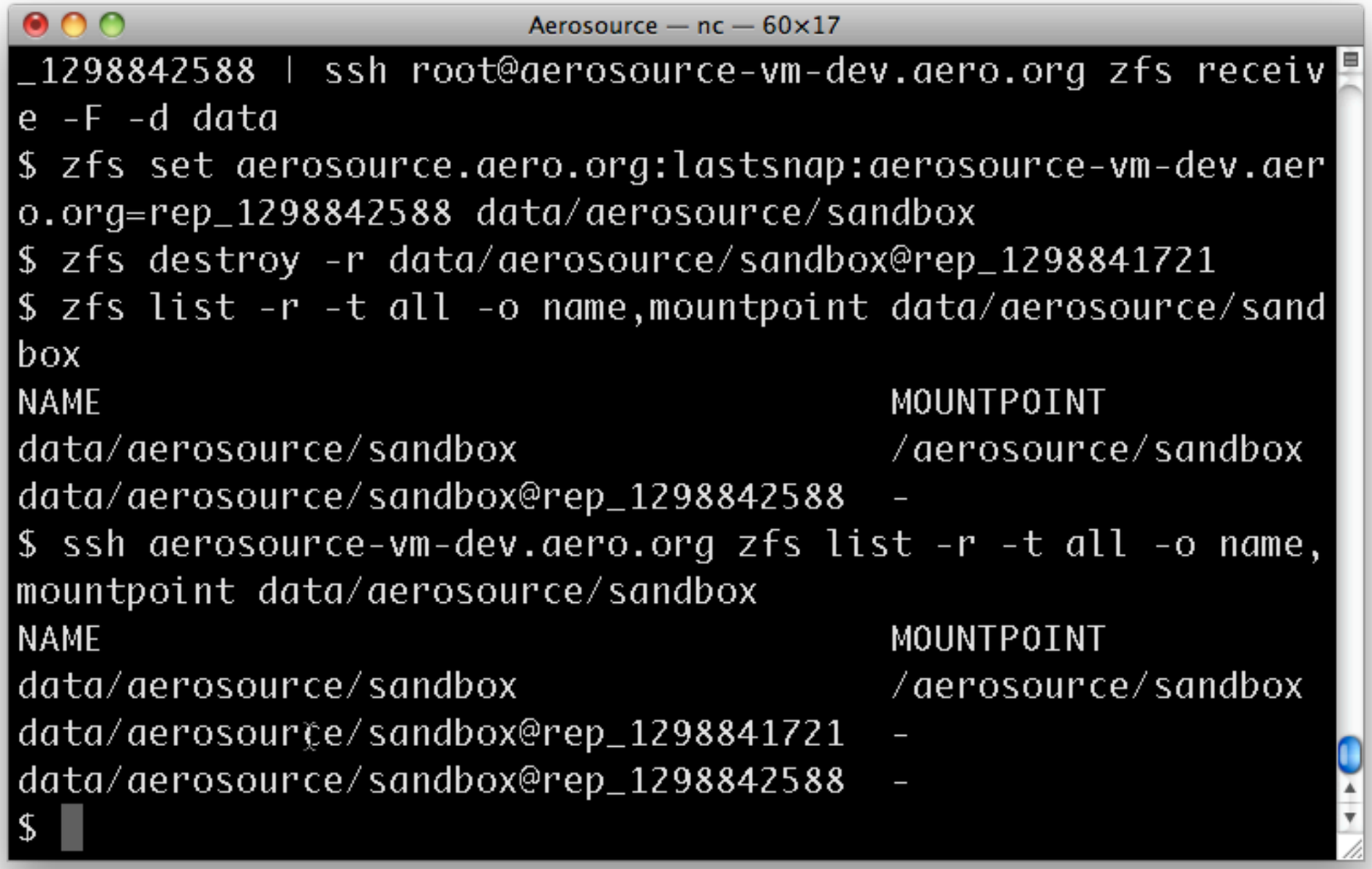

Aerosource — nc — 60×17
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010 -
data/aerosource/sandbox@rep_1298841721 -
$ zfs snapshot -r data/aerosource/sandbox@rep_1298842588
$ zfs send -R -I @rep_1298841721 data/aerosource/sandbox@rep_1298842588 | ssh root@aerosource-vm-dev.aero.org zfs receive -F -d data
$ zfs set aerosource.aero.org:lastsnap:aerosource-vm-dev.aero.org=rep_1298842588 data/aerosource/sandbox
$ zfs destroy -r data/aerosource/sandbox@rep_1298841721
$ zfs list -r -t all -o name,mountpoint data/aerosource/sandbox
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298842588 -
$

AEROSPACE

Aerosource — nc — 60×17

```
data/aerosource/sandbox                         /aerosource/sandbox
data/aerosource/sandbox@rep_1298838010  -
data/aerosource/sandbox@rep_1298841721  -
$ zfs snapshot -r data/aerosource/sandbox@rep_1298842588
$ zfs send -R -I @rep_1298841721 data/aerosource/sandbox@rep
_1298842588 | ssh root@aerosource-vm-dev.aero.org zfs receiv
e -F -d data
$ zfs set aerosource.aero.org:lastsnap:aerosource-vm-dev.aer
o.org=rep_1298842588 data/aerosource/sandbox
$ zfs destroy -r data/aerosource/sandbox@rep_1298841721
$ zfs list -r -t all -o name,mountpoint data/aerosource/sand
box
NAME                                            MOUNTPOINT
data/aerosource/sandbox                         /aerosource/sandbox
data/aerosource/sandbox@rep_1298842588  -
$ ssh aerosource-vm-dev.aero.org zfs list -r -t all -o name,
mountpoint data/aerosource/sandbox
```

AEROSPACE


Aerosource — nc — 60×17
_1298842588 | ssh root@aerosource-vm-dev.aero.org zfs receiv
e -F -d data
$ zfs set aerosource.aero.org:lastsnap:aerosource-vm-dev.aer
o.org=rep_1298842588 data/aerosource/sandbox
$ zfs destroy -r data/aerosource/sandbox@rep_1298841721
$ zfs list -r -t all -o name,mountpoint data/aerosource/sand
box
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298842588 -
$ ssh aerosource-vm-dev.aero.org zfs list -r -t all -o name,
mountpoint data/aerosource/sandbox
NAME MOUNTPOINT
data/aerosource/sandbox /aerosource/sandbox
data/aerosource/sandbox@rep_1298841721 -
data/aerosource/sandbox@rep_1298842588 -
$

