Vysoké učení technické v Brně

Fakulta informačních technologií

**ISA**

**Síťové aplikace a správa sítí**

**Programování síťové služby: Whois Tazatel**

Martin Macháček

17.11.2019 xmacha73

# 

[**Zadání**](#_klerc977mbi2) **2**

[**Problematika & Vysvětlení**](#_oufs75ny63ub) **2**

[1. WHOIS](#_2hf9t43l9vfn) 2

[2. DNS](#_eh2a0fu6ggw8) 2

[**Požadovaný výstup**](#_u2bzony3ydly) **2**

[1. WHOIS](#_8lyy6kgl3qwj) 2

[2. DNS](#_yxjli0ptjtex) 3

[**Návod spuštění**](#_9qaxugae6wls) **3**

[Vysvětlivky](#_e5pcd7ssdpaw) 3

[**Implementace**](#_55gj9qultf7p) **4**

[1. Parsování argumentů](#_ikpxezv1kubz) 4

[2. DNS](#_ciwxvttqfr5l) 5

[3. WHOIS](#_var1kf56nytq) 5

[**Testování**](#_niksbkr8fegn) **6**

[**Použité zdroje & Literatura**](#_93je5il60cjg) **13**

# 

# **Zadání**

Úkolem je vytvořit program v programovacím jazyce C nebo C++, který má dvě části. První částí je implementace tzv WHOIS Lookup, který bude o uživatelem zvolené doménové adrese/IP adrese zjišťovat informace z uživatelem uvedeného WHOIS serveru. Druhou částí je implementace tzv. DNS Lookup, který o zadané adrese bude zjišťovat DNS záznamy jako A, AAA, CNAME pomocí výchozího DNS v operačních systému. Program má podporovat IPv6.

# Problematika & Vysvětlení

## 1. WHOIS

Pojem WHOIS značí databázi, sloužící k evidenci informací o internetových doménách, IP adresách a jejich náležících majitelích. Databáze domén na nejvyšším (národním) řádu, je strukturována hierarchicky, kde každý národní registrátor má svůj vlastní WHOIS server. Všechny komunikace pro takovéto WHOIS dotazy běží na portu 43.

Po důkladnějším nastudování informací o této databázi je ale zřejmé, že velkým nedostatkem, někdy i problémem, je, že WHOIS servery nedodržují žádné standardy které by mohly definovat nutnost unifikace formátu pro dotazy na takové servery. Tato služba je volně dostupná a existuje mnoho služeb/způsobu na získání WHOIS informací, ovšem většina je vytvořena v podobě webové aplikace.

## 2. DNS

Domain Name Servers. Systém doménových jmen s informacemi, na kterém serveru se domény nacházejí. Jedná se v podstatě o “překladač” doménových jmen na odpovídající IP adresy a obráceně (dotaz pro reverzní překlad IP adresy). Naštěstí narozdíl od WHOIS je formát informací u absolutní většiny doménových serverů standardizovaný.

# Požadovaný výstup

## 1. WHOIS

Pro WHOIS Lookup jsou výstupem informace o zadané adrese, získané ze zadaného WHOIS serveru. Mezi údaje patří (tyto údaje mohou mít variabilní název, každý whois server má svůj vlastní formát, tahle aplikace neslouží pro vytvoření standardizovaného formátu výstupu, nýbrž pro zjištění důležitých, zajímavých informací, níže uvedený formát je z whois.ripe.net) inetnum, netname, descr, country, address, phone (\*). Nejedná se o všechny hledané údaje, ale jen o ty nejdůležitější. Pokud nějaká informace není nalezena, nebude vypsána.

## 2. DNS

U DNS Lookup části je očekávaný výstup list záznamů získaných z DNS o zadané domén. Patří mezi ně: A, AAA, MX, CNAME, NS, SOA a PTR. (\*). Seznam nenalezených záznamů bude vypsán na konci DNS Lookup.

