

The Presentation about NixOS by haruki7049

EN:haruki7049

JP:はるき

About me...

だいたいこんなかんじ。ふんいきでよんで

```
#[allow(dead_code)]
pub struct AboutHaruki7049 {
    name: String,
    age: u32,
    hobby: String,
    not_good_at: String,
}

impl AboutHaruki7049 {
    pub fn new() -> AboutHaruki7049 {
        AboutHaruki7049 {
            name: "haruki7049".to_string(),
            age: 17,
            hobby: "Installing the OS".to_string(),
            not_good_at: "Kanji".to_string(),
        }
    }
}
```

haruki7049 : <https://github.com/haruki7049/>

What is the NixOS?

関数型言語である、Nixという言語を用いているパッケージマネージャー（汎用性があるらしくて、パッケージマネージャーなのかも疑問、かつ不明）を使ったOS。しかし、実際にはパッケージや設定を宣言して設定しているために、関数型と言えるのか疑問の余地あり。単純に自分の知識がないだけかもしれへん。

What is the Nix????

こんなコードを求められる言語。

```
let
  pkgs = import <nixpkgs> {};
in
  pkgs.mkShell {
    packages = with pkgs; [
      cargo
      rustc
      rustfmt
      rust-analyzer
    ];
  }
```

ここでmkShellに気づく

シェルを作るとは？

このプロジェクトのディレクトリ構造の確認

```
|-- shell.nix  
|-- Cargo.toml  
|-- Cargo.lock  
|-- src  
    |-- main.rs
```

省略

そう、mkShellとは！！！！

環境を汚さずにプログラミングをする環境を作れる！！

先ほど書いた程度ならば、下記のようにコマンドを打っても良い。

```
nix-shell -p cargo rustc rustfmt rust-analyzer
```

しかし、Nixでのパッケージを作るためには、必ずnix-langを書かなくてはならない。その分野はこのスライドでは省略する。

NixOSでのNixの使い道

NixOSでは、`/etc/nixos/configuration.nix` ファイルを編集して、OS自身の設定をする。近いようなものだと、FreeBSD等の、`/etc/rc.conf` ファイルとかと似たようなものだ。

For example

一部分を記載する。

```
services.xserver = {  
  enable = true;  
  libinput.enable = true;  
  layout = "us";  
  windowManager.i3.enable = true;  
  displayManager.startx.enable = true;  
}
```

Second example

先ほどのコードと同様の意味を持つ、同様の部分のコードを記載。

```
services = {  
  xserver = {  
    enable = true;  
    libinput = {  
      enable = true;  
    };  
    layout = "us";  
    windowManager = {  
      i3 = {  
        enable = true;  
      };  
    };  
    displayManager = {  
      startx = {  
        enable = true;  
      };  
    };  
  };  
};
```

Third example

今度も同様。結構記述の自由度が高いと感じるかもしれない。

```
services.xserver.enable = true;  
services.xserver.libinput.enable = true;  
services.xserver.layout = "us";  
services.xserver.windowManager.i3.enable = true;  
services.xserver.displayManager.startx.enable = true;
```

そんなオプションたち、どこで知るの？

<https://search.nixos.org/options>

に乗ってる。というか検索出来る。検索すべき。

インストール編は省略...

尺的に仕方がない...

Youtubeにて、自分が作った、超スピードでインストールを解説する動画が存在するのでそちらを紹介。人工音声が苦手な人は注意。

<https://youtu.be/Z8Gx1-bW8rl>

一応、公式のドキュメントも紹介。当然ながら全て英語。

<https://nixos.org/manual/nixos/stable/>

おしまい