AEROSPACE

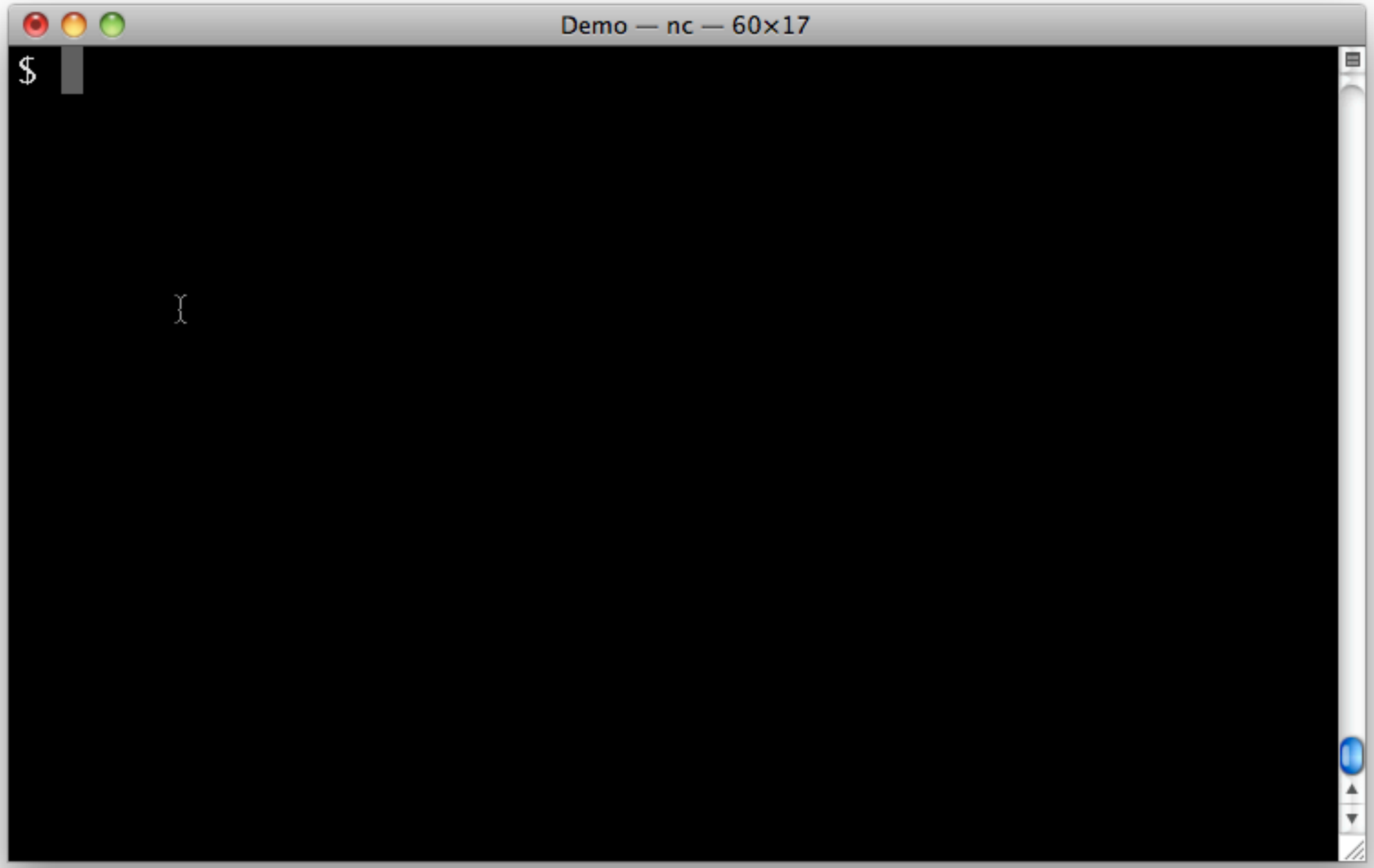
Scripting with ZFS

AEROSPACE

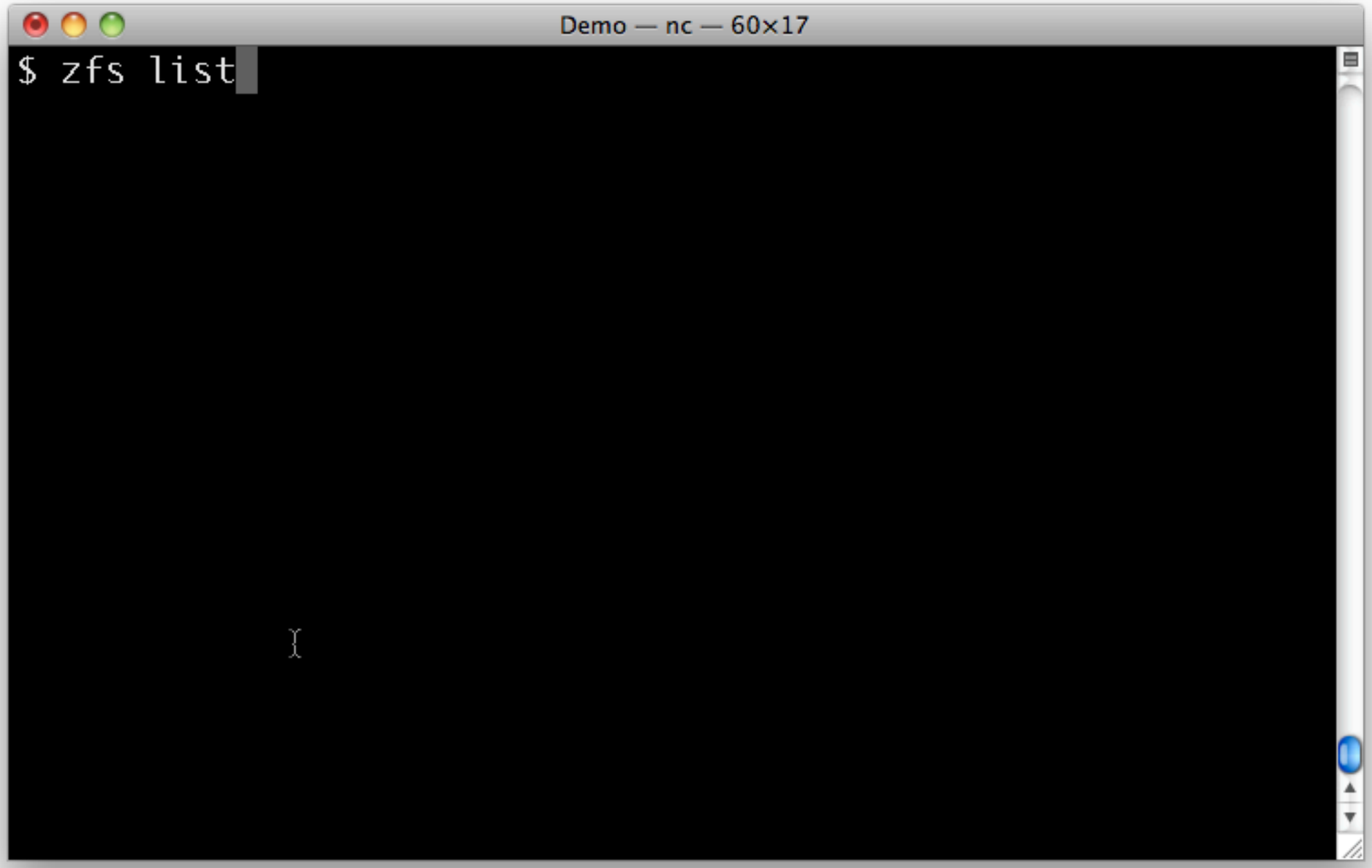
That's it for my concrete examples. For the replication I suggest looking over the script in the paper. I hope to release an updated version of the script soon with our latest improvements.
Now for some tidbits on ZFS and scripting.