(\*) vysvětlivky lze nalézt na internetu, např <https://ns1.com/resources/dns-records-explained>

# Návod spuštění

Program je navržen jako aplikace primárně určena pro operační systém Linux. Výsledkem implementace je binární soubor isa-tazatel (pomocí příkazumake). Soubor isa-tazatel pro spuštění příkazem ./isa-tazatel přijímá tyto argumenty/přepínače (pořadí není důležité):

-help

-q <IP|Hostname> (povinné)

-w <IP|Hostname> (povinné)

-d <IP> (nepovinné)

-f (nepovinné)

-o (nepovinné)

## Vysvětlivky

**-help** - nelze kombinovat s ostatními argumenty, vypisuje nápovědu k používání a spouštění  
**-q <IP|Hostname>**   
 - je IP adresa nebo doménové jméno, o kterém chce uživatel získat informace

**-w <IP|Hostname>**

- je IP adresa nebo doménové jméno WHOIS serveru, ze kterého chce získat informace o -q

**-d <IP>** (rozšíření)  
 - IP adresa DNS, pomocí kterého se bude provádět DNS lookup na adresu v -q

- Argument je nepovinný, jako výchozí DNS se vezme výchozí DNS z OS

**-f** (rozšíření)

- Pokud uživatel nechce, aby program jakkoliv modifikoval/upravoval formát přijatých informací, může si přepínačem -f nechat vypsat úplně celou odpověď, kde budou například i komentáře, přebytečné informace, duplicitní záznamy, atd.. Jedná se o vlastní rozšíření.

**-o** (rozšíření)

- Tento program ve WHOIS Lookup zjišťuje informace o všech nalezených IP adresách při zadání doménového jména do -q, tento přepínač bude limitovat program, aby zjistil informace pouze o první nalezené IP adrese. Jedná se o vlastní rozšíření

Přesné chování programu je popsáno v sekci Implementace níže

# Implementace

Program je implementován pomocí jazyka C++. Volba jazyka nebyla složitá. Díky přítomnosti “datového typu” string a možnosti intuitivní práce s vektory odpadá spousta nepříjemností a zbytečné práce, které by implementaci samotného nápadu akorát zdržovali.

Mezi zajímavé části implementace patřilo tvoření vlastních struktur pro získání ip adres v přehledné formě, invertování a modifikace formátu IP adres a také parsování odpovědí u DNS dotazů.

## 1. Parsování argumentů

Program nejprve kontroluje argumenty od uživatele, přesněji počet těchto zadaných argumentů. 2 povinné argumenty s nutnou hodnotou značí, že minimální “počet argumentů” je 4(+1 cesta).

Pokud je tohle pravidlo splněno, program využije funkci getopt, která parsování argumentů velice zjednodušší, protože umí zachytit potřebné hodnoty z jednotlivých argumentů. Pokud chybí povinný argument, či zachytí neznámý argument, program skončí s chybou a vypíše uživateli odpovídající chybovou hlášku.

Přepínače -o a -f v případě přítomnosti nastavují příslušné proměnné typu bool na pozdější fázi. Celé parsování argumentů je implementováno funkcí parse\_args, která vrací string pole. Na pozici 0 je hodnota argumentu -q a na pozici 1 hodnota -w. Program zkontroluje, jestli náhodou IP hodnoty -q a -w jsou ve správném formátu, aby se například nestalo, že projde adresa 70.72.0.

## 2. DNS

Tato fáze programu vyhledá záznamy o adrese -q ve zvoleném DNS přes argument -d, pokud není přítomen, pracuje se s výchozím DNS operačního systému. Pokud je zadaný argument -d neexistující adresa, či nemá správný formát, program taktéž bude pracovat s výchozí DNS operačního systému. Takovéhoto chování je dosáhnuto využitím funkce res\_init(), a modifikací \_res.nsaddr\_list[0].sin\_addr funkcí inet\_pton.  
 Implementace je provedena funkcí get\_dns. Tato funkce nejdříve pomocí funkce getaddrinfo zjistí, jestli doménová adresa je validní, pokud ano, provede kontrolu, jestli se jedná o IPv4 či IPv6 (funkce inet\_ntop z knihovny arpa/inet.h), protože v případě, že je zadaná IP adresa, je nutné provést tzv “reverzní lookup”, při kterém se z DNS získá záznam PTR, s jehož pomocí můžeme následně získat zbytek záznamů. U tohoto způsobu je nutné otočit a modifikovat formát IP adresy do příslušného tvaru (IPv6 jinak než IPv4).

