

# Filtrující DNS resolver

Autor: Dmitrii Ivanushkin (xivanu00)

Datum: 17.11.2025

# Obsah

<b>1</b>	<b>Problematika</b>	<b>2</b>
<b>2</b>	<b>Návrh aplikace</b>	<b>3</b>
2.1	Architektura programu . . . . .	3
<b>3</b>	<b>Implementace</b>	<b>4</b>
3.1	Zpracování argumentů a souboru domén . . . . .	4
3.2	Porovnávání domén a subdomén . . . . .	4
3.3	Zpracování dotazů . . . . .	4
3.4	Práce s vlákny, sockety a signaly . . . . .	4
3.5	Chybové stavy . . . . .	4
<b>4</b>	<b>Použití</b>	<b>5</b>
4.1	Spuštění programu . . . . .	5
4.2	Spuštění testů . . . . .	5
<b>5</b>	<b>Testování</b>	<b>6</b>
5.1	Popis testů . . . . .	6
5.2	Výsledky testů . . . . .	6
<b>6</b>	<b>Omezení</b>	<b>7</b>
<b>7</b>	<b>Použitá literatura</b>	<b>8</b>

# Kapitola 1

## Problematika

Cílem je vytvoření programu `dns`, který bude filtrovat příchozí dotazy typu A podle předem definovaného seznamu domén a jejich poddomén. Dotazy na blokované domény mají být odmítnuty. Všechny ostatní dotazy jsou přeposílány na určený resolver a získané odpovědi vráceny původnímu tazateli.

# Kapitola 2

## Návrh aplikace

### 2.1 Architektura programu

- **main.c**: Hlavní vstup, čtení parametrů a blokovaných domén
- **helpers.c/h**: Pomocné funkce pro CLI a práci se soubory
- **dns.c/h**: Hlavní logika serveru, sockety, vlákna, signály
- **dns\_utils.c/h**: Parsování DNS dotazů, sestavení odpovědí, kontrola blokovaných domén

# Kapitola 3

## Implementace

### 3.1 Zpracování argumentů a souboru domén

Program při spuštění čte argumenty příkazové řádky (server, port, filtr a volitelně verbose) pomocí funkce `parse_arguments` a následně načte blokované domény ze souboru specifikovaného parametrem `-f` pomocí funkce `read_domains`.

### 3.2 Porovnávání domén a subdomén

Při zpracování dotazu server porovnává doménu z DNS hlavičky s načtenými blokovanými doménami. Pokud je blokována doména `google.com`, jsou automaticky blokovány i všechny její subdomény (např. `ads.google.com`) pomocí funkce `is_blocked`.

### 3.3 Zpracování dotazů

Když server přijme UDP paket, nejprve analyzuje dotaz, aby zjistil cílovou doménu a typ dotazu. Následně ověří, zda doména není blokována. Pokud je doména blokována, server vrátí odpověď obsahující `rcode NXDOMAIN`. Pokud dotaz není typu A, odpoví `REFUSED` znovu nastavením `rcode`. Ve všech ostatních případech dotaz předá odesílá klientovi.

### 3.4 Práce s vlákny, sockety a signaly

Server nejprve vytvoří UDP socket a připojí jej k určenému portu, aby mohl přijímat DNS dotazy. Každý příchozí paket je zpracováván v samostatném vlákně, které získá informace o doméně a typu dotazu a podle konfigurace rozhodne, zda dotaz zablokovat, odmítnout nebo přeposlat resolveru. Pro bezpečné ukončení serveru se využívají signály `SIGINT` a `SIGTERM`, které zajišťují korektní uzavření socketu a uvolnění zdrojů.

### 3.5 Chybové stavy

Výpis chyby na `stderr` a ukončení programu pomocí `exit(EXIT_FAILURE)`

# Kapitola 4

## Použití

### 4.1 Spuštění programu

Kompilace

```
1 make
```

Po komplikaci uživatel může spustit program pomocí příkazu

```
1 dns -s server [-p port] -f filter_file [-v]
```

Popis parametrů:

- **-s**: IP adresa serveru DNS resolveru
- **-p port**: Port pro naslouchání příchozím DNS dotazům
- **-f filter\_file**: Cesta k souboru obsahujícímu blokované domény
- **-v**: Režim s podrobným výpisem

### 4.2 Spuštění testů

```
1 make test
```

# Kapitola 5

## Testování

### 5.1 Popis testů

Testování programu bylo provedeno automatizovaným skriptem, který ověřuje následující scénáře:

1. Spuštění programu bez povinného parametru **-s** selže
2. Spuštění programu bez povinného parametru **-f** selže
3. Dotaz na nepovolenou doménu je úspěšně přeposlán na resolver a vrátí odpověď typu A
4. Dotaz na blokovanou doménu je blokován
5. Dotaz na typ jiný než A je odmítnut

### 5.2 Výsledky testů

Všechny testy proběhly úspěšně, server korektně reagoval na blokované domény, správně přeposílal dotazy typu A a odmítal dotazy jiných typů. Program tedy funguje podle specifikace.

# Kapitola 6

## Omezení

1. Server vyžaduje, aby parametr **-s** byl platná IP adresa. Hostname (např. `google.com`) bohužel není podporován.
2. Spuštění s defaultním portem vyžaduje použití **sudo**

# Kapitola 7

## Použitá literatura

- Linux man pages: getopt(3) Dokumentace funkce `getopt` pro parsování argumentů příkazové řádky.
- Read file into array in C Ukázky načítání souboru do pole řetězců.
- UDP client-server in C (GeeksforGeeks) Příklad implementace UDP klienta a serveru.
- DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION Specifikace protokolu DNS.