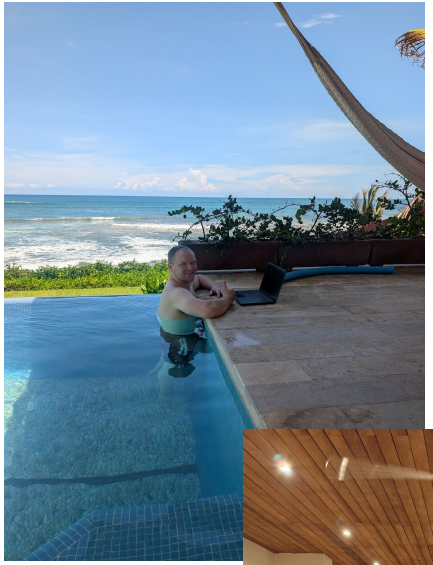
Relocatable Nix Binaries

Farid Zakaria <farid.m.zakaria@gmail.com>

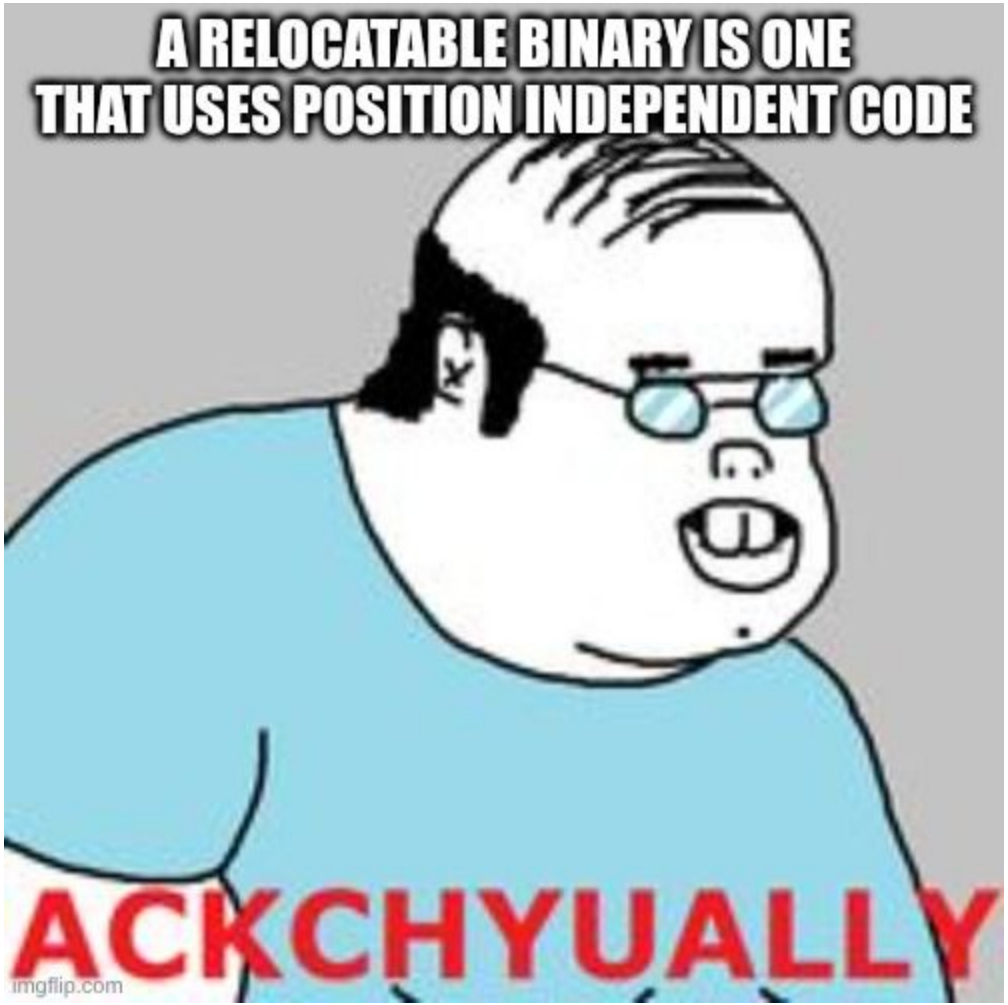
```
> whois fzakaria.com
[Querying whois.fzakaria.com]
Name: Farid Zakaria
Blog: https://fzakaria.com/
Age: 0b100110
Location: Santa Cruz, CA
Tags:
 - Open source advocate
 - NixOS enthusiast
 - Passionate about reproducibility
 - Completed PhD at UCSC
```