Při IPv4 se nejdříve z řetězce vytvoří vektor pomocí funkce vectorize\_string s rozdělujícím znakem tečky, ve kterém se poté invertuje pořadí prvků a znovu se IP adresa poskládá zpět. Na závěr se přidá řetězec “in-addr.arpa”. U IPv6 je nutné nejdříve roztáhnout adresu na plnou délku, tahle část je zajištěna pomocí funkce sprintf, poté je potřeba tuto adresu invertovat a místo dvojtečky program přidá za každou číslici znak tečku.

Detailní vysvětlení uvedeno zde:   
 <https://www.ripe.net/manage-ips-and-asns/db/support/configuring-reverse-dns>

Záznamy o adrese z DNS program získává použitím query\_info\_dns funkce, která při každém úspěšně získaném záznamu přidá důležité informace do vektoru řetězců (C++ string). V této funkci také dochází k samotnému dotazu na DNS pomocí funkce res\_query z knihovny resolv.h, kde se specifikuje konkrétní záznam (A, AAA, NS..).

U každého záznamu zanalyzuje odpověď, následně nahradí tabulátory mezerama, vymaže přebytečné mezery a rozdělujícím znakem mezery rozseká odpověď pomocí funkce vectorize\_string na prvky vektoru, a podle typu záznamu sestaví konečný výstup a ponechá si pouze informace, které potřebuje.

Po dotázání na všechny záznamy pomocí práce s vektory zjistí, které záznamy chybí, a vrátí je hlavní funkci, která je následně vypíše.

## 3. WHOIS

Po dokončení DNS dotazování program přejde do funkce get\_whois, která musí vyhledat informace o zvolené adrese na zvoleném WHOIS serveru. Tahle funkce začíná samozřejmě, jako všechny funkce, deklarací potřebných struktur a proměnných pro správný chod programu, poté se pomocí inet\_pton zkontroluje, jestli hodnota v -q není IP adresa, pokud není, tak si program zapamatuje, že má posílat dotaz i na konkrétní doménové jméno, a ne jen na nalezené IP adresy.

Po kontrole IP si vytvoří pomocí funkce get\_ip\_structs vlastní struktury, ve kterých jsou uvedeny všechny nalezené IP adresy, s přidanou bool vlastností, determinující verzi IP adresy (4/6). Funkce get\_ip\_structs funguje na základě funkce getaddrinfo a for cyklu, ve kterém pomocí ai\_next uvnitř získané struktury funkcí getaddrinfo.

Jakmile program získá všechny potřebné IP adresy a struktury, vejde do dlouhého cyklu for, ve kterém pro každou nalezenou adresu pro -q provede následující:

**1)** Zkontroluje, jestli se jedná o IPv6, následně vytvoří příslušný socket, naplní posílaný paket  
**2)** Nastaví timeout 4 vteřin pro následující connect funkci

**3)** Využije funkci connect na základě verze IP

**4)** Pošle zprávu a zachytí odpovídající paket skrz funkci send\_and\_collect.

* V této funkci se také zjišťuje bool hodnota, kterou program nastavoval dříve.
* Vyplní se zde zpráva, ve které je adresa, o které chceme zjišťovat informace.
* Zprávu program pošle funkcí send a pomocí cyklu while a funkce recv vytáhne informace z odpovědi od serveru.
* Tyto informace vrací nadřazené funkci.

**5)** Informace z předešlé funkce se nyní analyzují, a:

* Vymažou se řádky, které nejsou potřebné (komentáře, prázdné řádky..).
* Program vytvoří dlouhý seznam záznamů, které má ignorovat (např. OrgAbuse, Parent, RAbuseName..), a aplikuje tento “filtr”.

**6)**  Finální informace se vrací hlavní funkci a jsou poté vypsány uživateli ve vhodné podobě

Uživatel může ovlivnit do určité úrovně výstup programu, díky přepínačům -o a -f, které jsou uvedené a vysvětlené v předešlých kapitolách.

