Fifteen Years and Fifteen Minutes: Applying Occam's Razor to FreeBSD with OccamBSD

@MichaelDexter

NYC*BUG 2021-02-03
“Will think critically for food” – Occam’s Razor

“Pluralitas non est ponenda sine necessitate”

“Plurality should not be posited without necessity”

“The law of parsimony”

“The simplest explanation is usually the right one”
“Will think critically for food” – Occam’s Razor

“Simple theories are easier to verify, simple solutions are easier to execute”

“No more assumptions should be made than are necessary”

“Keep It Simple”

“Less is more, more (1) or less (1)”
Berkeley Software Distribution (BSD)

A largely-permissively licensed unified Unix operating system originating at the University of Berkeley and expanded by many, many people around the world
“4.3BSD in college was clearly a behind-the-camera view of the computer. *That* is a very familiar concept.”

“Coherent Unix was interesting but limited.”

“UnixWare looks great on paper.”

“The RedHat 5.1 documentation does not match the product.”
Background, briefly

“RedHat 5.2 is a great all-around Unix clone with a packaged base. Rejoice!”

“My search is over!”
Background, briefly

“RedHat 6.0 turned a good Unix clone into a stunningly-bad Windows clone with GNOME.”

“Binary updates and upgrades will come someday.”

“RPM HELL” and `linuxconf`
Solution? Move to Latvia

The pitch for *BSD in magazines and the Walnut Creek CD-ROM catalog was compelling but I didn’t know anyone.

The “assman@” chroot jail did not sound workable.

\texttt{jail(8)} however sounds like a solution
jail(8) on 4.7/4.8 is very promising!

jail(8) on 5.0 with jls(1) and jexec(8) are ALL THE THINGS

“My search is finally over!”
The BSD/FreeBSD Value Proposition

Before ZFS, DTrace, CTL, \texttt{gpart(8)}, PF, and...

Largely Permissively-Licensed

Unified OS/Unified Build

\texttt{jail(8)}
An Excellent Narrative

“A permissively-licensed OS that is perfect for embedded products and the things we will call ‘containers’ and ‘IoT’ many years from now.”

“Simply set a few build and *KERNCONF* options to reduce the system to its bare minimum and add what you need.”

“Just like ‘block all’ on your packet filter.”

AWESOME!!! Just add a hypervisor!
About that...

FreeBSD 5.0 with the j-tools delivered uptimes measured in days
Those build options didn't quite work

That was early 2003
Interlude

OpenBSD SysJail with Kristaps
NetBSD/Xen for mult and general use
Patience
More patience...
BHyVe
Updated Narrative

The highly-flexible, permissively-licensed OS with a “container” mechanism now has a hypervisor.

TIME TO REVISIT THOSE BUILD OPTIONS

That was 2012 and narrative remains valid.
Build Options

man src.conf

WITHOUT_ACCT

Set to not build process accounting tools such as accton(8) and sa(8).

WITHOUT_ACPI

Set to not build acpiconf(8), acpidump(8) and related programs.

WITH_

230+ Options in total for 13.0
Build Options

“Are you absolutely sure I need to build CLANG/LLVM to test this one thing?”

“This thing we are shooting into space (er, digital sign) doesn’t speak TCP/IP...”

(Insert clever yet discouraged-for-years use case)
Build Options

“The resulting system will be totally unrecognizable!”

Actually...
The Minimum OS

Documenting it? – Start here
Auditing it? – Start here
Learning it? – Start here
Fast forward to today, early 2021

Years of testing, reporting, nudging, and repairing has resulted in massive progress

FreeBSD 13.0-RELEASE will... (did we make it?)
OccamBSD: An application of Occam’s Razor to FreeBSD
a.k.a. "super svelte stripped down FreeBSD"

Fifteen Eighteen* Years in the Making

*Insert vote/drive/drink/buy a gun jokes here
OccamBSD

occambsd.sh

/usr/src/tools/tools/
build_option_survey/listallopts.sh
The Build Option Survey (BOS) August, 2005

/usr/src/tools/tools/
build_option_survey/
listallopts.sh
option_survey.sh
reduce.sh
mkhtml.sh
The Build Option Survey (BOS) August, 2005

<table>
<thead>
<tr>
<th>WITHOUT_CXGBETOOL=foo</th>
<th>MK_CXGBETOOL = no</th>
<th>+0</th>
<th>-0</th>
<th>*1577</th>
<th>3426728</th>
<th>-100</th>
<th>+0</th>
<th>-3</th>
<th>*1</th>
<th>3423676</th>
<th>-3152</th>
<th>+0</th>
<th>-3</th>
<th>*1578</th>
<th>3423576</th>
<th>-3252</th>
</tr>
</thead>
<tbody>
<tr>
<td>WITHOUT_CXX=foo</td>
<td>MK_CLANG = no</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MK_CLANG_FULL = no</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MK_CXX = no</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MK_GOOGLETEST = no</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MK_LLVM_COV = no</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MK_TESTS = no</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>failed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WITHOUT_DEBUG_FILES=foo</td>
<td>MK_DEBUG_FILES = no</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>failed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WITHOUT_DIALOG=foo</td>
<td>MK_BSDINSTALL = no</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MK_DIALOG = no</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>failed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

callfortesting.org/results/
The BOS-ng December, 2019

github.com/michaeldexter/bos-ng
calctimes.sh
findfaults.sh
With the hard parts out of the way...

```
sh /usr/src/tools/tools/build_option_survey/listallopts.sh | \
  grep -v WITH_ | sed 's/$/=YES/' | \ 
  grep -v WITHOUT_AUTO_OBJ | grep -v WITHOUT_UNIFIED_OBJDIR | \ 
  grep -v < INSERT BROKEN OPTIONS HERE > | \ 
  grep -v WITHOUT_BOOT | grep -v WITHOUT_LOADER_LUA | \ 
  grep -v WITHOUT_LOCALES | \ 
  grep -v WITHOUT_ZONEINFO | \ 
  grep -v WITHOUT_VI \ 
> /etc/src.conf
```
With the hard parts out of the way...