Default output is human friendly, not machine friendly

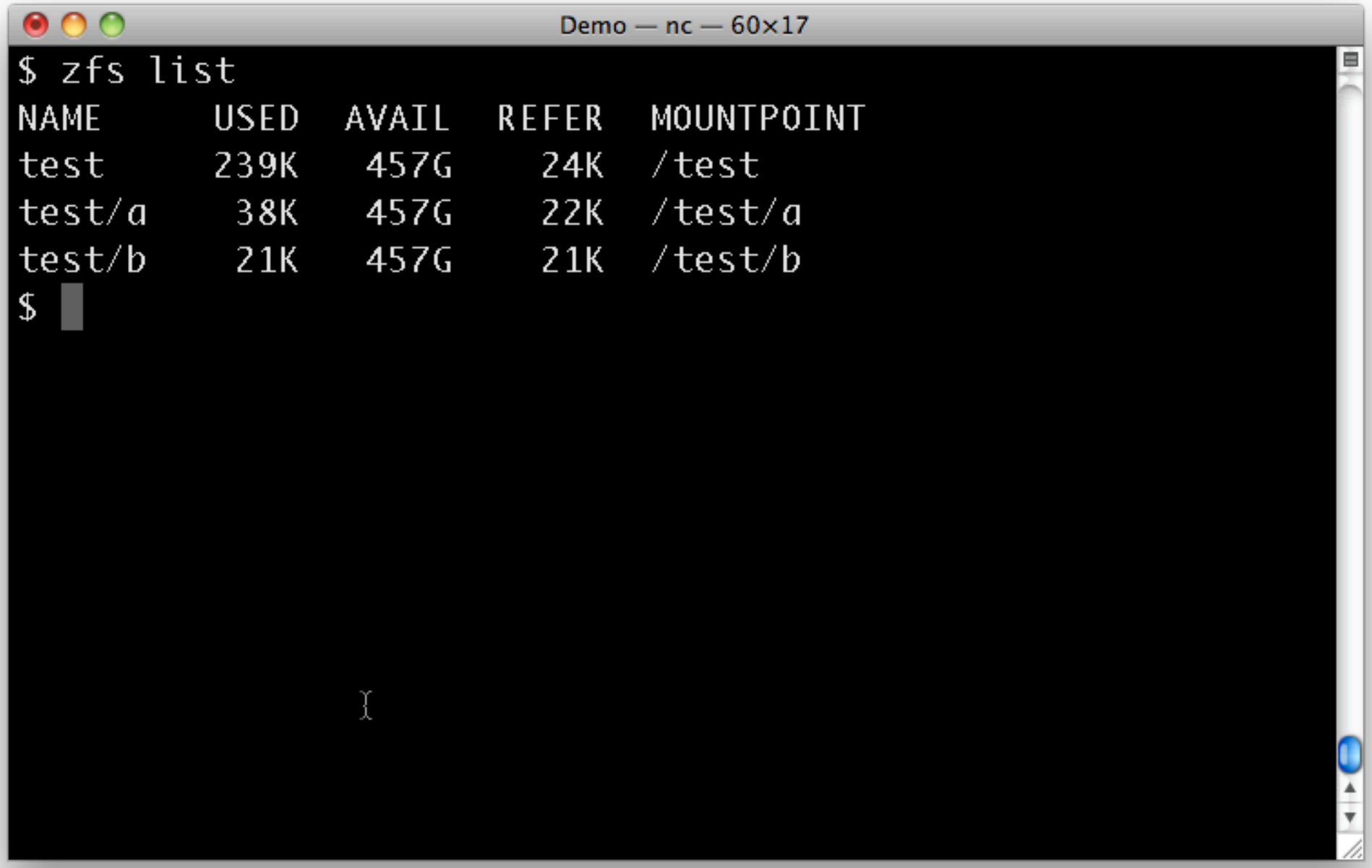
AEROSPACE



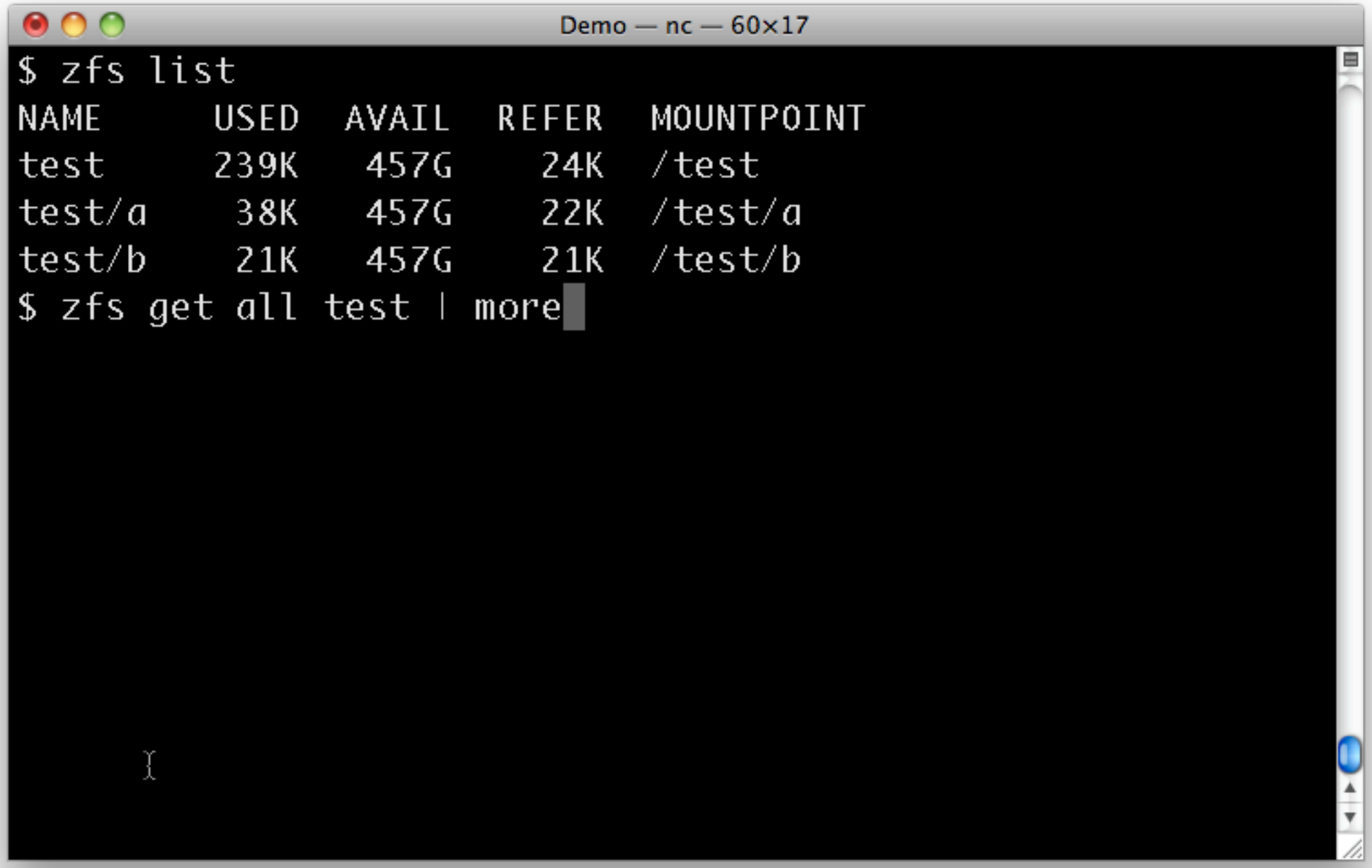
AEROSPACE



AEROSPACE


Demo — nc — 60×17
$ zfs list
NAME   USED AVAIL REFER MOUNTPOINT
test   239K   457G   24K /test
test/a   38K   457G   22K /test/a
test/b   21K   457G   21K /test/b
$