Tento program nijak nemodifikuje zadané adresy, například neodmazává, či nepřidává automaticky ‘www’ v doménovém názvu, nýbrž spoléhá na znalosti uživatele o whois serverech, na které se dotazuje.

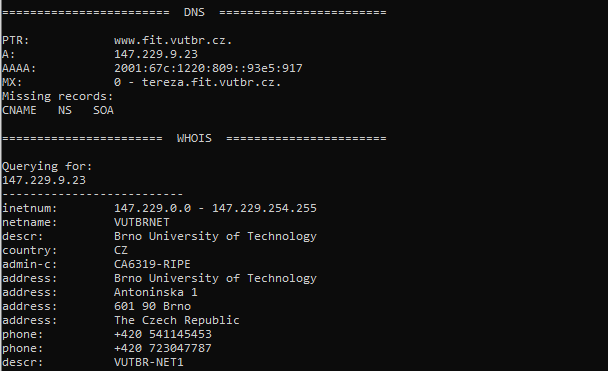
# Testování

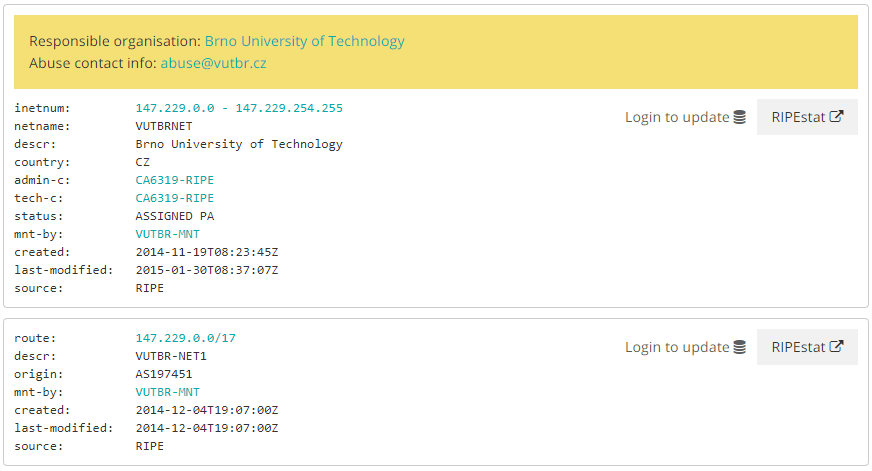
Testoval jsem převážně na online dostupných zdrojích & Web WHOIS & DNS Lookup aplikacích

Níže uvádím několik testovacích subjektů a jejich výsledky, pod nimi/vedle nich je vložen vždy výstup z existujícího online testovacího zdroje (např ripe.net a nic.cz).