tacosprint.org

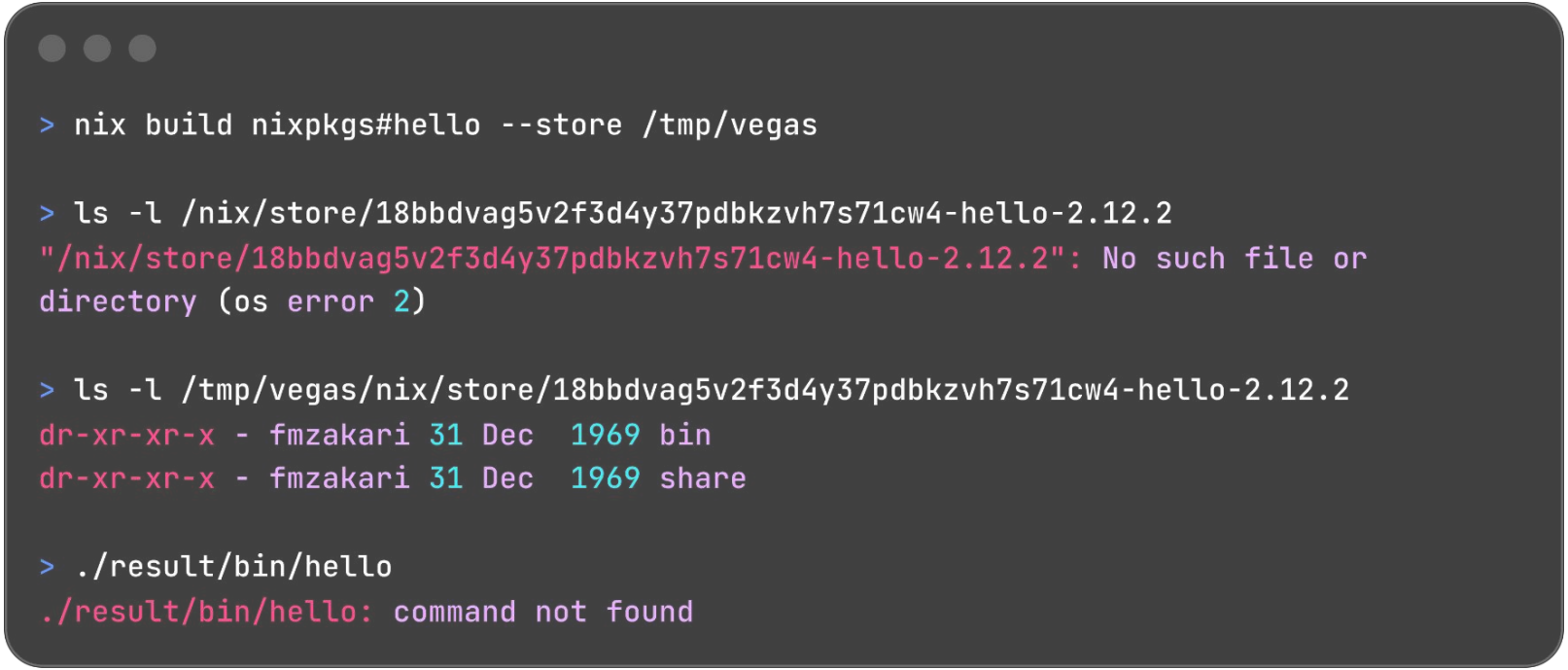
WTF is a Relocatable Binary?

