

NIX

THE PURELY FUNCTIONAL PACKAGE MANAGER



MARIUSZ GLIWIŃSKI <SHD@NAWIA.NET>

TWO APPROACHES

$[\] \text{ -- } [\] + A \text{ --> } [A] \text{ -- } [A] + B \text{ --> } [A, B] \text{ -- } [A, B] - B \text{ --> } [A]$

OR

$[\] \text{ -- } [\] + A \text{ --> } [A] \text{ -- } [\] + A + B \text{ --> } [A, B] \text{ -- } [A, B] - B \text{ --> } [A]$

OCCAM'S ADVICE: WHY DELETE?

TWO APPROACHES

YEAH, RIGHT... CLUTTER? DISK SPACE? BUILD TIMES?

`[] [] -- [] + A --> [A][A] -- [A] [] + B --> [A,B] [] -- [A,B] [] + A --> [A,B][A]`

DISK SPACE?

`GC([A,B][A]) --> [A][A]`

ATOMIC

[] --[]+A--> [A] --[A]+B--> [A,B] --[A,B]-B--> [A]

is it? ideally yes...

[] --[]+A--> [A] --[A]+B(problem)--> where am i?

what now?

[] [] --[]+A--> [A][A] --[A] []+B--> [A,B] [] --[A,B] []+A--> [A,B][A]

How?

Ok, just dwelve into!

INSTALLATION

SUPPORTED PLATFORMS

- Hardware

arm i686 mips x86_64

- Software

cygwin darwin illumos(solaris) linux netbsd
openbsd freebsd

INSTALLATION

SINGLE-USER

```
#!/bin/bash
curl https://nixos.org/nix/install | sh
```

MULTI-USER

```
#!/bin/bash
curl https://nixos.org/nix/install | sh
groupadd -r nixbld
for n in $(seq 1 10); do useradd -c "Nix build user $n" \
    -d /var/empty -g nixbld -G nixbld -M -N -r \
    -s "$(which nologin)" nixbld$n;
done
nix-daemon
```

PROFILE

SYSTEM PROFILE

```
$ echo $PATH | tr : '\n'
/run/current-system/sw/bin
/run/current-system/sw/sbin
/run/current-system/sw/lib/kde4/libexec
$ ls -l /run/current-system/sw
/run/current-system/sw -> /nix/store/dkba63p5z5af8qiq9kgjaml6nswiqp2p-system-path
$ tree -L 1 /nix/store/dkba63p5z5af8qiq9kgjaml6nswiqp2p-system-path
/run/current-system/sw/
├── bin
├── etc
├── info
├── lib
├── sbin
└── share
$ ls -l /nix/store/dkba63p5z5af8qiq9kgjaml6nswiqp2p-system-path/bin
 '[' -> '/nix/store/95wmyhk0y60r09b98l6v4vswv5m7xm4q-coreutils-8.25/bin/['
2to3 -> /nix/store/awbf3jh262m1k16hxgbgngdmtxv66gd79-python-2.7.11/bin/2to3
7z -> /nix/store/ynvzkiqy3j17aln8rwpv2bjw64l08yci-p7zip-9.38/bin/7z
7za -> /nix/store/ynvzkiqy3j17aln8rwpv2bjw64l08yci-p7zip-9.38/bin/7za
7zr -> /nix/store/ynvzkiqy3j17aln8rwpv2bjw64l08yci-p7zip-9.38/bin/7zr
```

We just gained atomic deployments by updating a symlink.

PROFILE

USER PROFILE

```
$ echo $PATH | tr : '\n'
/home/shd/.nix-profile/bin
/home/shd/.nix-profile/sbin
/home/shd/.nix-profile/lib/kde4/libexec
$ ls -l /home/shd/.nix-profile
/home/shd/.nix-profile -> /nix/var/nix/profiles/default
$ ls -l /nix/var/nix/profiles/default
/nix/var/nix/profiles/default -> default-71-link
$ ls -l /nix/var/nix/profiles/default-71-link
/nix/var/nix/profiles/default-71-link -> /nix/store/54filhqqcmzhdg0a1b5ic7pbv70a19p5
$ tree -L 1 /nix/store/k16kc171xwbanfc72yn9pamygzada7h6-user-environment
├── bin
├── include
├── lib
├── libexec -> /nix/store/smjxvqdr0ndzkrbyxq157cl6z3267wf6-strategoxt-1.8pre24429/li
├── manifest.nix -> /nix/store/q648pxahaadfpmcbasy97sv7czz8-env-manifest.nix
└── share
$ tree -L 1 /nix/store/k16kc171xwbanfc72yn9pamygzada7h6-user-environment/bin
├── 2to3 -> /nix/store/d945ibfx9x185xf04b890y4f9g3cbb63-python-2.7.11/bin/2to3
└── abox2html -> /nix/store/smjxvqdr0ndzkrbyxq157cl6z3267wf6-strategoxt-1.8pre24429/
```

