Packaging proprietary software with Nix

Or, how I learned to stop worrying and start loving vmTools

George Huebner

2026-03-05

`whoami`

Nix on Darwin enjoyer

Jack of all trades who uses Nix to dabble in many different fields

Frustrated VCS user

Case study: Synopsys EDA toolchain

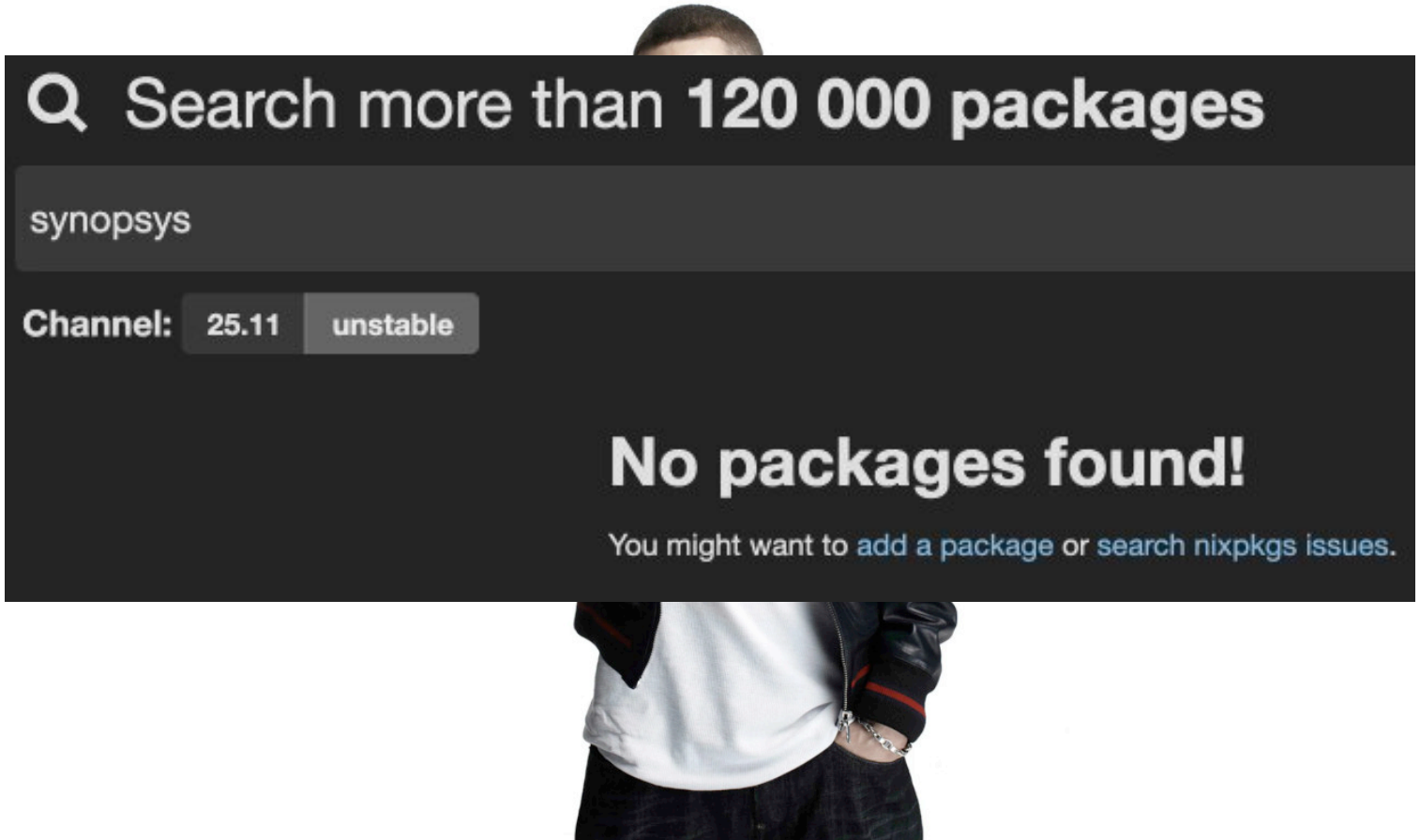
Tired of using my 8-core processor to SSH into a 1 vCPU VM to run synth

No source but nothing to compile, should be pretty straightforward

(It was not very straightforward)

Attempt #1

Attempt #1


Search more than 120 000 packages
synopsys
Channel:
25.11
unstable
No packages found!
You might want to add a package or search nixpkgs issues.