There are 2 hard problems in computer science: cache invalidation, naming things, and off-by-1 errors.
-- *Leon Bambrick*

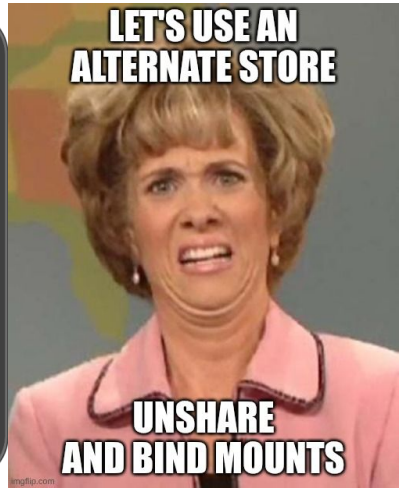

A RELOCATABLE BINARY IS ONE
THAT USES POSITION INDEPENDENT CODE
ACKCHYUALLY
imgflip.com


> nix build nixpkgs#hello --print-out-paths
/nix/store/18bbdvag5v2f3d4y37pdbkzvh7s71cw4-hello-2.12.2


nix-store --query --tree $(nix build nixpkgs#hello --print-out-paths)
/nix/store/18bbdvag5v2f3d4y37pdbkzvh7s71cw4-hello-2.12.2
├──/nix/store/7nbi22pcc92y2fqbkyp7h3srvvklmckb-glibc-2.40-224
│ ├──/nix/store/fv5lgysa3hmf3l3dkkpwvndcg6xwhy8m-xgcc-14.3.0-libgcc
│ ├──/nix/store/qywg7bxskvihq62ms2g51fkzkrdnyfkh-libidn2-2.3.8
│ │ ├──/nix/store/hjwppd89fk8781xl4r35xqlddwqi5f66-libunistring-1.4.1
│ │ │ └──/nix/store/hjwppd89fk8781xl4r35xqlddwqi5f66-libunistring-1.4.1 [...]
│ │ └──/nix/store/qywg7bxskvihq62ms2g51fkzkrdnyfkh-libidn2-2.3.8 [...]
│ └──/nix/store/7nbi22pcc92y2fqbkyp7h3srvvklmckb-glibc-2.40-224 [...]
└──/nix/store/18bbdvag5v2f3d4y37pdbkzvh7s71cw4-hello-2.12.2 [...]

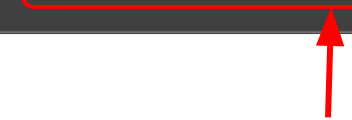

> nix build nixpkgs#hello --store /tmp/vegas
> ls -l /nix/store/18bbdvag5v2f3d4y37pdbkzvh7s71cw4-hello-2.12.2
"/nix/store/18bbdvag5v2f3d4y37pdbkzvh7s71cw4-hello-2.12.2": No such file or directory (os error 2)
> ls -l /tmp/vegas/nix/store/18bbdvag5v2f3d4y37pdbkzvh7s71cw4-hello-2.12.2
dr-xr-xr-x - fmzakari 31 Dec 1969 bin
dr-xr-xr-x - fmzakari 31 Dec 1969 share
> ./result/bin/hello
./result/bin/hello: command not found

```
> unshare --user --mount --map-root-user

> mount --bind /tmp/vegas/nix/store /nix/store

>  ./result/bin/hello
Hello, world!
```




THEY DOES NOT KNOW WHAT WE
HAS IN OUR HOME DIRECTORY, PRECIOUS.
WE WANTS A NIX
STORE. WE WANTS IT.
imgflip.com

```
> XDG_CACHE_HOME=/tmp/vegas/cache \
    nix eval --store 'local?
store=/tmp/vegas/store&state=/tmp/vegasstate&log=/tmp/fzakaria/log' \
    --raw nixpkgs#hello.outPath

/tmp/vegas/store/bwqmy19c01llnqwvphihifxm8nrvy408-hello-2.12.2
```