Different users can use different sets of packages (in different versions).
But what about libraries?

ISOLATION

SOURCE PACKAGES

```
gcc -Wl,-dynamic-linker,$(cat $NIX_CC/nix-support/dynamic-linker) \  
-Wl,-rpath,$out/lib
```

ISOLATION

BINARY PACKAGES

```
patchelf --set-interpreter \  
    $(cat $NIX_CC/nix-support/dynamic-linker) \  
    $f  
patchelf --set-rpath $out/lib $f
```

ISOLATION

RESULT

```
$ ldd `which vim`  
    linux-vdso.so.1 (0x00007fff9e7df000)  
    libm.so.6 => /nix/store/pv9sza1cf2kpawck7wbwdnhlip5h57lg-glibc-2.23/lib/libm.so.6  
    libncursesw.so.5 => /nix/store/frv0qy52hk6vv1fn6nh7m4grw7qxs-glibc-2.23/lib/libncursesw.so.5  
    libdl.so.2 => /nix/store/pv9sza1cf2kpawck7wbwdnhlip5h57lg-glibc-2.23/lib/libdl.so.2  
    libc.so.6 => /nix/store/pv9sza1cf2kpawck7wbwdnhlip5h57lg-glibc-2.23/lib/libc.so.6  
    /nix/store/pv9sza1cf2kpawck7wbwdnhlip5h57lg-glibc-2.23/lib/libc-2.23/lib/libc.so.6
```

CHANNELS

<https://nixos.org/channels/nixpkgs-unstable/>

- binary-cache-url
- git-revision
- github-link
- manual/
- nixexprs.tar.xz
- nixexprs.tar.xz.sha256
- src-url
- store-paths.xz

NETWORK CACHE (CHANNELS)

```
$ tree ~/.nix-defexpr/  
/root/.nix-defexpr/  
└─ channels -> /nix/var/nix/profiles/per-user/root/channels
```

```
$ ls /nix/var/nix/profiles/per-user/root/  
channels  channels-20-link  channels-21-link
```

```
$ tree /nix/var/nix/profiles/per-user/root/channels  
/nix/var/nix/profiles/per-user/root/channels  
├─ binary-caches -> /nix/store/m1nblhmhn4zgl4ydv1dxc4xfgnpq4s52-nix  
├─ manifest.nix -> /nix/store/5cbwmyl60sa4sxq5k7f67hkqh70yd4yl-env-  
└─ nixos -> /nix/store/m1nblhmhn4zgl4ydv1dxc4xfgnpq4s52-nixos-16.09
```

```
$ cat /nix/store/5cbwmyl60sa4sxq5k7f67hkqh70yd4yl-env-manifest.nix  
[  
  { meta = { };  
    name = "nixos-16.09pre82105.7465bcd";  
    out = { outPath = "/nix/store/m1nblhmhn4zgl4ydv1dxc4xfgnpq4s52-nix";  
      outPath = "/nix/store/m1nblhmhn4zgl4ydv1dxc4xfgnpq4s52-nix";  
      outputs = [ "out" ];  
      system = "x86_64-linux";  
      type = "derivation";  
    }  
]  
]
```

NIX LANGUAGE

```
nix-repl> 1+1  
2
```

Lazy, pure functional language suited for configuration

NIX LANGUAGE

```
nix-repl> person = {name="john"; surname="kowalski";}

nix-repl> person
{ name = "john"; surname = "kowalski"; }

nix-repl> person = person // {surname="smith";}

nix-repl> person.surname
"smith"
```