AEROSPACE


Demo — nc — 60×17
$ zfs list
NAME   USED AVAIL REFER MOUNTPOINT
test   239K   457G   24K /test
test/a   38K   457G   22K /test/a
test/b   21K   457G   21K /test/b
$ zfs get all test | more

AEROSPACE

Demo — nc — 60×17

```
NAME  PROPERTY              VALUE                  SOURCE
test  type                  filesystem             -
test  creation              Sat Feb 26 21:54 2011  -
test  used                  239K                   -
test  available             457G                   -
test  referenced            24K                    -
test  compressratio         1.00x                  -
test  mounted               yes                    -
test  quota                 none                   default
test  reservation           none                   default
test  recordsize            128K                   default
test  mountpoint            /test                  default
test  sharenfs              off                    default
test  checksum              on                     default
test  compression           off                    default
test  atime                 on                     default
--More--(byte 901)
```

AEROSPACE

Demo — nc — 60×17

```
test  devices               on                    default
test  exec                  on                    default
test  setuid                on                    default
test  readonly              off                   default
test  jailed                off                   default
test  snapdir               hidden                default
test  aclmode               groupmask             default
test  aclinherit            restricted            default
test  canmount              on                    default
test  shareiscsi            off                   default
test  xattr                 off                   temporary
test  copies                1                     default
test  version               4                     -
test  utf8only              off                   -
test  normalization         none                  -
test  casesensitivity       sensitive             -
--More--(byte 1823)
```

AEROSPACE

Demo — nc — 60×17

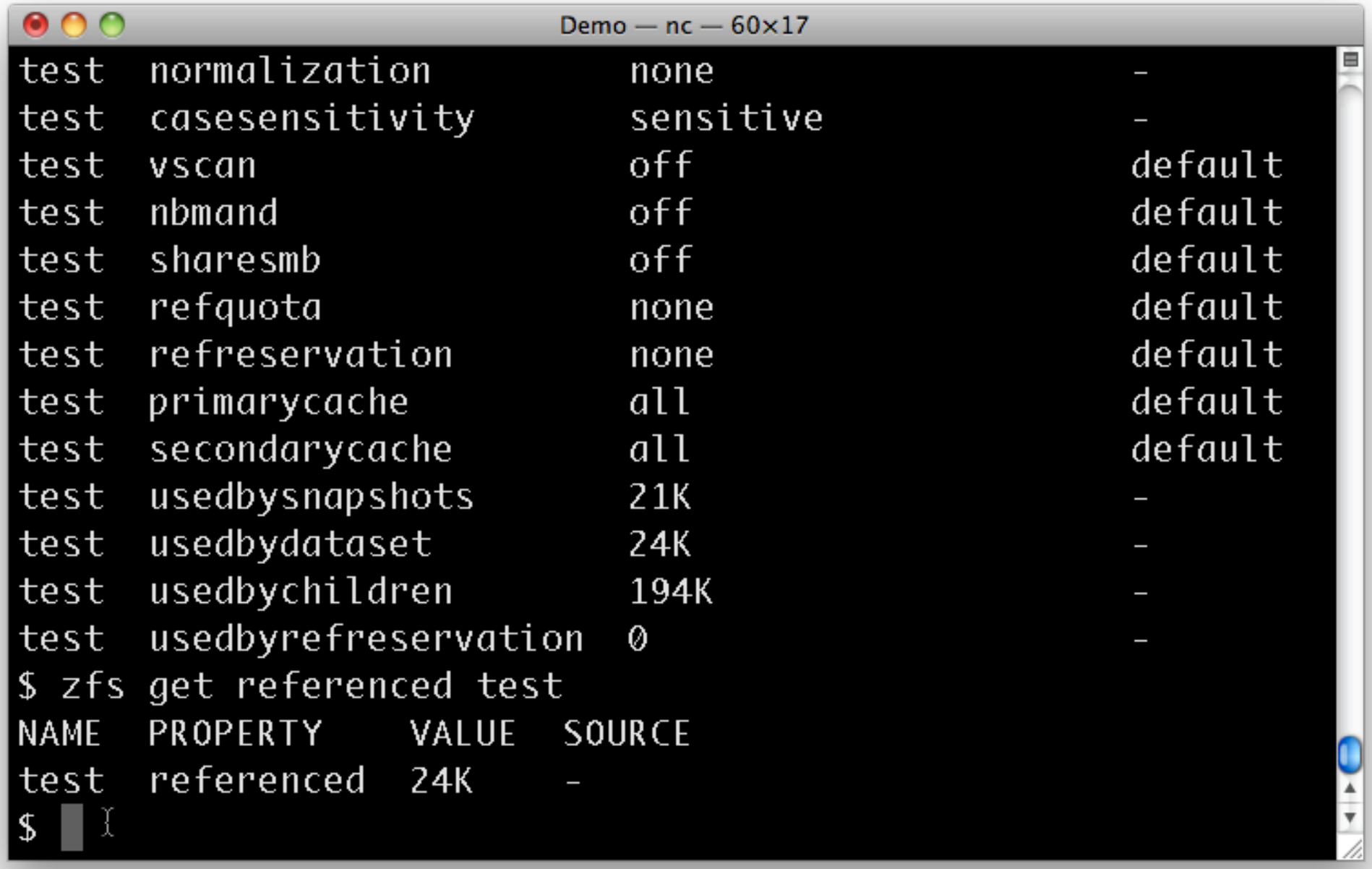
```
test  copies                1                    default
test  version               4                    -
test  utf8only              off                  -
test  normalization         none                 -
test  casesensitivity       sensitive            -
test  vscan                 off                  default
test  nbmand                off                  default
test  sharesmb              off                  default
test  refquota              none                 default
test  refreservation        none                 default
test  primarycache          all                  default
test  secondarycache        all                  default
test  usedbysnapshots       21K                  -
test  usedbydataset         24K                  -
test  usedbychildren        194K                 -
test  usedbyrefreservation  0                    -
$
```

AEROSPACE

Demo — nc — 60×17

```
test  copies                1                    default
test  version               4                    -
test  utf8only              off                  -
test  normalization         none                 -
test  casesensitivity       sensitive            -
test  vscan                 off                  default
test  nbmand                off                  default
test  sharesmb              off                  default
test  refquota              none                 default
test  refreservation        none                 default
test  primarycache          all                  default
test  secondarycache        all                  default
test  usedbysnapshots       21K                  -
test  usedbydataset         24K                  -
test  usedbychildren        194K                 -
test  usedbyrefreservation  0                    -
$ zfs get referenced test
```

AEROSPACE


Demo — nc — 60×17
test normalization none -
test casesensitivity sensitive -
test vscan off default
test nbmand off default
test sharesmb off default
test refquota none default
test refreservation none default
test primarycache all default
test secondarycache all default
test usedbysnapshots 21K -
test usedbydataset 24K -
test usedbychildren 194K -
test usedbyrefreservation 0 -
$ zfs get referenced test
NAME PROPERTY VALUE SOURCE
test referenced 24K -
$