NOT THE SAME ANYMORE

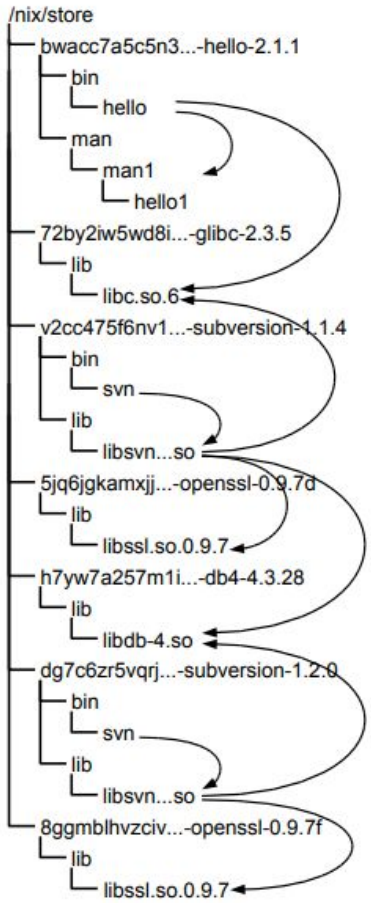


Figure 2.1.: The Nix store

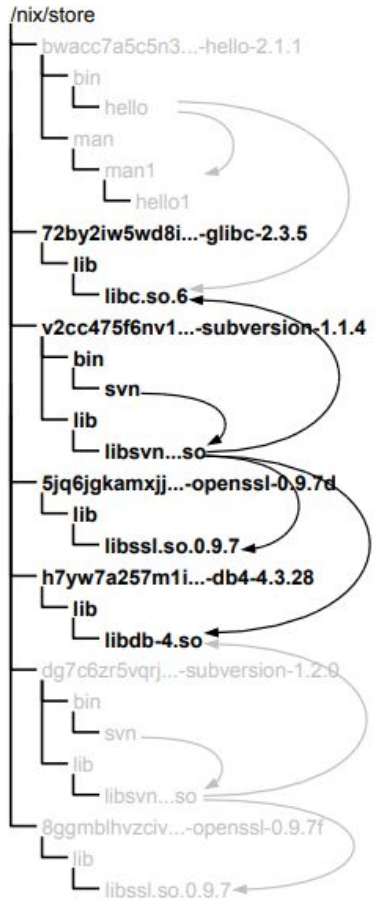
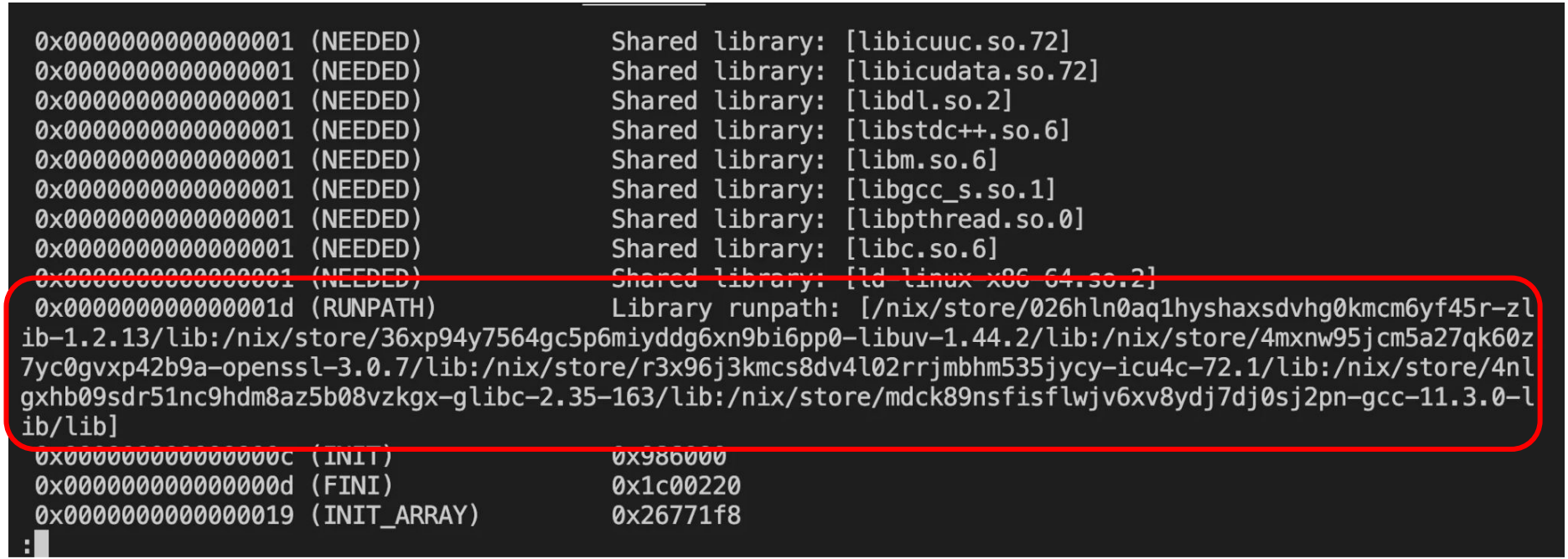