**Test 1**

./isa-tazatel -q 147.229.9.23 -w whois.ripe.net



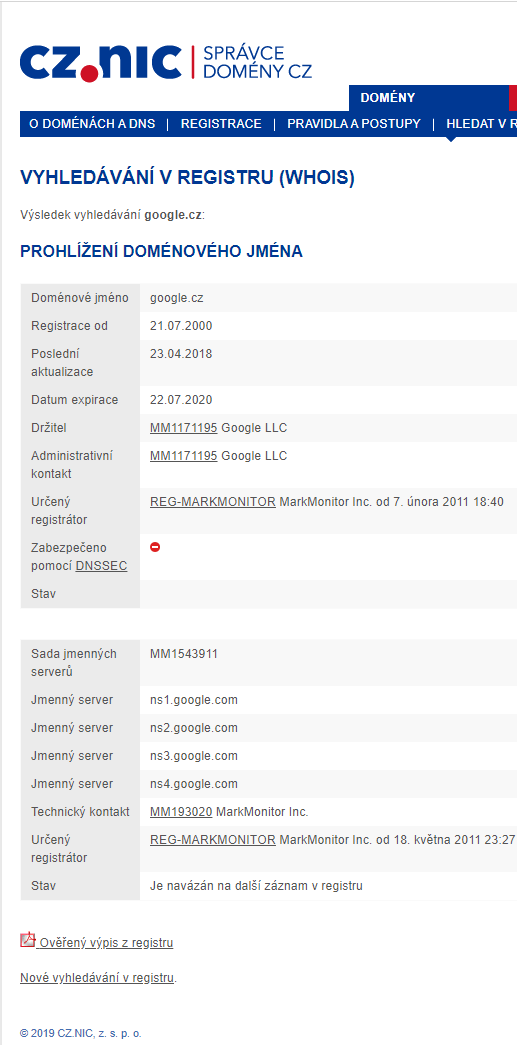
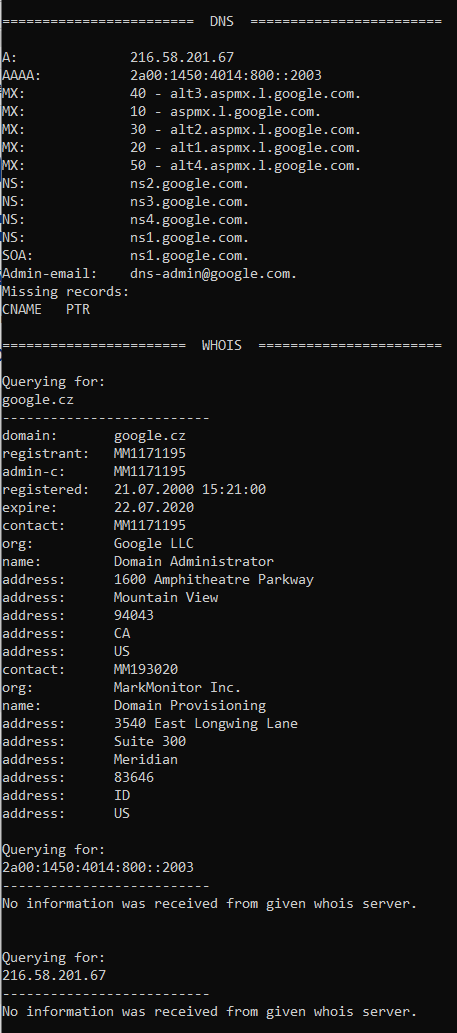
<https://apps.db.ripe.net/db-web-ui/#/query?searchtext=147.229.9.23>

**Test 2**

**./isa-tazatel -q google.cz -w whois.nic.cz**

<https://nic.cz/whois/domain/google.cz>

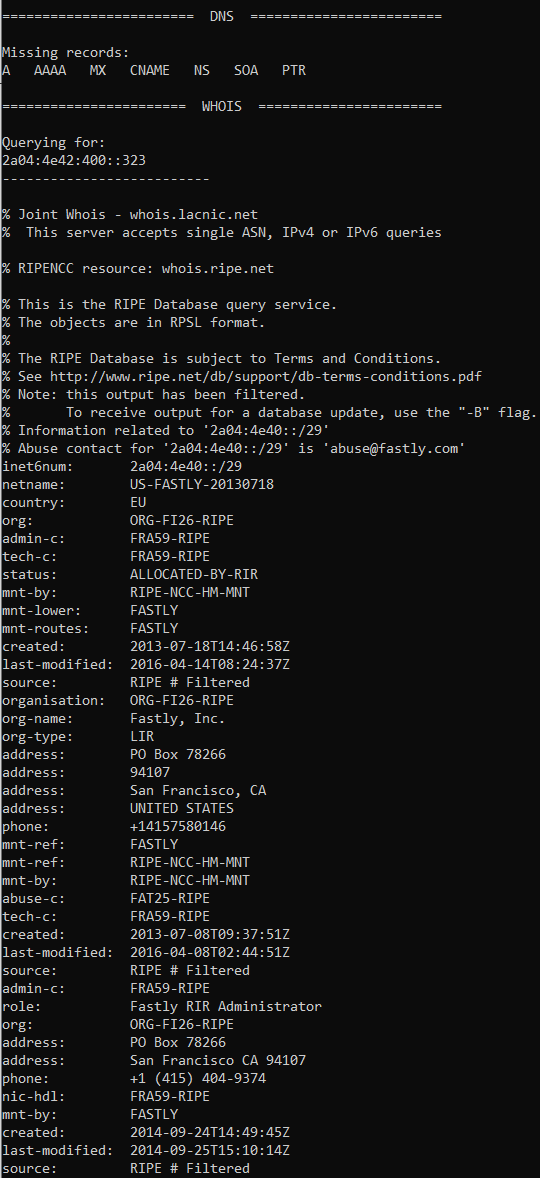
Whois.nic.cz je patří mezi servery, které odpovídají jen na zprávu s doménovým jménem uvnitř.