- {} defines a set
- // is an overriding operator

NIX LANGUAGE

```
nix-repl> mul = a: b: a*b

nix-repl> with import <nixpkgs/lib>; map (mul 2) (range 1 3)
[ 2 4 6 ]

nix-repl> people = [ {name="Anna"; kids=3;}
                    {name="Jonathan"; kids=2;} ]

nix-repl> with import <nixpkgs/lib>; let value = 500; in
    (foldl
     (x: y: {kids=x.kids+y.kids;})
     {kids=0;} people).kids * value
2500
```

- `:` follows every function parameter
- `import` includes expressions from different package
- functions are curried by default
- `[]` denotes a list definition
- `let ...; in` allows to define a variable within a given scope

NIX LANGUAGE

```
$ ls /nix/var/nix/profiles/per-user/root/channels/nixos/  
COPYING  default.nix  doc  lib  maintainers  nixos  nixpkgs  pkgs  R
```

```
$ ls /nix/var/nix/profiles/per-user/root/channels/nixos/lib  
attrsets.nix          customisation.nix  default.nix        licens  
composable-derivation.nix  debug.nix        deprecated.nix     lists.
```

```
$ ls $NIXPKGS/lib/default.nix
```

`import <nixpkgs/lib>` includes expression located under `$NIXPKGS/lib/default.nix`

NIX PACKAGE

- Is a function (actually derivation)
- Written in Nix language (Scheme in case of Guix)
- Abstraction between stateful Makefile and pure nix expression.

NIX PACKAGE

```
nix-repl> builtins.currentSystem  
"x86_64-linux"
```

```
nix-repl> :l <nixpkgs>  
Added 6400 variables.
```

```
nix-repl> :b derivation {  
  name = "example";  
  builder = "${bash}/bin/bash";  
  args = [ (builtins.toFile "builder.sh" "echo 'Hello World!'");  
  system = builtins.currentSystem;  
}
```

these derivations will be built:

```
  /nix/store/nqva1zxm8rpf0ymql0l75scyzga30rpb-example.drv  
building path(s) \nix/store/4d65rhhfbv9rvz6rgxpamb6h44a9h3b7-example  
warning: you did not specify '--add-root'; the result might be removed  
/nix/store/4d65rhhfbv9rvz6rgxpamb6h44a9h3b7-example
```

this derivation produced the following outputs:

```
$ cat /nix/store/4d65rhhfbv9rvz6rgxpamb6h44a9h3b7-example  
Hello World!
```

NIX PACKAGE

```
with import <nixpkgs> {};  
  
stdenv.mkDerivation {  
  name = "hello-2.1.1";  
  builder = ./builder.sh;  
  src = fetchurl {  
    url = ftp://ftp.nluug.nl/pub/gnu/hello/hello-2.1.1.tar.gz;  
    md5 = "70c9ccf9fac07f762c24f2df2290784d";  
  };  
  inherit perl;  
}
```

```
source $stdenv/setup  
  
PATH=$perl/bin:$PATH  
  
tar xvfz $src  
cd hello-*  
./configure --prefix=$out  
make  
make install
```

```
$ nix-build
```

NIX PACKAGE

```
{ stdenv, fetchFromGitHub, curl, boost, jsoncpp, liboauth, rhash, tinyxml, htmlcxx, help2man }:  
  
stdenv.mkDerivation rec {  
  name = "lgogdownloader-${version}";  
  version = "2.26";  
  
  src = fetchFromGitHub {  
    owner = "Sude-";  
    repo = "lgogdownloader";  
    rev = "v${version}";  
    sha256 = "0277g70nvq7bh42gnry7lz7wqhw8wl2hq6sfxwhn8x4ybkalj2gx";  
  };  
  
  buildInputs = [ curl boost jsoncpp liboauth rhash tinyxml htmlcxx help2man ];  
  
  makeFlags = [ "release" "PREFIX=$(out)" ];
```

- Replace import of all packages with selective import
- Specify buildInputs to differentiate between build/runtime dependencies

HACKABLE

```
$ git clone git@github.com:NixOS/nixpkgs.git  
$ export NIXPKGS=`pwd`/nixpkgs
```

```
lgogdownloader = callPackage ../games/lgogdownloader { };  
boost = self.boost160;  
boost160 = callPackage ../development/libraries/boost/1.60.nix { };
```

```
lgogdownloader = callPackage ../games/lgogdownloader {  
    boost = self.boost159  
};  
boost159 = callPackage ../development/libraries/boost/1.59.nix { };
```