What does a program need to run?

If your program

- Doesn't use dynamic libraries
- Doesn't use unstable OS ABIs

and your machine

- Is the same OS (Linux) and can run the architecture/file type

Chances are, things will just work!

```
a.out: ELF 64-bit LSB executable, x86-64, version 1 (SYSV),
statically linked, with debug_info, not stripped
```

Round one: Spyglass (linter)

```
$ .../SPYGLASS_HOME/bin/sg_shell
.../SPYGLASS_HOME/bin/../lib/SpyGlass/standard-environment.sh: line 1428: tr: command not found
.../SPYGLASS_HOME/bin/../lib/SpyGlass/standard-environment.sh: line 1428: sort: command not found
.../SPYGLASS_HOME/bin/../lib/SpyGlass/standard-environment.sh: line 1428: sed: command not found
.../SPYGLASS_HOME/bin/../lib/SpyGlass/standard-environment.sh: line 1436: tr: command not found
.../SPYGLASS_HOME/bin/../lib/SpyGlass/standard-environment.sh: line 158: uname: command not found
.../SPYGLASS_HOME/bin/../lib/SpyGlass/standard-environment.sh: line 158: tr: command not found
.../SPYGLASS_HOME/bin/../lib/SpyGlass/standard-environment.sh: line 1432: dirname: command not found
.../SPYGLASS_HOME/bin/../lib/SpyGlass/standard-environment.sh: line 1433: dirname: command not found
...
spyglass:                A perl installation was expected at: `.../SPYGLASS_HOME/lib/multi-perl'
spyglass:                either `$SPYGLASS_HOME' was guessed incorrectly or the installation is
                         corrupted.
```

Round one: Spyglass (linter)

```
$ ldd .../SPYGLASS_HOME/lib/multi-perl/bin/../obj/Linux4/perl
        linux-vdso.so.1 (0x00007fdf2bc4e000)
        libnsl.so.1 => /nix/store/...-glibc-2.40-218/lib/libnsl.so.1 (0x00007fdf2bc2c000)
        libdl.so.2 => /nix/store/...-glibc-2.40-218/lib/libdl.so.2 (0x00007fdf2bc27000)
        libm.so.6 => /nix/store/...-glibc-2.40-218/lib/libm.so.6 (0x00007fdf2bb3f000)
        libcrypt.so.1 => not found
        libutil.so.1 => /nix/store/...-glibc-2.40-218/lib/libutil.so.1 (0x00007fdf2bb38000)
        libpthread.so.0 => /nix/store/...-glibc-2.40-218/lib/libpthread.so.0 (0x00007fdf2bb33000)
        libc.so.6 => /nix/store/...-glibc-2.40-218/lib/libc.so.6 (0x00007fdf2b92b000)
        /lib64/ld-linux-x86-64.so.2 => /nix/store/...-glibc-2.40-218/lib64/ld-linux-x86-64.so.2
        (0x00007fdf2bc50000)
```

“Defensive” coding

```
PATH="/usr/bin:/bin" export PATH
```

```
platform_species () {
    case `uname -sr | tr ' ' -` in
    Linux-2*) ...
    Linux-3*) ...
    Linux-4*) ...
    Linux-5*) ...
    SunOS-4*) ...  # oh... the pain
    esac
}
```

Technique: `buildFHSEnv`

Let the program remain blissfully ignorant

```
buildFHSEnv {
  targetPkgs = pkgs: (with pkgs; [
    glibc
    libxcrypt-legacy
    (deterministic-uname.override {
       modDirVersion = "4.18.0";
    })
  ]);
}
```

```
                SpyGlass Predictive Analyzer (R)
                     Synopsys TestMAX(TM)


     Version R-2020.12-SP1-1 for linux64 - Apr 27, 2021

To access quickstart manual, please use `man quickstart'

sg_shell>
```

Minimizing copy overhead

- Environment variables
  - → Eval-time `getEnv "XYZ"` vs run-time `"$XYZ"`
- Lazy trees (Determinate Nix only)
- If you must use `requireFile`, use `zstd`/`xz` with an aggressive `unpackPhase` (`tar`'s `--exclude` option)

Above all else, use a separate source derivation to avoid unnecessary rebuilds!