**Test 3**

./isa-tazatel -q 2a04:4e42:400::323 -w whois.lacnic.net **-f**

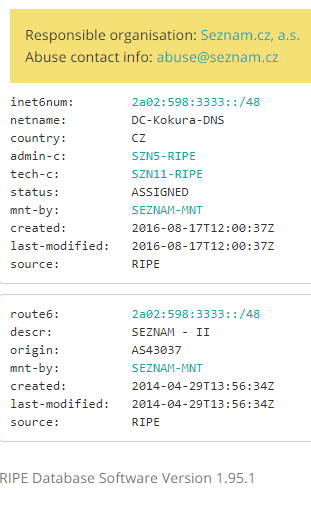
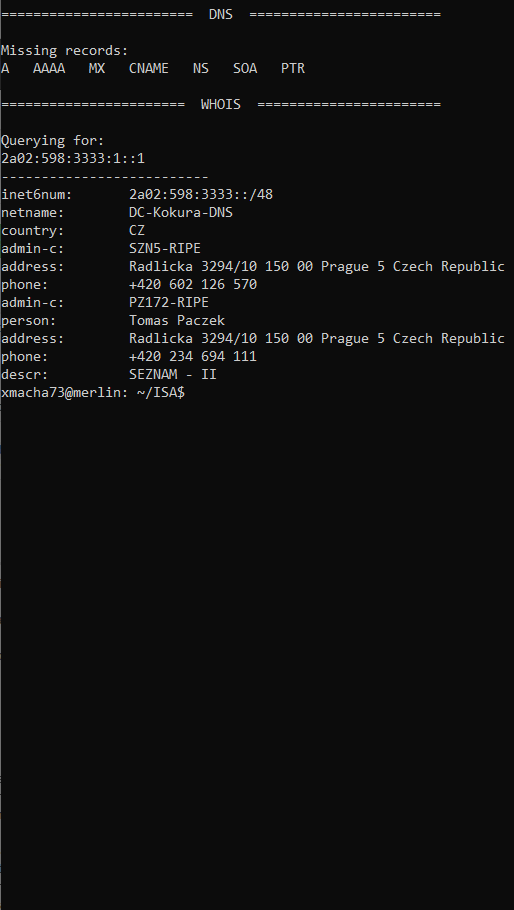
<http://lacnic.net/cgi-bin/lacnic/whois?lg=EN>



