うわっ...私のコマンド、遅すぎ...？

Rust再実装コマンドでターミナルをもっと便利に

甲本健太

@kentakom1213

Rustとは

@kentakom1213

今回紹介するコマンドたち

- `find` → `fd`
- `grep` → `ripgrep`
- `ls` → `exa`
- `cat` → `bat`

`find` → `fd`（ファイルを探す）

- 高速！
  `README.md`（7569個）を計測
  `find`：8.0 sec
  `fd`：0.75 sec
- みやすい！
  デフォルトで色付けあり
- 賢い！
  デフォルトで`.gitignore`を無視

@kentakom1213

```
❯ find . -name "README.md"
./Animaker/README.md
./nuinfo-seminar/README.md
./nuinfo-seminar/.nuinfo-seminar/lib/python3.10/site-pack
./nuinfo-seminar/.nuinfo-seminar/lib/python3.10/site-pack
./nuinfo-seminar/.nuinfo-seminar/lib/python3.10/site-pack
./old_rust_env/README.md
./Cica/README.md
./zero-one-seminar/deep-learning-from-zero/README.md
./zero-one-seminar/practice-datavisualization/.practice-d
./zero-one-seminar/practice-datavisualization/.practice-d
./zero-one-seminar/practice-datavisualization/.practice-d
./zero-one-seminar/practice-datavisualization/.practice-d
./zero-one-seminar/practice-datavisualization/.practice-d
./zero-one-seminar/practice-datavisualization/.practice-d
./zero-one-seminar/practice-datavisualization/.practice-d
./zero-one-seminar/practice-datavisualization/.practice-d
./zero-one-seminar/practice-datavisualization/README.md
./zero-one-seminar/2022-f-DeepLearning-02/dl-seminar/lib/
./zero-one-seminar/2022-f-DeepLearning-02/README.md
./zero-one-seminar/2022-f-DesignPattern/3.template_method
./zero-one-seminar/2022-f-DesignPattern/README.md
./zero-one-seminar/2022-f-DesignPattern/2.adapter/kotlin/
./zero-one-seminar/zero-one-semi-Marp-theme/README.md
./zero-one-seminar/2022-f-DeepLearning-01/README.md
./dotfiles/README.md
./nuocw-system-2021_backup/README.md
./Python/cpython_source_code/README.md
./practice-datavisualization/.practice-datavisualization/
./practice-datavisualization/.practice-datavisualization/
./practice-datavisualization/.practice-datavisualization/
./practice-datavisualization/.practice-datavisualization/
./practice-datavisualization/.practice-datavisualization/
./practice-datavisualization/.practice-datavisualization/
```

```
❯ fd -HI README.md
nu_totp_autologin-master/README.md
nuocw-system-2021/README.md
docker_clientside_env/README.md
atcoder_problem_parser/README.md
docker_clientside_env/workspace/nuocw-release_backup/READ
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
docker_clientside_env/workspace/nuocw-release_backup/node
```

`grep` → `ripgrep`

- 100倍程度高速！
  1434ファイルの中から `"アルゴリズム"` という文字列を検索
  `grep` ：0.062 sec
  `ripgrep` ：3.8 sec
- みやすい！

@kentakom1213

grep コマンド

```
❯ grep -r "アルゴリズム" .
./abc_training/C/C_IncresingSequense.py:# O(n^2) のアルゴリズム、部分点はもらえる
./Sakumon/Product_of_Range/Product_of_Range_ans.md:ある数列の連続する部分列の総和を高速に処理
./Sakumon/Product_of_Range/Product_of_Range_ans.md:累積和アルゴリズムと同様に、前処理が $O(N
./algorithm/repeated_squeres.ipynb:    "# 高速累乗計算アルゴリズム\n",
./algorithm/repeated_squeres.ipynb:    "O(log(n))のアルゴリズムは強い\n",
./algorithm/LCS.py:### アルゴリズム ###
./algorithm/binary_tree.ipynb:    "- Pythonによるアルゴリズム入門（酒井和哉）"
./algorithm/README.md:# アルゴリズムレパートリー
./algorithm/README.md:競プロでこれまでに勉強したアルゴリズムと、これから挑戦したいもの
./algorithm/README.md:一部アルゴリズムと呼べるかあやしいものもあります。
./algorithm/README.md:参考：[競技プログラミングでの典型アルゴリズムとデータ構造](https://alg
./algorithm/README.md:## アルゴリズム
./algorithm/README.md:### ソートアルゴリズム
./algorithm/README.md:### 探索アルゴリズム
./algorithm/README.md:- [ ] カラツバ法（高速乗算アルゴリズム）
./algorithm/rust_binary_tree/README.md:- [二分木 - Rustではじめるデータ構造とアルゴリズム（第
./algorithm/PartitionFunction.ipynb:    "# 分割数を求めるアルゴリズム\n",
./paiza/jack/in.txt:競プロ経験者の方だけでなく、「やってみたいけど敷居が高い...」とか「アルゴ
./paiza/jack/in.txt:内容は「最近勉強したアルゴリズム」「こだわりのエディタ」「コンテスト前の
./ac-library/document_ja/string.html:<p>文字列アルゴリズム詰め合わせです。
./ac-library/document_ja/string.html:文字列に関する様々なアルゴリズムが入っています。</p>
./ac-library/document_ja/appendix.html:<p>C++初心者には難しいかもしれない機能を表すマークです
が該当します。</p>
./ac-library/document_ja/modint.html:<p>自動でmodを取る構造体です。AC Libraryはmodintを使わな
>
./ac-library/document_ja/math.html:<p>数論的アルゴリズム詰め合わせです。</p>
./notebooks/graph_algorithms.ipynb:    "# アルゴ式 グラフアルゴリズム"
```

ripgrep コマンド

```
❯ rg "アルゴリズム"
notebooks/graph_algorithms.ipynb
7:    "# アルゴ式 グラフアルゴリズム"

algorithm/PartitionFunction.ipynb
74:    "# 分割数を求めるアルゴリズム\n",

algorithm/rust_binary_tree/README.md
4:- [二分木 - Rustではじめるデータ構造とアルゴリズム（ 第1回 ）](https://laysakura.github.io/2

algorithm/README.md
1:# アルゴリズムレパートリー
2:競プロでこれまでに勉強したアルゴリズムと、これから挑戦したいもの
3:一部アルゴリズムと呼べるかあやしいものもあります。
5:参考：[競技プログラミングでの典型アルゴリズムとデータ構造](https://algo-logic.info/competi
8:## アルゴリズム
10:### ソートアルゴリズム
18:### 探索アルゴリズム
67:- [ ] カラツバ法（高速乗算アルゴリズム）

algorithm/LCS.py
11:### アルゴリズム ###

algorithm/binary_tree.ipynb
11:    "- Pythonによるアルゴリズム入門（酒井和哉）"

algorithm/repeated_squeres.ipynb
7:    "# 高速累乗計算アルゴリズム\n",
345:    "O(log(n))のアルゴリズムは強い\n"
```

`ls` ⟶ `exa`

- みやすい！
  デフォルトで色付けあり
- かわいい！
  絵文字を設定できる😄

```
❯ ls -1
Dockerfile.django
LOG.md
README.md
blog
db.sqlite3
django_blog
docker-compose.yml
manage.py
requirements.txt
```

```
❯ exa -1 --icons
 blog
 db.sqlite3
 django_blog
 docker-compose.yml
 Dockerfile.django
 LOG.md
 manage.py
 README.md
 requirements.txt
```

@kentakom1213

`cat` → `bat`