Figure 2.2.: Closure in the store


$ORIGIN
LIGHTYEAR
$ORIGIN EVERYWHERE
imgflip.com



https://maulana.id/soft-dev/2022--11--20--00--patching-elf-header-with-nix/#checking-the-existence-of-rpath

$ORIGIN (or equivalently *${ORIGIN}*)

This expands to the directory containing the program or shared object. Thus, an application located in *somedir/app* could be compiled with

```
gcc -Wl,-rpath,'$ORIGIN/../lib'
```

so that it finds an associated shared object in *somedir/lib* no matter where *somedir* is located in the directory hierarchy. This facilitates the creation of "turn-key" applications that do not need to be installed into special directories, but can instead be unpacked into any directory and still find their own shared objects.

https://man7.org/linux/man-pages/man8/ld.so.8.html

```
$ patchelf --print-rpath /nix/store/18bbdvag5v2f3d4y37pdbkzvh7s71cw4-hello-2.12.2/bin/hello
$ORIGIN/../../7nbi22pcc92y2fqbkyp7h3srvvklmckb-glibc-2.40-224/lib
```

auto-patchelf & patchelf: add options to generate relative RPATHs ($ORIGIN) #534339

View status | Code

Open fzakaria wants to merge 2 commits into `NixOS:staging` from `fzakaria:auto-patchelf-relative-rpaths`

Conversation 17 | Commits 2 | Checks 29 | Files changed 6 | +242 -2

fzakaria commented on Jun 22 • edited — Member

This PR introduces support for generating relative RPATHs (using `$ORIGIN`) for both **`auto-patchelf`** (via `autoPatchelfHook`) and **`patchelf`** (via its own setup hook during `stdenv`'s `fixupPhase`).

When enabled, absolute Nix store paths in the final RPATH are translated to relative paths starting with `$ORIGIN/`. Paths outside of the discovered Nix store (such as `/run/opengl-driver/lib` or other system paths) are preserved as absolute paths to prevent regressions.

Proposed Changes

1. **`auto-patchelf`**:
   - Added the `--relative-rpaths` flag to the `auto-patchelf` Python script.
   - Exposed it via the `autoPatchelfRelativeRpaths` environment variable in the `autoPatchelfHook` setup hook.
2. **`patchelf` setup hook**:
   - Added the `patchelfRelativeRpaths` boolean option to `setup-hook.sh`. When `patchelfRelativeRpaths = true;` is set on a derivation, the `fixupPhase` automatically converts absolute Nix store RPATH entries to relative ones.

Motivation

This is an important step towards enabling fully relocatable binaries in Nix.