AEROSPACE

Demo — nc — 60×17

```
test  normalization         none                  -
test  casesensitivity       sensitive             -
test  vscan                 off                   default
test  nbmand                off                   default
test  sharesmb              off                   default
test  refquota              none                  default
test  refreservation        none                  default
test  primarycache          all                   default
test  secondarycache        all                   default
test  usedbysnapshots       21K                   -
test  usedbydataset         24K                   -
test  usedbychildren        194K                  -
test  usedbyrefreservation  0                     -
$ zfs get referenced test
NAME  PROPERTY    VALUE  SOURCE
test  referenced  24K    -
$ zfs get -H referenced test
```

AEROSPACE

Demo — nc — 60×17

```
test  vscan                 off                    default
test  nbmand                off                    default
test  sharesmb              off                    default
test  refquota              none                   default
test  refreservation        none                   default
test  primarycache          all                    default
test  secondarycache        all                    default
test  usedbysnapshots       21K                    -
test  usedbydataset         24K                    -
test  usedbychildren        194K                   -
test  usedbyrefreservation  0                      -
$ zfs get referenced test
NAME  PROPERTY    VALUE  SOURCE
test  referenced  24K    -
$ zfs get -H referenced test
test    referenced      24K     -
$
```

Demo — nc — 60×17

```
test  vscan                off                    default
test  nbmand               off                    default
test  sharesmb             off                    default
test  refquota             none                   default
test  refreservation       none                   default
test  primarycache         all                    default
test  secondarycache       all                    default
test  usedbysnapshots      21K                    -
test  usedbydataset        24K                    -
test  usedbychildren       194K                   -
test  usedbyrefreservation 0                      -
$ zfs get referenced test
NAME  PROPERTY    VALUE  SOURCE
test  referenced  24K    -
$ zfs get -H referenced test
test    referenced      24K     -
$ zfs get -H -o value referenced test
```

AEROSPACE

Demo — nc — 60×17

```
test  sharesmb              off                   default
test  refquota              none                  default
test  refreservation        none                  default
test  primarycache          all                   default
test  secondarycache        all                   default
test  usedbysnapshots       21K                   -
test  usedbydataset         24K                   -
test  usedbychildren        194K                  -
test  usedbyrefreservation  0                     -
$ zfs get referenced test
NAME  PROPERTY    VALUE  SOURCE
test  referenced  24K    -
$ zfs get -H referenced test
test    referenced      24K     -
$ zfs get -H -o value referenced test
24K
$ 
```

AEROSPACE

Demo — nc — 60×17

```
test  sharesmb              off                    default
test  refquota              none                   default
test  refreservation        none                   default
test  primarycache          all                    default
test  secondarycache        all                    default
test  usedbysnapshots       21K                    -
test  usedbydataset         24K                    -
test  usedbychildren        194K                   -
test  usedbyrefreservation  0                      -
$ zfs get referenced test
NAME  PROPERTY    VALUE  SOURCE
test  referenced  24K    -
$ zfs get -H referenced test
test    referenced      24K     -
$ zfs get -H -o value referenced test
24K
$ zfs get -H -p -o value referenced test
```

AEROSPACE

Demo — nc — 60×17

```
test  refreservation         none                           default
test  primarycache           all                            default
test  secondarycache         all                            default
test  usedbysnapshots        21K                            -
test  usedbydataset          24K                            -
test  usedbychildren         194K                           -
test  usedbyrefreservation   0                              -
$ zfs get referenced test
NAME  PROPERTY    VALUE  SOURCE
test  referenced  24K    -
$ zfs get -H referenced test
test    referenced      24K     -
$ zfs get -H -o value referenced test
24K
$ zfs get -H -p -o value referenced test
24576
$
```