Round two: Verdi (waveform/circuit viewer)

Requires significantly more libraries, many of which are quite old (Qt3 libs)

We could…

- Use an old pinned version of nixpkgs for some/all packages
- ~~Roll our own derivations for missing packages~~

Technique: `patchelf`

```
patchelf --set-interpreter my-ld-linux-x86-64.so.2 a.out
```

```
nativeBuildInputs = [ autoPatchelf ];
runtimeDependencies = [ ... ];
postFixup = ''
  patchelf --remove-needed libQt3Support.so.4 $out/bin/a.out
'';
```

Idea: return to RHEL

Designed to run on RHEL, let's run it on RHEL

Can we get the benefits of `FROM rhel7` without losing reproducibility..?

dockerTools

Create a lightweight OCI container image with a Nix derivation

Slight problem: libraries from nixpkgs are too new

```
let
  baseImage = pkgs.dockerTools.pullImage {
    imageName = "centos";
    imageDigest = "sha256:be65f488b7764ad3638f236b7b515b3678369a5124c47b8d32916d6487418ea4";
    arch = "amd64";
  };
in
  pkgs.dockerTools.buildImage {
    name = "synopsys-centos";
    tag = "latest";

    fromImage = baseImage;
    config = { ... };
  };
```

Enter `vmTools`

Mostly used for VMs back in the dark ages before Nix was popular

`rpmClosureGenerator` allows us to create a FOD of RPMs from a package mirror so we install exactly the same packages every time

```
missingLibs = with pkgs; with vmTools;
  let
    centos7-9x86_64 = rpmDistros.centos7x86_64;
    rpmsFrom = distro: extraPackages: { ... };
    rpms = [ ... ];
  in import (
    rpmClosureGenerator (rpmsFrom centos7-9x86_64 rpms)
  ) { inherit fetchurl; };
```

Enter `vmTools`

```
pkgs.dockerTools.buildImage {
  runAsRoot = ''
    cat ${yumExtraConf} >> /etc/yum.conf
    yum localinstall ${builtins.concatStringsSep " " missingLibs}
    # https://github.com/CentOS/sig-cloud-instance-images/issues/71

    mkdir -p /etc/ssh/sshd_config.d; cp ${x11Conf} /etc/ssh/sshd_config.d/
  '';
```