Reviewers
- nixpkgs-branch-check[bot]
- layus
- Scrumplex
- flokli
- xokdvium

Request Copilot review

Still in progress? Convert to draft

Assignees

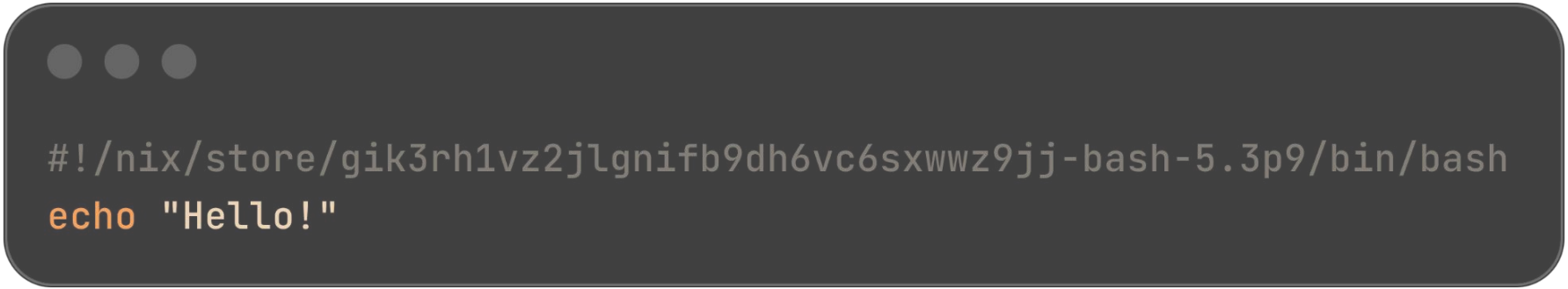
No one—assign yourself

Labels

10.rebuild-darwin: 11-100 | 10.rebuild-linux: 501+ | 10.rebuild-linux: 5001+ | 10.rebuild-linux-stdenv | 10.rebuild-nixos-tests | llm-assisted


COPY THE /NIX/STORE
ANYWHERE AND WATCH MATCH MAGIC
COPIUM
FAILS MISERABLY
DUE TO LD.SO
imgflip.com


$ readelf -a /nix/store/18bbdvag5v2f3d4y37pdbkzvh7s71cw4-hello-2.12.2/bin/hello \|
grep interpreter
[Requesting program interpreter: /nix/store/7nbi22pcc92y2fqbkyp7h3srvvklmckb-glibc-2.40-224/lib/ld-linux-x86-64.so.2]
$ patchelf --print-interpreter \
/nix/store/18bbdvag5v2f3d4y37pdbkzvh7s71cw4-hello-2.12.2/bin/hello
/nix/store/7nbi22pcc92y2fqbkyp7h3srvvklmckb-glibc-2.40-224/lib/ld-linux-x86-64.so.2


#!/nix/store/gik3rh1vz2jlgnifb9dh6vc6sxwwz9jj-bash-5.3p9/bin/bash
echo "Hello!"

linux-fsdevel.vger.kernel.org archive mirror

search help / color / mirror / Atom feed

* [PATCH RFC POC 0/4] binfmt_misc: bpf-backed binary type handlers
@ 2026-07-07 19:36 Christian Brauner
  2026-07-07 19:36 ` [PATCH RFC POC 1/4] exec: stash a bpf-selected interpreter in struct linux_binprm Christian Brauner
                    ` (3 more replies)
  0 siblings, 4 replies; 5+ messages in thread
From: Christian Brauner @ 2026-07-07 19:36 UTC (permalink / raw)
  To: Farid Zakaria
  Cc: Daniel Borkmann, Alexei Starovoitov, Kees Cook, Alexander Viro,
      Jan Kara, Jonathan Corbet, linux-fsdevel, linux-mm, linux-kernel,
      bpf, jannh, linux-mm, mail, Christian Brauner (Amutable)

This is a POC for the nix people and Farid and Eric in particular. I
would take my hands off the wheel now that I POCed this and hand it to
Farid if he likes to take it forward.

VL;MR (very long, must read):

For a while now Farid has been trying to make relocatable, hermetic
binaries (think Nix-style store layouts) work without patchelf tricks
or wrapper scripts. For such binaries the right dynamic loader can only
be determined relative to the location of the binary itself, which
neither PT_INTERP nor a fixed binfmt_misc interpreter string can
express.