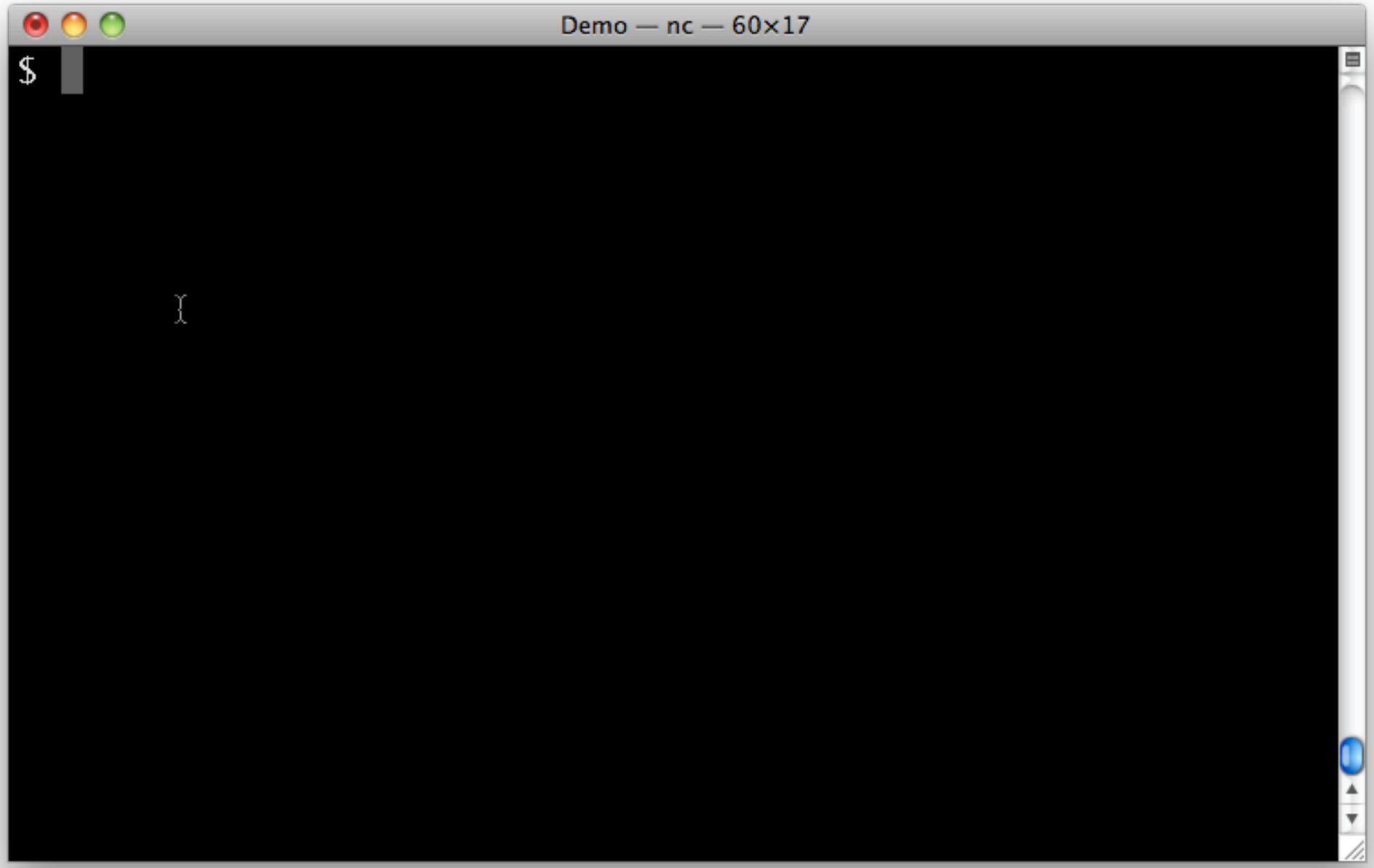
AEROSPACE

-H and -o options to “zfs list” work similarly

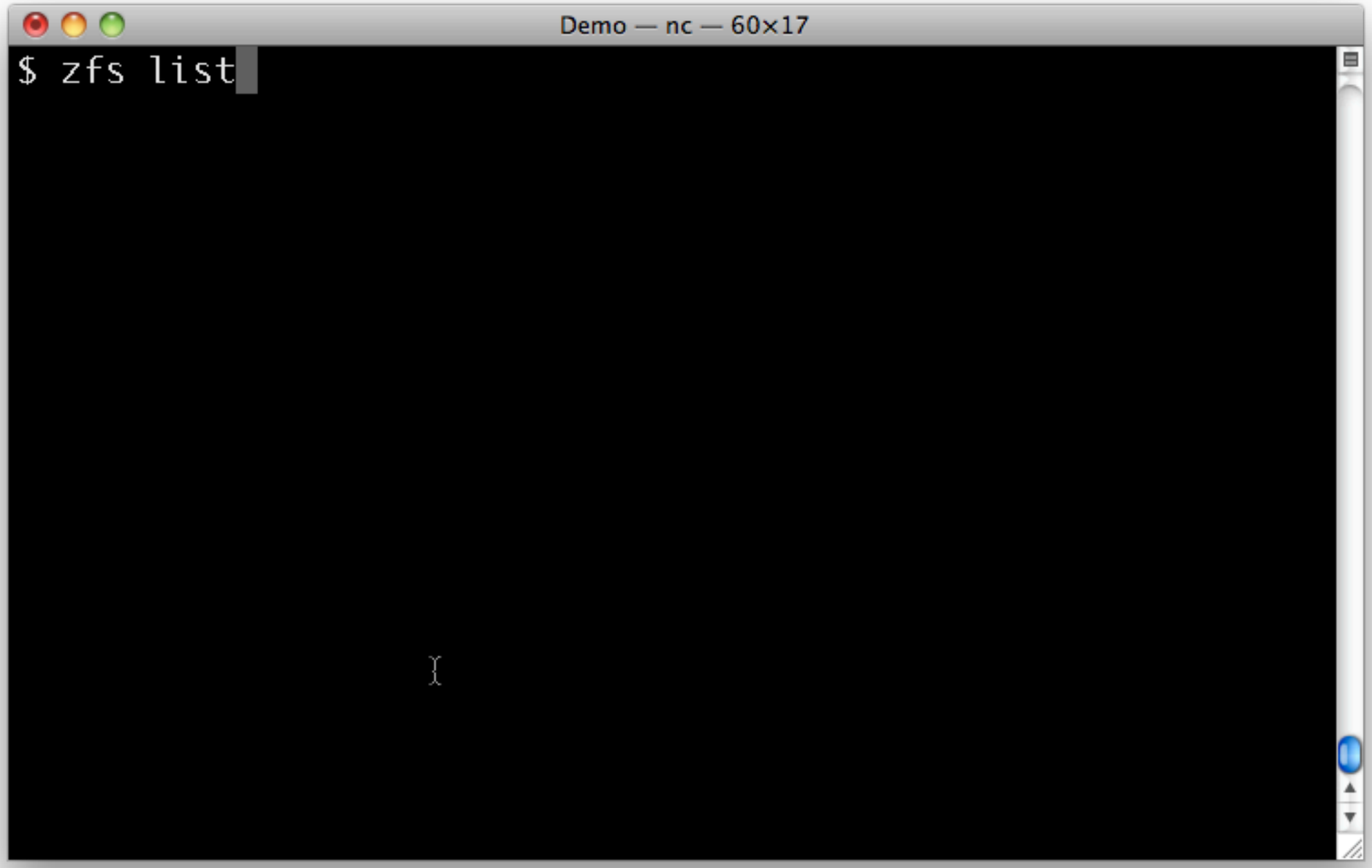
Restricting zfs list output

AEROSPACE

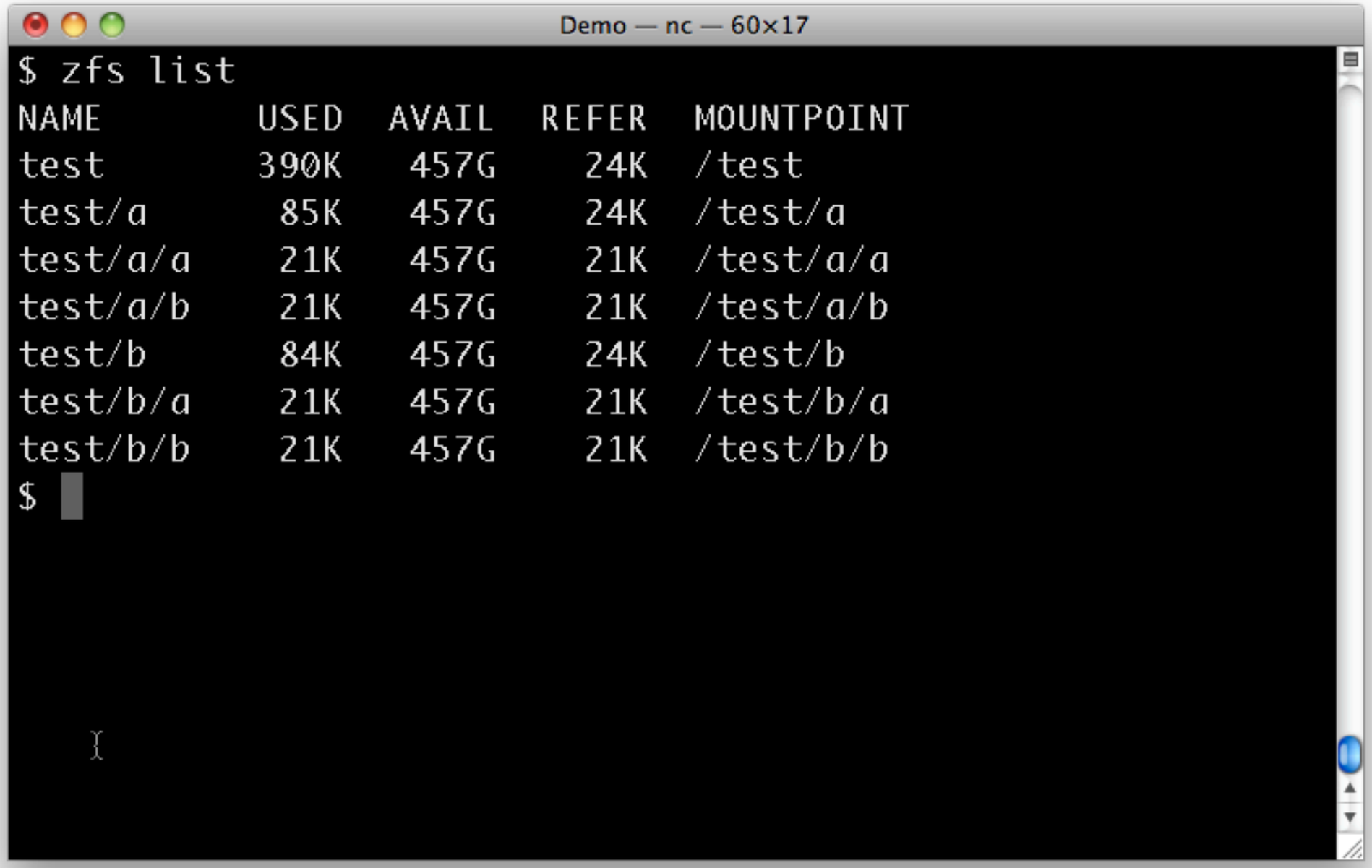
One other thing that is useful to do is to restrict the output of “zfs list”