**Test** **4**

./isa-tazatel -q 2a02:598:3333:1::1 -w whois.ripe.net

<https://apps.db.ripe.net/db-web-ui/#/query?searchtext=2a02:598:3333:1::1>

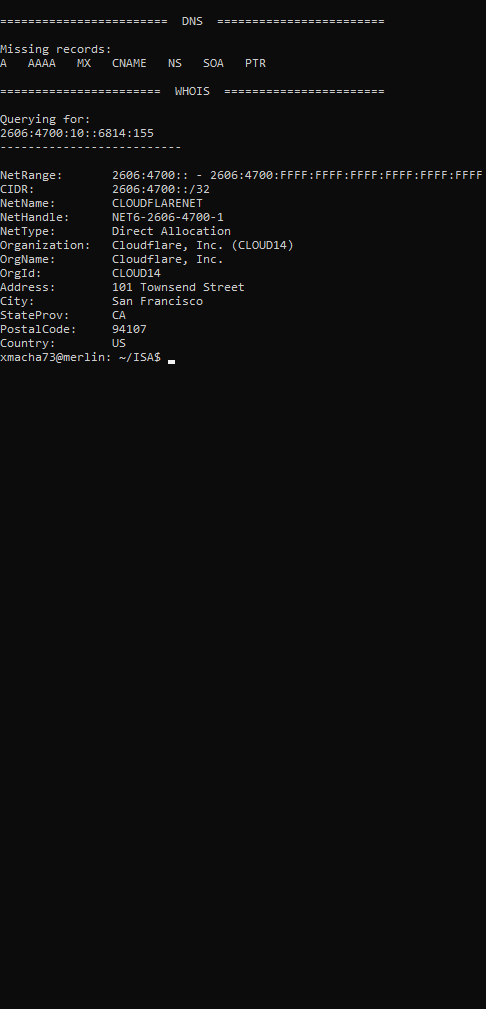
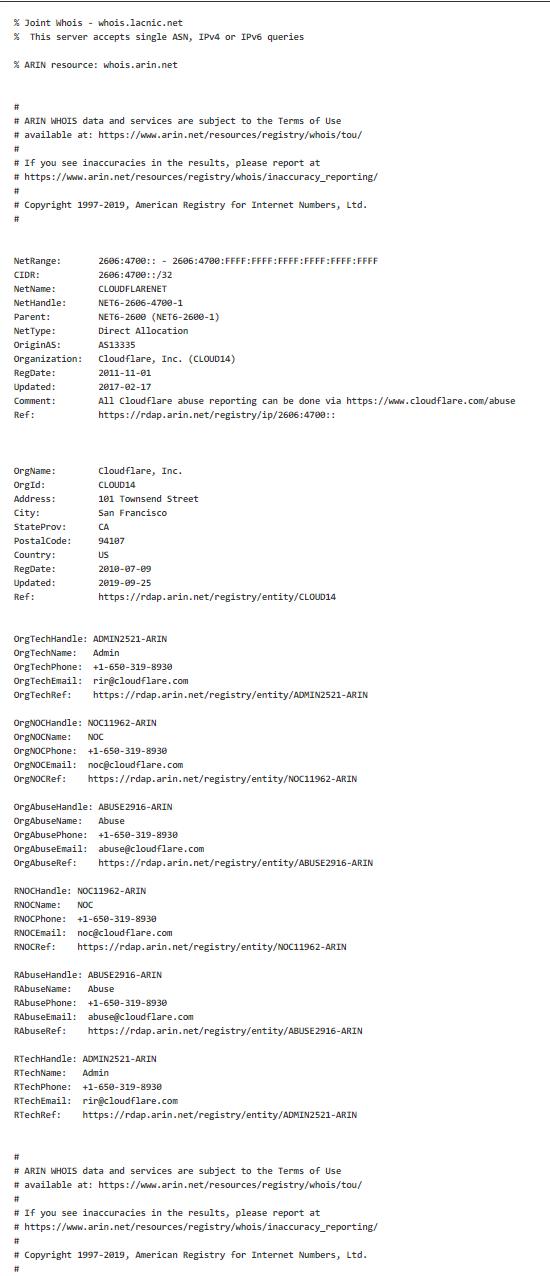


**Test 5**

[**http://lacnic.net/cgi-bin/lacnic/whois?lg=EN**](http://lacnic.net/cgi-bin/lacnic/whois?lg=EN)

**./isa-tazatel -w whois.lacnic.net -q 2606:4700:10::6814:155**

Naprosto stejných výsledků lze dosáhnout použitím -f přepínače



**Test 6**

**(linux): dig skype.com any**

**./isa-tazatel -q skype.com -w whois.iana.org -o -d 8.8.8.8**

Slouží pro ověření DNS výsledků, lze i zadávat argument -d (bonus)

# 

# 

# 

# 

# Použité zdroje & Literatura

[1] RFC 954: NICNAME/WHOIS

<https://tools.ietf.org/html/rfc954>

[2] DNS Explained  
 <https://cs.wikipedia.org/wiki/Domain_Name_System>

[3] RFC 1834: Whois and Network Information Lookup Service, Whois++

<https://tools.ietf.org/html/rfc1834>

[4] Resolv.h Linux manual page

<http://man7.org/linux/man-pages/man3/resolver.3.html>

[5] DNS Lookup - MxToolBox

<https://mxtoolbox.com/DNSLookup.aspx>

[6] Whois - Wikipedie

<https://cs.wikipedia.org/wiki/Whois>

[7] Stackoverflow (v případě zaseknutí nad problémem)

<https://stackoverflow.com/>