

# Projektová dokumentácia Whois tazatel

## 1. Obsah

<b>2.</b>	<b>ÚVOD .....</b>	<b>3</b>
<b>3.</b>	<b>NÁVRH A IMPLEMENTÁCIA.....</b>	<b>3</b>
3.1.	MAIN() .....	3
3.2.	HELP() .....	3
3.3.	ERR_ARGUMENTS() .....	4
3.4.	ERROR_EXIT(INT CODE, STRING STR) .....	4
3.5.	SET_DNS_SERVER(INT *DFLAG, CHAR *DNS_SERVER).....	4
3.6.	RESOLVER(CHAR *DNAME, INT *DFLAG, CHAR *DNS_SERVER) .....	4
3.7.	PRINTF_CNAME(STRING S, INT *FLAG) .....	4
3.8.	PRINT_SOA_ADMIN_EMAIL(STRING S) .....	4
3.9.	PRINT_NS(STRING S).....	4
3.10.	PRINT_MX(STRING S) .....	4
3.11.	PRINT_REGEX(REGEX RX, CHAR *OUTPUT, INT *FLAG ) .....	4
3.12.	PRINT_WHOIS(CHAR *OUTPUT) .....	4
<b>4.</b>	<b>NÁVOD NA POUŽITIE .....</b>	<b>5</b>
4.1.	PRELOŽENIE PROGRAMU .....	5
4.2.	SPUSTENIE PROGRAMU .....	5
<b>5.</b>	<b>TESTOVANIE .....</b>	<b>6</b>
<b>6.</b>	<b>BONUSY .....</b>	<b>6</b>
<b>7.</b>	<b>ZÁVER.....</b>	<b>6</b>
<b>8.</b>	<b>LITERATÚRA .....</b>	<b>6</b>

## 2. Úvod

Cieľom projektu bolo vytvoriť aplikáciu v C/C++. Vytvoril som program, ktorý po internete posiela pakety na Whois servery, odpoveď od týchto serverov následne spracuje a vypíše ju vo formátovanej verzii.

Program na vstupe spracuje vstupné argumenty vďaka ktorým zistí na ktorý server sa má poslať žiadosť o zaslanie informácií o určenej doméne (sub-doméne, IP adrese, ...).

Všetky teoretické informácie som čerpal prednášok a z knihy Ing. Matouška [1]

## 3. Návrh a implementácia

Aplikácia je rozdelená do dvoch zdrojových a jedného hlavičkového súboru. V tejto kapitole sú stručne popísané jednotlivé funkcie.

### 3.1. Main()

Základná funkcia main() sa delí do viacerých častí a volá priamo alebo nepriamo všetky funkcie opísané nižšie.

V prvom rade funkcia main() za pomoci volania funkcie getopt() zisťuje či boli zadane argumenty programu a zároveň overuje správnosť argumentov (či počet argumentov sa rovná minimálnemu počtu povinných argumentov).

Ako ďalšie funkcia volá viac krát funkciu resolver(). Viackrát z toho dôvodu aby bolo získané čo najviac informácií o zadanej doméne / IP adrese. V prípade kde užívateľ zadal na vstup IP adresu som si pre preklad na doménu pomohol kódom z [2].

Po vypísaní DNS záznamov funkcia vytvára socket pre pripojenie na daný whois server. V prípade neúspešného pripojenia program ukončuje a oznamuje chybovú hlášku. V prípade úspechu sa získava IP adresa domény o ktorej chce užívateľ získať informácie, pomohol som si stránkou [3].

Alokuje sa miesto pre získanú odpoveď od servera a posiela sa na sever dotaz obsahujúci IP adresu získanú v kroku vyššie.

Ak je alokované miesto malé a špecifické informácie z odpovede sa do neho nezmestia re-alokuje sa. Získavanie informácií od servera beží v cykle pokiaľ server neodošle EOF (poslal všetky dostupné informácie k danej IP adrese). Nakoľko niektoré servery posielajú duplicitné informácie do premennej ktorá obsahuje odpoveď od servera, sa porovnáva či táto informácia už bola obsiahnutá v odpovedi, ak áno informácia sa nezapíše a pokračuje sa ďalej v cykle.

Na získanú odpoveď sa zavolá funkcia print\_whois() ktorá ešte raz skontroluje získané informácie a vypíše ich v skupinách, aby užívateľ mal všetky informácie rovnakého druhu pod sebou.

V poslednom kroku funkcie main() sa uvoľní alokované miesto pre odpoveď a program sa korektne ukončí.

### 3.2. Help()

Táto funkcia má len informačný význam. Informuje užívateľa o správnom použití argumentov. Je volaná len v prípade keď je program spustený so zlými argumentami, volaná funkciou err\_arguments().

