The History and Future of Core Dumps in FreeBSD

Sam W. Gwydir, Texas A&M University
sam@samgwydir.com

February 19, 2017

Who Am I

My name is Sam Gwydir

- I've used *nix for a little over 10 years
- OpenBSD and later FreeBSD around 2013

Overview

There are three main sections of this talk:

- The Past
  - Research UNIX v5 through FreeBSD 12-CURRENT
- The Present
  - FreeBSD
  - Illumos
  - Mac OS X
  - Backtrace.io
- The Future
  - Core Dump Extensions

Background

What is a Core Dump?

Core Dump A machine readable form of the state of a machine at some point in time, usually after a `panic(9)`.

Why do I want them?

- Operators can debug crashes offline; a production machine can go back online
- Allows archiving of crash data for later comparison
- Useful when crashes aren't predicted i.e. production

The History

- A Comprehensive list of architectures that support core dumps, and related features
- Essentially the odyssey that `doadump()` has been through since UNIX/32V
- Starts at 5th Edition Research UNIX's `crash(8)`
- Ends at FreeBSD 12-CURRENT's Encrypted Dump

Operating Systems

- Core Dump Features in:
  - FreeBSD
  - Illumos
  - Mac OS X

FreeBSD

- Full Dump
  - A classic core dump – the full contents of memory at the time of a crash
- Minidump
  - Introduced in FreeBSD 6.2
  - Contains only memory pages in use by kernel
  - Much smaller than the full contents of memory
- Text Dump
  - Introduced in FreeBSD 7.1
  - Custom ddb scripting in lieu of a dump
  - A small tar of text files

How to take a Core Dump in FreeBSD

```
# mkdir /var/crash  # create the dumpdev
# chmod 700 /var/crash
# swapinfo # find a suitable swap partition
# dumpon -v /dev/da0p2
# sysctl debug.kdb.panic=1
```

- Notes: - You are purposely panicking your machine. Save your stuff.

How Core Dumps Work

- important part: written backwards

Illumos

Not technically a BSD but the features are important

- Online dump size estimation
  - Includes different calculations for settings, e.g. compression
- Compressed Dump
  - gzip compression
- Dump to Swap on zvol
  - Versatility of zvols vs partitions
- Live Dump
  - Useful for production machines where interactive debugging is not possible
  - Especially for debugging hangs

Mac OS X

- Compressed Dump
  - gzip compression
- netdump
  - Using a modified `tftpd(8)` from FreeBSD!

Tools

- Backtrace.io

Extant Core Dump Extensions

- Compressed Dump
  - 6:1 to 14:1 compression ratio
  - A 32 GB Core becomes 5.34 GB!
- Core Dump Extensions not in FreeBSD
  - netdump
    - Holding on since FreeBSD 4.x
  - Dump Size Estimation
    - `minidumpsz` for FreeBSD 10 and 11
  - Modular Dump Code
    - Embedded systems
    - Rod Grimes

Proposed Core Dump Extensions

- Dump to Swap on zvol
  - The ZFS way
- Live Dump
  - Gauge interest

Links

Thank you for coming!

- github.com/gwydirsam/bsd-coredump-history
- github.com/dspinellis/unix-history-repo