/tmp/occambsd

src.conf  OccamBSD src.conf
OCCAMBSD  OccamBSD kernel configuration file

bhyve-kernel  bhyve kernel directory
bhyve-mnt  bhyve disk image mount point
bhyve.raw  bhyve disk image with kernel
load-bhyve-vmm-module.sh  Script to load vmm.ko
load-bhyve-disk-image.sh  Script to load bhyve kernel from disk image
load-bhyve-directory.sh  Optional script to load bhyve kernel from directory
boot-occambsd-bhyve.sh  Script to boot bhyve
destroy-occambsd-bhyve.sh  Script to clean up the bhyve virtual machine
With the hard parts out of the way...

/tmp/occambsd

xen-kernel                    Xen kernel directory
xen-mnt                       Xen disk image mount point
xen.raw                       Xen disk image with kernel
xen-occambsd.cfg              Xen disk image boot configuration file
xen-occambsd-kernel.cfg       Xen directory boot configuration file
boot-occambsd-xen.sh          Script to boot Xen kernel from disk image
boot-occambsd-xen-kernel.sh   Script to boot Xen kernel from directory
destroy-occambsd-xen.sh       Script to clean up the Xen virtual machine

/tmp/occambsd/jail                       Jail root directory
/tmp/occambsd/jail.conf                  Jail configuration file
/tmp/occambsd/boot-occambsd-jail.sh      Script to boot the jail(8)
Demon
Michael W(Warren) Lucas
@mwlauthor

Replying to @michaeldexter

I have wanted this to exist since... since about the same time. But making it happen would have been work, and y'all know I'm lazy.

So: thank you!

3:58 PM · Jan 22, 2021 · Twitter Web App
Olivier Cochard @ocochardlabbe

Replies to @michaeldexter

Thanks! This could be a main component of my idea of a git-bisec-regression-test script: Input will be regression test name and hashes range, and it will quickly build world & kernel, and run it under bhyve to get regression test result.

1:38 PM · Jan 25, 2021 · Twitter Web App
Initial Feedback

@michaeldexter have you considered adding that functionality to poudriere image, perhaps as a 'hypervisor' target? Because that would be INCREDIBLE.
Choose Your Own Adventure

Should this...
Offer a choice of bhyve, Xen, or Jail?
Make rebuilding world or kernel easier?

< Your amazing idea here >
Observations

/usr/share/locale/C.UTF-8 (37K) is all you need
/usr/share/zoneinfo/UTC (512B) is...

Separate those from their options?
Related: up.bsd.lv

FreeBSD CURRENT and STABLE Binary Upgrades

"Because tracking CURRENT shouldn't be a rite of passage"

Disclaimer

This proof-of-concept is not a publication of FreeBSD.

Description

up.bsd.lv is a proof-of-concept of binary updates using `freebsd-update(8)` for FreeBSD 13.0-CURRENT and 12-STABLE to facilitate the exhaustive testing of FreeBSD and the bhyve hypervisor and OpenZFS 2.0, and to help elevate ARM64 to Tier 1 status. Updates are based on the SVN revisions of official FreeBSD Release Engineering weekly snapshots.

Requirements

Upgrades use SSL to the extent possible and thus require the `ca_root_nss` package, `certctl(8)`, or another certificate bundle to support the retrieval of two `.ssl` files. Note that an accurate system time and date may be required for proper SSL handling.

Upgrades can be performed on 12.1 RELEASE or updated systems, or with a provided `disc1.iso` or memstick image.
OccamBSD’s 1650+ line older sibling

```
cd /usr/src ; make buildworld ; make buildkernel
  cd release ; make release
```

*With many, many options.*

Related: headbanger.sh
Related: headbanger.sh

sh headbanger.sh -s "/build/7ae27c2d6c4/src" \  
-o "/build/7ae27c2d6c4/obj" \  
-l "/build/7ae27c2d6c4/log" -r 7ae27c2d6c4 \  
-g "/pub/FreeBSD/git/src" \  
-t arm64 -T aarch64 -R "10.0.0.40" -k "GENERIC-NODEBUG" -u

sh headbanger.sh -O "WITHOUT_AUTO_OBJ WITHOUT_UNIFIED_OBJDIR WITHOUT_INSTALLLIB WITHOUT_BOOT WITHOUT_LOADER_LUA WITHOUT_LOCALES WITHOUT_ZONEINFO WITHOUT_VI" -K /tmp/OCCAM"
What about `nanobsd.sh`?

```bash
#!/bin/sh

# Copyright (c) 2005 Poul-Henning Kamp.
...

00:00:00 ## Construct build make.conf
(/usr/obj/nanobsd.full/make.conf.build)

00:00:00 ## run buildworld

00:00:01 ### log: /usr/obj/nanobsd.full/_bw

NANO_PMAKE="make -j 3"
```
What about `nanobsd.sh`?

--- ASTMatchers/Dynamic/Registry.o ---
Killed
*** [ASTMatchers/Dynamic/Registry.o] Error code 137
make[6]: stopped in /usr/src/lib/clang/libclang
00:00:05 ## Construct build make.conf
(/usr/obj/nanobsd.full/make.conf.build)
00:00:05 ## run buildworld
00:00:05 ### log: /usr/obj/nanobsd.full/_.bw

1h43m5.37s real 6h23m19.11s user 11m45.98s sys
EPYC THANKS

Kyle Evans
Mitchell Horne
Ed Maste
NYC*BUG