AEROSPACE

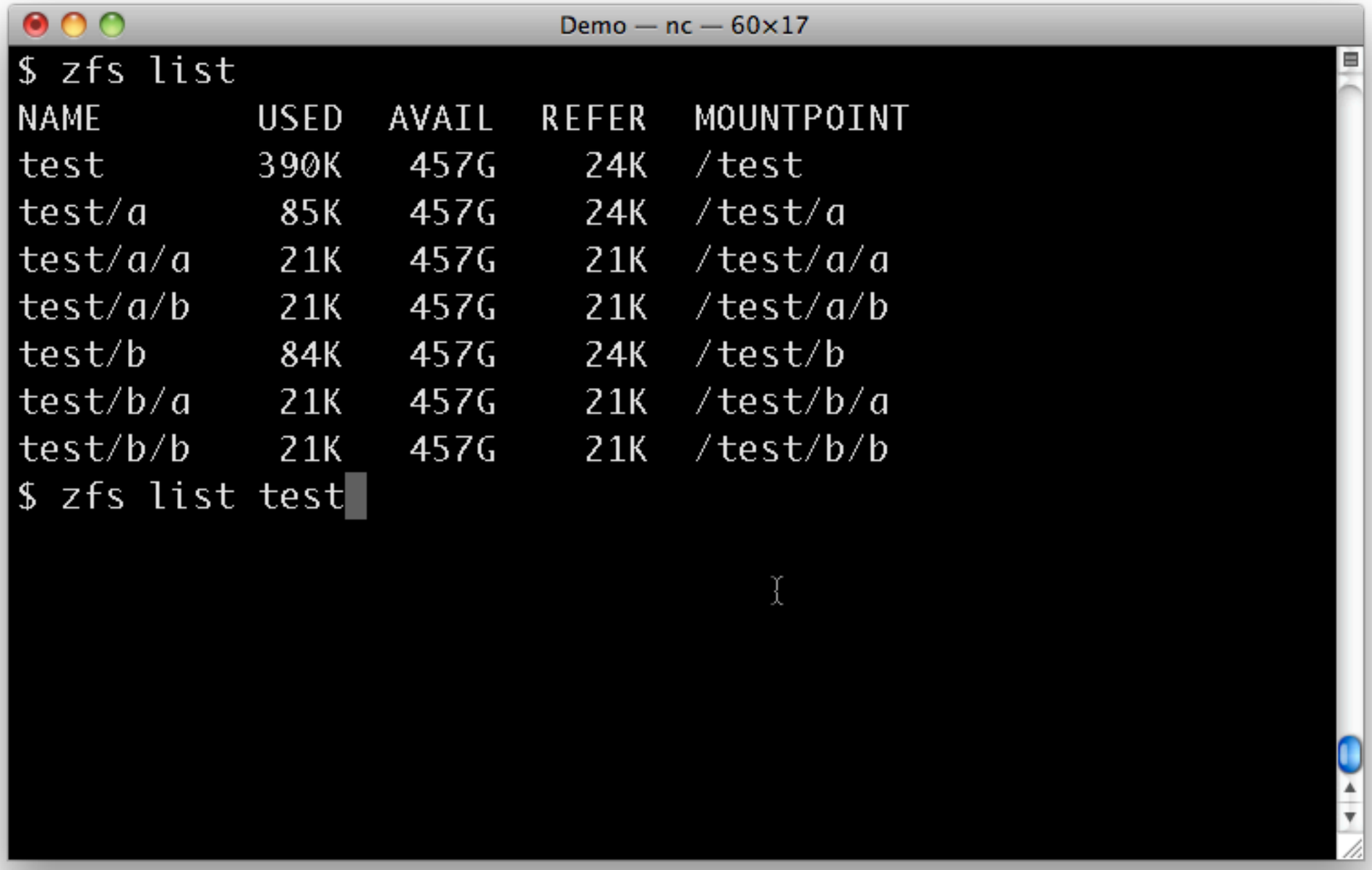


AEROSPACE



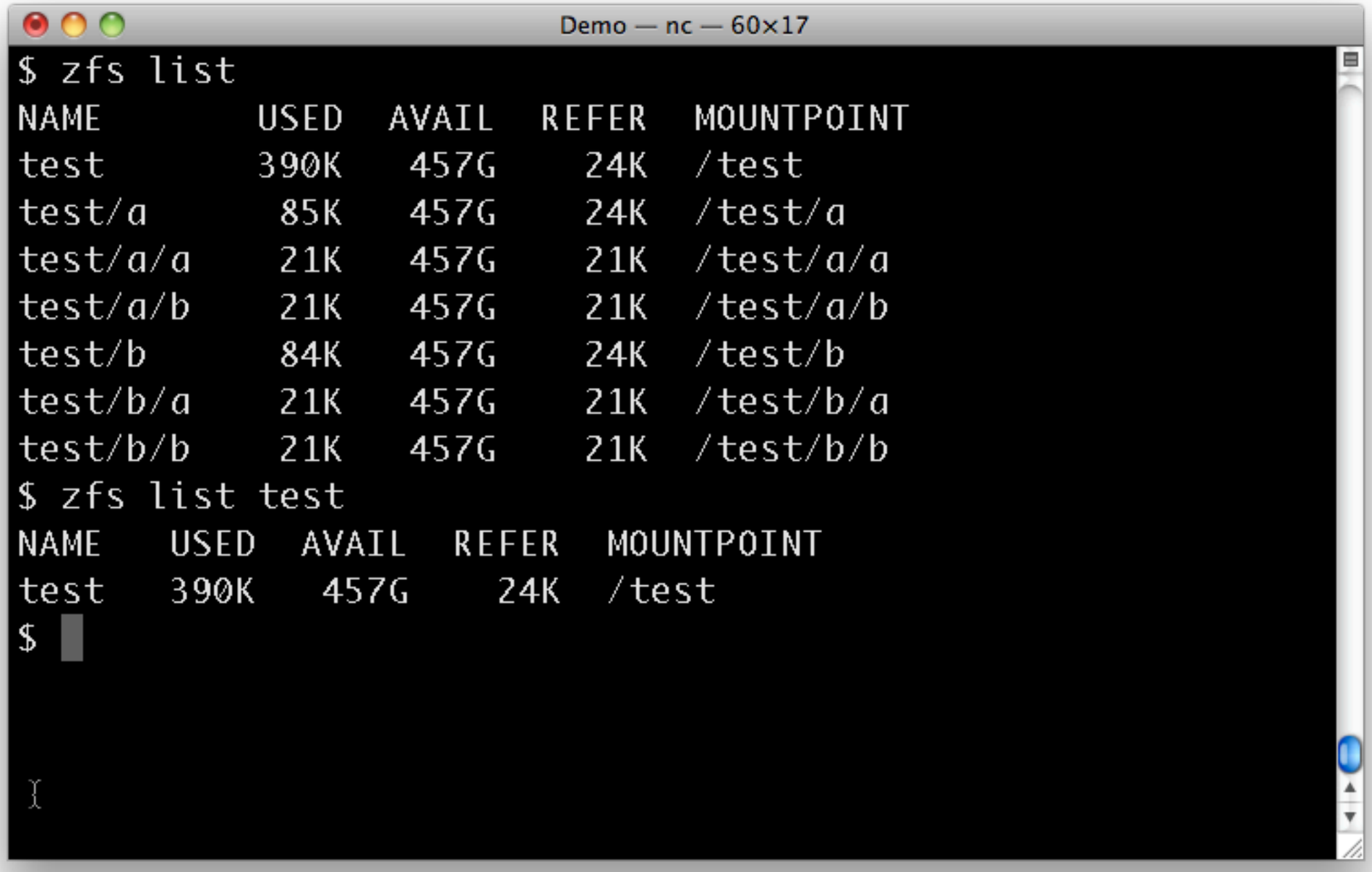
AEROSPACE

For example you might have a file system hierarchy like this.



AEROSPACE

You can of course show only a single file system


Demo — nc — 60×17
$ zfs list
NAME   USED AVAIL REFER MOUNTPOINT
test   390K   457G   24K /test
test/a   85K   457G   24K /test/a
test/a/a   21K   457G   21K /test/a/a
test/a/b   21K   457G   21K /test/a/b
test/b   84K   457G   24K /test/b
test/b/a   21K   457G   21K /test/b/a
test/b/b   21K   457G   21K /test/b/b
$ zfs list test
NAME   USED AVAIL REFER MOUNTPOINT
test   390K   457G   24K /test
$

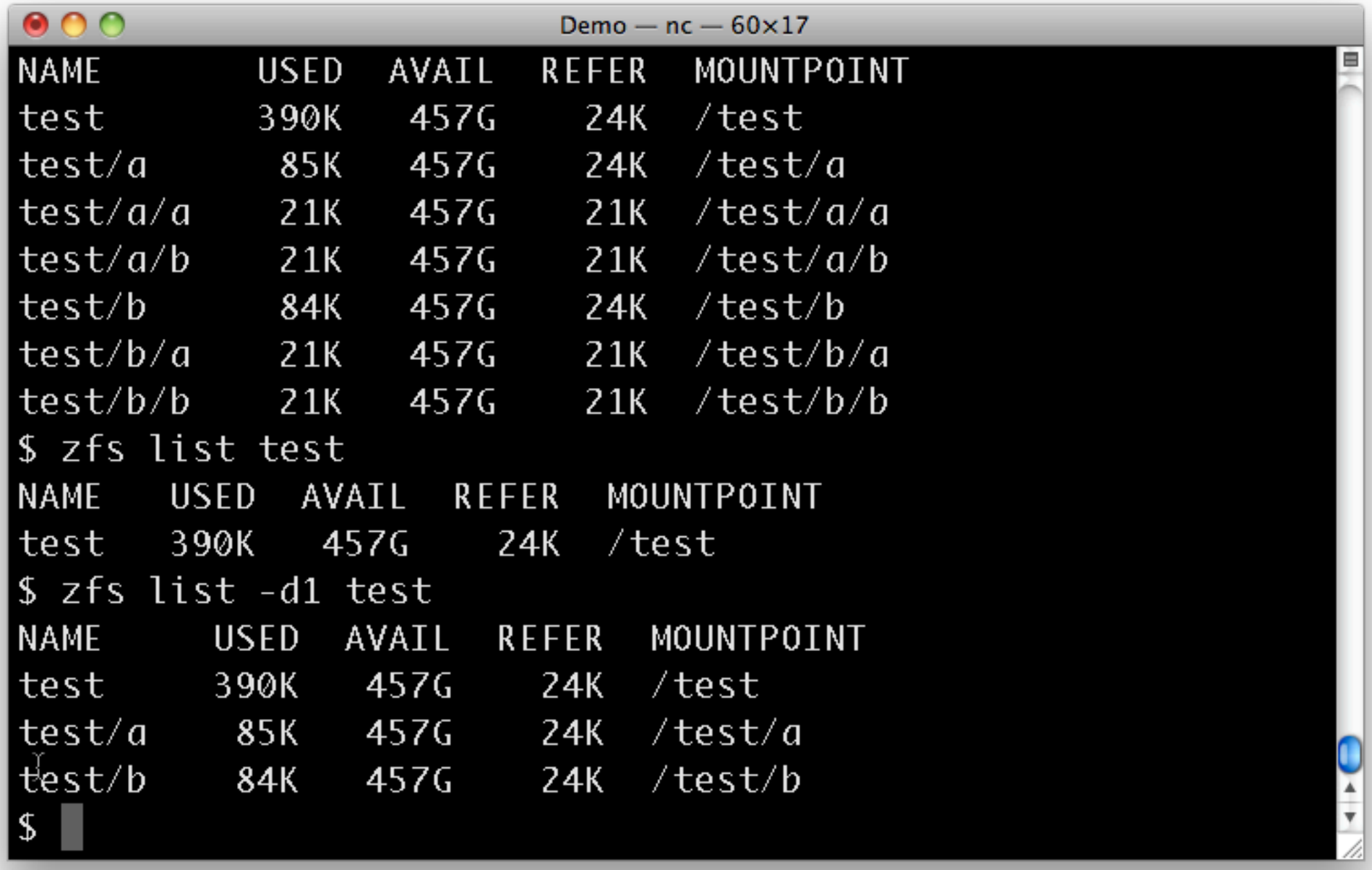
Demo — nc — 60×17

```
$ zfs list
NAME        USED  AVAIL  REFER  MOUNTPOINT
test        390K   457G    24K  /test
test/a       85K   457G    24K  /test/a
test/a/a     21K   457G    21K  /test/a/a
test/a/b     21K   457G    21K  /test/a/b
test/b       84K   457G    24K  /test/b
test/b/a     21K   457G    21K  /test/b/a
test/b/b     21K   457G    21K  /test/b/b
$ zfs list test
NAME   USED  AVAIL  REFER  MOUNTPOINT
test   390K   457G    24K  /test
$ zfs list -d1 test
```

AEROSPACE

Monday, September 26, 2011

or you might want to only show a limited depth



You would often do these things in combination with -H and -o

ZFS Scripting Deficiencies

- Need to get all props if you use variable prop names like aerosource.aero.org:lastsnap:*host*
- Need to scan "zfs list" output to get lists of snapshots, filesystems, etc
- Adding fnmatch() support to the get and list commands would help

AEROSPACE

Scripting support is not quite as good as it could be.
More control over listed values would be helpful -D# for specific depth?

Conclusions

AEROSPACE

We're using ZFS in a number of interesting ways and I think it's making our lives easier.
I hope this talk inspires you to think about more interesting things you can do with ZFS.
There are a lot of basic scripts out there, but there's room for more interesting frameworks.

Questions?

AEROSPACE

Disclaimer

- All trademarks, service marks, and trade names are the property of their respective owners.

AEROSPACE