You can override your package directly in the source tree

HACKABLE

```
chromium = callPackage ../applications/networking/browsers/chromium
  channel = "stable";
pulseSupport = config.pulseaudio or true;
enablePepperFlash = config.chromium.enablePepperFlash or false;
enableWideVine = config.chromium.enableWideVine or false;
hiDPISupport = config.chromium.hiDPISupport or false;
};
```

```
packageOverrides = pkgs: with pkgs; rec {
  chromium = callPackage ../applications/networking/browsers/c
    channel = "stable";
    pulseSupport = config.pulseaudio or true;
    enablePepperFlash = true;
    enableWideVine = false;
    hiDPISupport = true;
};
};
```

Or in your `~/.nixpkgs/config.nix`

As long as package and its parameters will be in remote cache you wont need a recompilation.

PACKAGE BUILDING PROCESS

- Receive human-readable package description(.nix)
- Convert that to machine-readable form (.drv)
- Build out-result

QUICK LOOK ON .DRV

```
$ pp-aterm -i /nix/store/l7mimhvayraxccihaw9ihmddq7dhll0n-vim_configurable-7
Derive(
  [("out", "/nix/store/znd8bg8msphy5ly69lz4zffzzwcpq9qj-vim_configurable-7.4
", "")]
, [ ("/nix/store/0z6lnifsapzkaz3y0ni3hyj96bqxgsx8-vim-nox-hg-2082fc3.tar.bz2
"out")]
, ("/nix/store/m87wishf15ya7n60j68fj55zrmf8qnlq-glib-2.46.2.drv", ["out"])
]
, ["/nix/store/9krlzvny65gdc8s7kpb6lkx8cd02c25b-default-builder.sh"]
, ("configureFlags", "--enable-gui=auto --with-features=huge --disable-acl
, ("nativeBuildInputs", "/nix/store/zjla10glbmd7ih6ir1k38br0imv282q3-ncurs
```

Unique /nix/store hash is computed before binary is built (parallel builds).

Trying to reach the point where everything that contributes to package output is a part of hash.

Obviously including other packages depended on.

Build-dependend things such as timestamps are tried to be cleared.

Packages are tested in chroot with only their dependencies included.

TESTING PACKAGE WITH NOX

```
$ cd $NIXPKGS && nox-review wip
$ vim pkgs/tools/networking/ntopng/default.nix
buildInputs = [ libpcap libxml2 glib geoip geolite-legacy
                sqlite which autoreconfHook git pkgconfig groff curl json_c
                rrdtool
];
buildInputs = [ libxml2 glib geoip geolite-legacy
                sqlite which autoreconfHook git pkgconfig groff curl json_c
                rrdtool
];
$ nix-shell -I nixpkgs=$NIXPKGS --pure -p ntopng
$ echo $PATH
```

TESTING PACKAGE WITH NOX

```
$ vim pkgs/development/libraries/glibc/builder.sh
source $stdenv/setup
touch $out/hello
$ cd $NIXPKGS && nox-review wip
```

TESTING NIXOS PACKAGES

```
$ nix-build '<nixpkgs/nixos/tests/login.nix>'
machine# [ 15.286364] reboot: machine restart
vde_switch: EOF data port: Interrupted system call
collecting coverage data
syncing
6 out of 6 tests succeeded
vde_switch: EOF on stdin, cleaning up and exiting
cleaning up
/nix/store/pak4h9irxfa30rw9x9zb7lzfdd7ymr6v-vm-test-run-login
```

TESTING NIXOS PACKAGES

```
$ nixos-rebuild build-vm
$ nix-build -A iso_graphical.x86_64-linux $NIXPKGS/nixos/release.nix
$ nix-build -A netboot.x86_64-linux $NIXPKGS/nixos/release.nix
$ nix-build -A ova.x86_64-linux $NIXPKGS/nixos/release.nix
$ NIXOS_CONFIG=$NIXPKGS/nixos/modules/virtualisation/nova-image.nix
```