The first attempt was $ORIGIN expansion in PT_INTERP [1]. I pushed back
on that. Userspace guards $ORIGIN behind AT_SECURE so the kernel would
have to make the used loader depend on the type of binary, LSMs would
need a say, it changes long-standing behavior in ways that are ripe for
loader injection attacks, and bprm->file may not have a usable path at
all (memfds, deleted files, unresolvable paths). Making the kernel
splice bprm->file back together with PT_INTERP is terrible. The second
attempt was a pluggable ELF interpreter loader registry [2] which would
mean actual kernel modules for custom binary formats. Also no.
binfmt_misc was invented to kill exactly this horrendous past.

https://lore.kernel.org/linux-fsdevel/20260707-work-bpf-binfmt_misc-v1-0-74b995c84ec1@kernel.org/T/

me: "please allow $ORIGIN"

Brauner (maintainer): *"what if it were programmable?"*

```
> bpftool struct_ops register nix_origin.bpf.o /sys/fs/bpf
> echo ':origin:B::::nix:' > /proc/sys/fs/binfmt_misc/register
```

```c
SEC("struct_ops.s/match")
bool BPF_PROG(nix_match, struct linux_binprm *bprm)
{
  return !bpf_strncmp(bprm->buf, 4, "\x7f" "ELF");
}

SEC("struct_ops.s/load")
int BPF_PROG(nix_load, struct linux_binprm *bprm)
{
  char path[256];
  long n = bpf_path_d_path(&bprm->file->f_path, path, sizeof(path));
  if (n < 0)
    return n;
  /* derive the loader location from the binary's own path */
  return bpf_binprm_set_interp(bprm, path, sizeof(path));
}
```

```nix
{ config, lib, pkgs, ... }:
let cfg = config.boot.binfmt.nixOrigin;
in {
  options.boot.binfmt.nixOrigin.enable =
    lib.mkEnableOption "relocatable Nix binaries via BPF interpreter selection";

  config = lib.mkIf cfg.enable {
    systemd.services.nix-origin-binfmt = {
      wantedBy = [ "multi-user.target" ];
      after    = [ "proc-sys-fs-binfmt_misc.mount" ];
      serviceConfig = { Type = "oneshot"; RemainAfterExit = true; };
      script = ''
        ${pkgs.bpftool}/bin/bpftool struct_ops register \
          ${pkgs.nix-origin-bpf}/lib/nix_origin.bpf.o /sys/fs/bpf
        echo ':nix-origin:B::::nix:L' > /proc/sys/fs/binfmt_misc/register
      '';
    };
  };
}
```

Profit?

Farid Zakaria

2026-06-21 · 5 MIN READ

Nix needs relocatable binaries

> This is my problem statement and proposal for a TacoSprint 2026 project 🏄.

Nix, or *store-based systems*, are a class of package managers that use a well-defined prefix to store all packages. This can be `/nix/store` for Nix or `/gnu/store` for Guix.

This is simple. It makes *rewriting* paths to binaries or libraries easy. Derivations only need to `sed` the strings with the full store-path; `/bin/bash` becomes `/nix/store/gik3rh1vz2jlgnifb9dh6vc6sxwwz9jj-bash-5.3p9/bin/bash` for instance.

What if you wanted a different path, one not prefixed at the root `/` ?

This could be desirable if you don't have Nix installed already or are missing necessary permissions – "rootless Nix".

Farid Zakaria

2026-07-20 · 7 MIN READ

Linux kernel will support $ORIGIN, sort of

For some reason, during TacoSprint 2026 I decided to see if we could tackle relocatable binaries in Nix.

I enjoy these lofty goals to push Nix and the surrounding ecosystem forward. I am *bold if not stupid.*

I left the last earlier post with one potential idea of how to get there:

> We could patch the Linux kernel so that $ORIGIN is supported in PT_INTERP and the shebang.

https://fzakaria.com/2026/06/21/nix-needs-relocatable-binaries