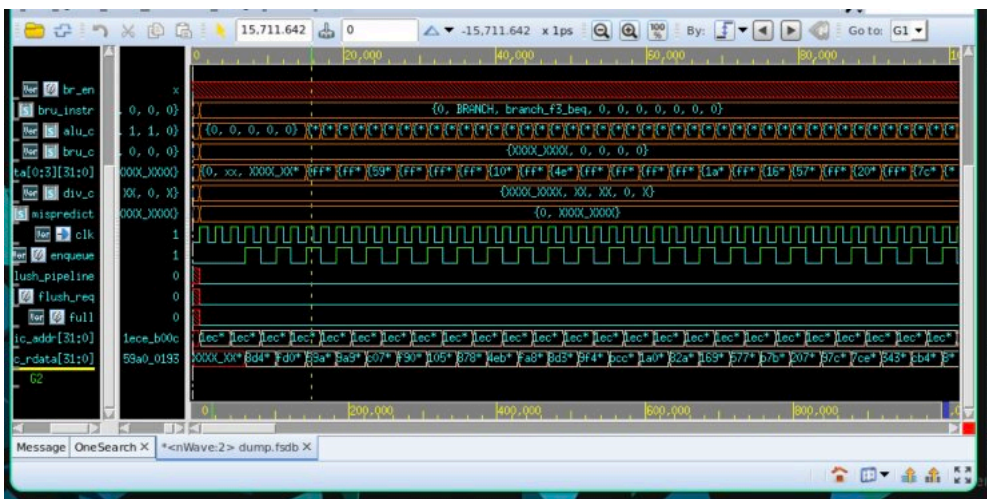
A leaky abstraction

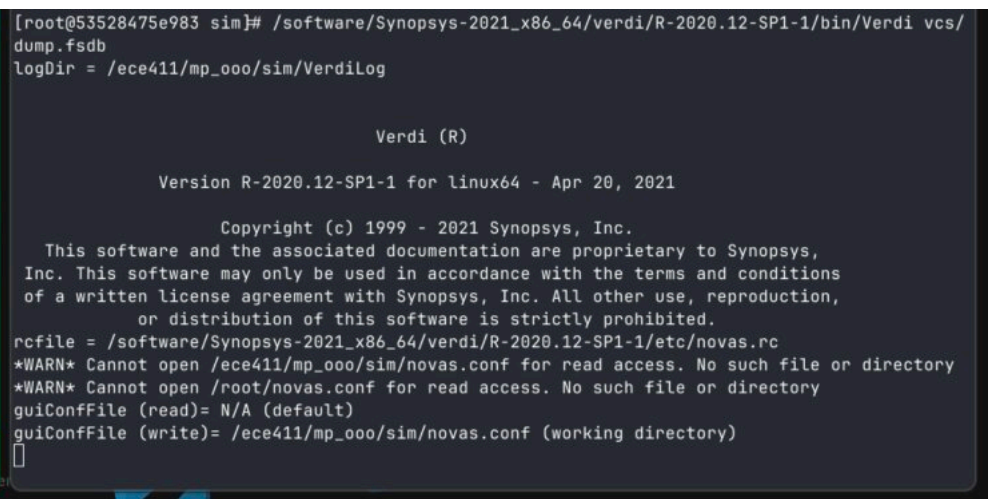
We're using Docker, so our host kernel info bleeds through

Welcome back `deterministic-uname`!

Extra credit: use eBPF to obfuscate `/proc/1/sched`, `/proc/self/mountinfo`, `/proc/1/cgroup`

Free at last


15,711.642
-15,711.642 x 1ps
Go to: G1
br_en
bru_instr
alu_c
bru_c
div_c
mispredict
clk
enqueue
flush_req
full
Message
OneSearch
*<nWave:2> dump.fsdb


[root@53528475e983 sim]# /software/Synopsys-2021_x86_64/verdi/R-2020.12-SP1-1/bin/Verdi vcs/
dump.fsdb
logDir = /ece411/mp_ooo/sim/VerdiLog
Verdi (R)
Version R-2020.12-SP1-1 for linux64 - Apr 20, 2021
Copyright (c) 1999 - 2021 Synopsys, Inc.
This software and the associated documentation are proprietary to Synopsys,
Inc. This software may only be used in accordance with the terms and conditions
of a written license agreement with Synopsys, Inc. All other use, reproduction,
or distribution of this software is strictly prohibited.
rcfile = /software/Synopsys-2021_x86_64/verdi/R-2020.12-SP1-1/etc/novas.rc
WARN Cannot open /ece411/mp_ooo/sim/novas.conf for read access. No such file or directory
WARN Cannot open /root/novas.conf for read access. No such file or directory
guiConfFile (read)= N/A (default)
guiConfFile (write)= /ece411/mp_ooo/sim/novas.conf (working directory)

Using this in practice

```
podman unshare vpnns --name helena -- sleep 24h 0<>/dev/null 1>&0 2>&1 &
openconnect --script-tun --script="vpnns --attach --name helena"

# get vpn nameserver
VPN_DNS_IP=$(grep -o -E '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' "~/.helena/etc/resolv.conf")
# get pid of namespace
VPN_NS_PID=$(lsns -p "$(cat "~/.helena/vpnns.pid")" | grep -m 1 "vpnns --name helena" | awk '{print
$4}')

podman run -itd --rm \
  --network "ns:/proc/$VPN_NS_PID/ns/net" \
  --dns="$VPN_DNS_IP" \
  -v "${XAUTHORITY:-$HOME/.Xauthority}:/root/.Xauthority:rw" \
  -v ~/synopsys-toolchain/Synopsys-2021_x86_64:/software/Synopsys-2021_x86_64:ro \
  -v ~/work/mps:/work
  --security-opt label=type:container_runtime_t \
  synopsys-centos:latest
```

Using this in practice

```
podmanWrapper = bin: writeShellApplication {
  name = bin;
  runtimeInputs = [ podman ];
  text = ''
    podman exec -i helena-chroot "${bin}" "$argv"
  '';
...
packages = (map podmanWrapper [
  "verdi"
  "vcs"
  "fsdb2vcd"
  "dc_shell" "design_vision"
  "sg_shell"
]);
```

Talk is cheap, show me the code

GitHub gist with relevant code

Link bash to /bin and glibc/lib to /lib and be happy

— NixOS Wiki