UNIX-FRIENDLY

```
$ nix-store -q --references $(which bash)
/nix/store/9xsjqbsldr86vvs7p5ps4qlkqv9dhc0k-glibc-2.23
/nix/store/hbgd7mbpylfz6zxj5bkdyi07yag77b8l-bash-4.3-p42
/nix/store/rgvq65vi785dhqkhh5w4frs2d365589k-ncurses-5.9
/nix/store/jzlfj459383d93gf388ar3skhybwmi44-readline-6.3p08
/nix/store/s4v4jfw6595iaan5gz4dc1z69q20mlhd-bash-4.3-p42
```

```
$ nix-store -q --referrers $(which bash)
/nix/store/s4v4jfw6595iaan5gz4dc1z69q20mlhd-bash-4.3-p42
/nix/store/2j3w04nyyfqilss9j8dmjzvbfq53awrd-system-path
/nix/store/fspj8zz1ki16whsw0ii0400f8p3f82fi-system-path
/nix/store/5cf3yd59wbz07babgsdcb921zwc441kf-etc-shells
/nix/store/c2mm54gyiif7rp9wwv46xma8rnzx2bkh-steam-usr-target
```

```
$ nix-store -q --tree ~/.nix-profile
/nix/store/8b9872z5yrlhdwkrqpsjr208il1d2k6-user-environment
+---/nix/store/bj9bnmfkr52ghcc9zdpzfp239x02j9rp-neovim-0.1.2
|   +---/nix/store/pv9sza1cf2kpawck7wbwdnhlip5h57lg-glibc-2.23
|   |   +---/nix/store/gygp4inn8w53wy161yy08ilf4kvzw0ki-linux-header
|   |   |   +---/nix/store/gygp4inn8w53wy161yy08ilf4kvzw0ki-linux-he
|   |   +---/nix/store/pv9sza1cf2kpawck7wbwdnhlip5h57lg-glibc-2.23 [
|   +---/nix/store/idm1067y9i6m87crjqrbamsdq2ma5r7p-bash-4.3-p42
|   |   +---/nix/store/pv9sza1cf2kpawck7wbwdnhlip5h57lg-glibc-2.23 [
|   |   +---/nix/store/idm1067y9i6m87crjqrbamsdq2ma5r7p-bash-4.3-p42
```

EXTENDING THIS CONCEPT

```
{ config, lib, pkgs, ... }:  
  
with lib;  
  
let  
  cfg = config.services.locate;  
in {  
  options.services.locate = {  
    enable = mkOption {  
      type = types.bool;  
      default = false;  
      description = '  
        If enabled, NixOS will periodically update the database of  
        files used by the locate command.  
      '';  
  };  
};
```

EXTENDING THIS CONCEPT

```
{ nrMachines ? 2 }:  
  
with import ;  
  
let  
  adminAddr = "shd@nawia.net";  
  makeMachine = n: nameValuePair "backend${toString n}"  
    ({ config, pkgs, nodes, ... }:  
      { deployment.targetEnv = "virtualbox";  
        deployment.virtualbox.headless = true;  
        networking.firewall.allowedTCPPorts = [ 80 ];  
        services.httpd.enable = true;  
        services.httpd.adminAddr = adminAddr;  
        services.httpd.documentRoot = "${pkgs.valgrind.doc}/share/doc/va  
      }  
    );
```


EXTENDING THIS CONCEPT

```
{system, pkgs}:

rec {
  rooms = import ../pkgs/databases/rooms {
    inherit (pkgs) stdenv;
  };

  staff = import ../pkgs/databases/staff {
    inherit (pkgs) stdenv;
  };

  zipcodes = import ../pkgs/databases/zipcodes {
    inherit (pkgs) stdenv;
  };

  ZipcodeService = import ../pkgs/webservices/ZipcodeService { 2
```

EXTENDING THIS CONCEPT

```
{stdenv}:\n\nstdenv.mkDerivation {\n  name = "staff";\n  src = ../../../../services/databases/staff;\n  installPhase = '\n    mkdir -p $out/mysql-databases\n    cp *.sql $out/mysql-databases\n  '\n};\n}
```

EXTENDING THIS CONCEPT

```
{stdenv, apacheAnt, jdk, axis2}:

stdenv.mkDerivation {
  name = "ZipcodeService";
  src = ../../../../services/webservices/ZipcodeService;
  buildInputs = [ apacheAnt jdk ];
  AXIS2_LIB = "${axis2}/lib";
  AXIS2_WEBAPP = "${axis2}/webapps/axis2";
  buildPhase = "ant generate.war";
  installPhase = ''
    mkdir -p $out/webapps
    cp *.war $out/webapps
  '';
}
```

THANK YOU



MARIUSZ GLIWIŃSKI <SHD@NAWIA.NET>