Reproducible science and Nix/NixOS

Stephen Huan

https://cgdct.moe

Theory club 2023-10-20

Overview

Introduction

Text reflow

Open science

Open science, open access, open source, etc.

Today: *free* software and *reproducibility*

Linux filesystem

Linux filesystem hierarchy standard

- `/bin`: binaries
- `/boot`: bootloader
- `/dev`: devices
- `/etc`: system configuration
- `/home`: home directories for users
- `/lib`: shared libraries
- `/opt`: application software
- `/root`: root home directory
- `/run`: runtime data
- `/sbin`: system binaries
- `/srv`: services
- `/tmp`: temporary files
- `/usr`: read-only shared data, e.g.
  - `/usr/bin`, `/usr/include`, `/usr/lib`

Linux distributions

Linux is just a kernel

Need *operating system* (e.g. Freedesktop, `systemd`, . . .)

Use a *distribution*, e.g. Debian, Arch, . . .

```
sudo pacman -S firefox
```

Implicitly *imperative* (stateful) model

Declarative (pure) model?

The Nix model

Store everything read-only in the Nix store (`/nix/store`)

When building software, sandbox

Explicitly declare dependencies

Nearly guaranteed reproducibility

NixOS

```
ikue@sora ~> ls -l /
total 12
drwxr-xr-x   2 root root   60 Oct 20 16:19 bin
drwxr-xr-x   4 root root 4096 Dec 31  1969 boot
drwxr-xr-x  20 root root 3700 Oct 20 16:12 dev
drwxr-xr-x  28 root root 1500 Oct 20 18:14 etc
drwxr-xr-x   3 root root   60 Oct 20 16:12 home
drwxr-xr-t   4 root root 4096 Jul  4 17:59 nix
drwxr-xr-x  17 root root 4096 Oct 18 08:10 persistent
dr-xr-xr-x 376 root root    0 Oct 20 16:12 proc
drwx------   4 root root   80 Oct 20 16:19 root
drwxr-xr-x  22 root root  560 Oct 20 16:19 run
drwxr-xr-x   2 root root   40 Oct 20 16:12 srv
dr-xr-xr-x  13 root root    0 Oct 20 16:12 sys
drwxrwxrwt  15 root root  420 Oct 20 18:15 tmp
drwxr-xr-x   9 root root  240 Oct 20 16:12 var
```

Conventional commits

```
<type>[optional scope]: <description>

[optional body]

[optional footer(s)]
```

Header ≈ 50 characters long, description wrapped to ≤ 72

type: type of change, scope: what was changed, e.g.

```
* b3d5e09 feat(experiments/gp_order.py): compare orderings
* 9e30922 feat(misc/ordering.py): add greedy nystrom/kl
* 31dfb09 feat(figures/sensors.py): add maximin ordering
* cb9d8c9 refactor(experiments/gp_regr.py): include kernel
* e65eaf6 fix(KoLesky/sensor.py): delay cholesky import
* d166c01 feat(KoLesky/ordering.py): random sparsity
* 1fe84e7 fix(KoLesky/gp_regression.py): use rng object
* 4fe401c feat(KoLesky/cholesky.py): reference sparsity
* 11ad93e feat(KoLesky/cknn.py): knn subsampling
* 4dedae6 feat(KoLesky/sensor.py): greedy kl ordering
* 5444164 feat(KoLesky/gp_regression.py): rpcholesky points
* a32e0b7 feat(KoLesky/sensor.py): greedy nystrom ordering
* 9466f08 feat(misc/ordering.py): ordering comparison
* 5dda9ee feat(KoLesky/ordering.py): information-theoretic
* b6b29b9 fix(KoLesky/sensor.py): kernel typehint
* 69e688a refactor(figures/sensors.py): list of methods
* c851f54 refactor(KoLesky/sensor.py): style and typehints
* 69c3c7f feat(KoLesky/sensor.py): add sensor implementation
```

Conventional commit types

build: build system/external dependencies

ci: continuous intergration

docs: documentation

feat: new feature (`MINOR` in semver)

fix: bug fix (`PATCH`)

BREAKING CHANGE: in footer or **!** before : (`MAJOR`)

perf: improve performance

refactor: neither fixes a bug nor adds a feature

style: non-semantic changes (formatting, etc.)

test: tests

Language tooling

language	lsp	formatter	linter
sh/bash	bashls	shfmt	shellcheck
css	css-lsp	prettier	v.Nu
html	html-lsp	prettier	v.Nu
json		prettier	jsonlint
julia	julials	JuliaFormatter.jl	Lint.jl
latex	texlab		
lua	luals	selene	stylua
lean	lean4		
markdown		prettier	
nix	nil	nixpkgs-fmt	statix
python	pyright	black/isort	ruff
toml	taplo	taplo	
yaml	yamlls	yamlfmt	yamllint

Neovim plugins

LuaSnip
bclose.vim
cmp-buffer
cmp-cmdline
cmp-nvim-lsp
cmp-path
cmp_luasnip
formatter.nvim
fzf
fzf.vim
gitsigns.nvim
goyo.vim
julia-vim
lean.nvim
leap.nvim
lightline.vim
markdown-preview.nvim
mason.nvim
nvim-autopairs
nvim-cmp
nvim-lint
nvim-lspconfig
nvim-treesitter
packer.nvim
playground
plenary.nvim
polar.nvim
ranger.vim
tabular
tcomment_vim
undotree
vim-cython-syntax
vim-lastplace
vim-matchup
vim-polyglot
vim-repeat
vim-sleuth
vim-snippets
vim-startify
vimtex

Paragraph reflow

```
We the people of the United    |  1
States, in order to form a more|  2
perfect union, establish       |  3
justice, insure domestic       |  4
tranquility, provide for the   |  5
common defense, promote the    |  6
general welfare, and secure the|  7
blessing of liberty to         |  8
ourselves and our posterity, do|  9
ordain and establish the       | 10
Constitution of the United     | 11
States of America.             | 12

We the people of the United    |  1
States, in order to form a     |  2
more perfect union, establish  |  3
justice, insure domestic       |  4
tranquility, provide for the   |  5
common defense, promote the    |  6
general welfare, and secure    |  7
the blessing of liberty to     |  8
ourselves and our posterity,   |  9
do ordain and establish the    | 10
Constitution of the United     | 11
States of America.             | 12
```

Three simple rules

1. Minimize the variance of the lengths of each line. . .
2. . . . subject to the constraint that the number of lines is minimal
3. Ignore the last line, while making sure it's shorter than average

Variance objective

$$\mathbb{V}\mathrm{ar}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2] = \frac{1}{n}\sum_{x \in X}\left(x - \frac{1}{n}\sum_{x \in X} x\right)^2$$

Cumulant to moment conversion

$$\mathbb{V}\mathrm{ar}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$$

Minimize variance $\iff$ sum of squares (for fixed $\mathbb{E}[X]$)

Optimal substructure $\Rightarrow$ dynamic programming

Available for download at
https://cgdct.moe/blog/far/