

# Изграждане на системи върху NixOS

Универсални конфигурации с Nix

Павел Атанасов    Камен Младенов

13.05.2025

# Преговор

- Запознахме се с модулната система
- Разгледахме как се пише конфигурация на система и как я пускаме във виртуална машина

# Раздел 1

## Задачи

# Задачи

- Може (трябва) да използвате функции в `builtins` и `nixpkgs`
- **Не** може да използвате (други) външни библиотеки за пакетиране
- Използвайте интернет, особено <https://search.nixos.org/packages> и <https://search.nixos.org/options>
- Командата за пускане на виртуална машина е:

```
nix-build '<nixpkgs/nixos>' -A vm -I nixpkgs=channel:nixos-24.11  
-I nixos-config=./configuration.nix
```

# Прости конфигурации

*Ако нещо не е дефинирано, изберете подходяща стойност/метод по ваш избор.*

**Задача 1:** Реализирайте конфигурация с потребител `james` и парола `james`.

**Задача 2:** Реализирайте конфигурация с потребители `michael` и `thomas`, и пароли съответстващи на потребителските им имена. Запишете паролата като хеширана стойност в Nix конфигурацията.

**Задача 3:** Реализирайте конфигурация с потребители `rsmith` и `gsmith`. Първия потребител има програмите `gcc`, `vim` и `cmake`, които не са налични при другия потребител. Втория има програмите `python314`, `lynx` и `emacs`, които не са налични при другия потребител. И двата потребителя трябва да имат командите `htop` и `lsblk`.

**Задача 4:** Разширете предходната конфигурация, като добавите `gdm display manager` и `deepin desktop manager`. Може да е нужно да увеличите хардверните ограничения на виртуалната машина!

**Задача 5:** От предходната конфигурация (на задачи 3 и 4), изведете настройките за всеки потребител съответно в `rsmith.nix` и `gsmith.nix` модули, и изведете настройките за графичната среда в трети модул - `gui.nix`.

**Задача 6:** Реализирайте модул `rust-dev.nix`, който добавя програмите `rustc`, `rustup`, `cargo` и `rustfmt` като глобални (налични за всички потребители), ако опцията `rust-dev` е истина.

**Задача 7:** Реализирайте модул `international.nix`, който приема опция `enable` и опция `location`. `location` е списък с една или повече от: "Sofia", "Paris" и "London". Спрямо избраните опции, трябва да се зададат:

- времева зона (спрямо първата стойност)
- езици на клавиатурата
- UTF-8 и `18n.defaultLocale` (спрямо последната стойност)

**Задача 8:** Обновете модулът от бта задача, така че опцията да е спрямо потребител. Тоест, програмите няма да се добавят глобално, а ще се добавят само за поискалия потребител. Потребител поисква, когато зададе `rust-dev.<username>.enable` като истина.



**Задача 9:** Apache е известен уеб-сървър. Неговите настройки могат да се конфигурират чрез `services.httpd`. Реализирайте модул, който дефинира опция `vhosts`, приемаща атрибутно множество (допуснете, че атрибуtnото множество ще е във формата, който `services.httpd.virtualHosts` очаква). Ако `vhosts` не е празно, трябва да:

- добавите `apacheHttpd` в глобалните пакети
- включите уеб сървъра чрез `services.httpd.enable`
- да добавите съответните виртуални хостове
- да добавите ред(ове) в низа `networking.extraHosts`, така че `hostName` на всеки виртуален хост да бъде “пренасочено” към `127.0.0.1` (форматът е като на `hosts` файла)

**Задача 10:** Разширете модулт от предходната задача, така че да добавя `systemd` сървис, който проверява на всеки 10 минути дали всички виртуални хостове работят (дали може да бъдат успешно ping-нати).