### 3.3. Err\_arguments()

Funkcia s návratovým typom integer. Ako už vyplýva z názvu funkcie stará sa o korektné ukončenie programu pri chybnom zadaní argumentov. Vypíše na štandardný chybový výstup chybovú hlášku. Ako návratovú hodnotu vždy vracia 1.

### 3.4. Error\_exit(int code, string str)

Funkcia pre ukončenie programu v prípade chyby. Vďaka svojim argumentom má možnosť meniť návratovú hodnotu a tak aj chybovú hlášku ktorá bude vypísaná na štandardný chybový výstup.

### 3.5. Set\_dns\_server(int \*dflag, char \*dns\_server)

Funkcia ktorá na základe ukazatela na dflag, vyhodnotí či bol zadaný DNS server (ak áno dflag má hodnotu 1) a tento DNS server zapíše do \_res štruktúry na prvé miesto. Vďaka tomu sa zaručí preklad cez tento server.

### 3.6. Resolver(char \*dname, int \*dflag, char \*dns\_server)

Funkcia slúži na dotazovanie DNS resolvera. Spracúvajú sa tu taktiež jednotlivé odpovede, pre ktoré sú potom volané samostatné funkcie pre ich formátované vypísanie. Funkcia je inšpirovaná [4].

Využíva sa v nej funkcia res\_search() ktorá je volaná v cykle, vďaka cyklu sa funkcia dotazuje na všetky potrebné údaje (A, AAAA, SOA, ...). Menenie dotazovaných údajov je zabezpečené jednoduchým switchom a prepisovaním premennej nType.

### 3.7. Printf\_cname(string s, int \*flag)

Funkcia vypíše na štandardný výstup kanonické meno, flag zaručuje neduplikovaný výpis.

### 3.8. Print\_soa\_admin\_email(string s)

Funkcia vypíše na štandardný výstup SOA záznam a emailovú adresu admina.

### 3.9. Print\_ns(string s)

Funkcia vypíše na štandardný výstup NS záznam.

### 3.10. Print\_mx(string s)

Funkcia vypíše na štandardný výstup MX záznam.

### 3.11. Print\_regex(regex rx, char \*output, int \*flag)

Do funkcie sú predané informácie ktoré sú uložené v premennej output. Za pomoci regulárnych výrazov sa vypíše konkrétny riadok s odpovedajúcou informáciou a následne na túto akciu sa prepíše hodnota ukazatela ukazujúceho na out\_flag vo funkcií print\_whois().

### 3.12. Print\_whois(char \*output)

Funkcia je volaná tesne pred ukončením funkcie main() v prípade že dotazovaný server odpovedal a tak naplnil premennú output informáciami, ktoré budú za pomoci vytvorených regulárnych výrazov v tejto funkcii poslané do funkcie print\_regex() kde sa potrebné informácie vypíšu.

V prípade že žiadne informácie nepodliehajú žiadnemu regulárnemu výrazu a teda sever nedrží potrebné záznamy, premenná vytvorená v tejto funkcii s názvom `out_flag` ostáva nastavená na `false` a vypíše sa hláška ktorá informuje o absencii potrebných dát zo servera.

## 4. Návod na použitie

### 4.1. Preloženie programu

Pre preloženie programu je vytvorený Makefile ktorý stačí spustiť príkazom `make`. Prípadne je možnosť spustiť program pomocou Makefilu a predom nastavených hodnôt argumentov programu:

- `make run` spúšťa program s argumentami:  
`-q www.fit.vutbr.cz -w whois.ripe.net`
- `make run2` spúšťa program s argumentami:  
`-q www.mobilmania.cz -d 8.8.8.8 -w whois.iana.org`
- `make run3` spúšťa program s argumentami:  
`-q fit.vutbr.cz -d 208.67.222.222 -w whois.arin.net`
- `make run4` spúšťa program s argumentami:  
`-q www.seznam.cz -d 8.8.8.8 -w whois.arin.net`
- `make runip` spúšťa program s argumentami:  
`-q 2001:67c:1220:809::93e5:917 -d 8.8.8.8 -w whois.ripe.net`
- `make runip2` spúšťa program s argumentami:  
`-q 147.229.9.23 -d 8.8.8.8 -w whois.iana.org`
- `make runip3` spúšťa program s argumentami:  
`-q 147.229.9.23 -w whois.arin.net`
- `make runipip` spúšťa program s argumentami:  
`-q 147.229.9.23 -w 193.0.6.135`

### 4.2. Spustenie programu

Po preklade ktorý je opísaný v predchádzajúcej podkapitole sa vytvorí spustiteľný program *isa-tazatel*. Program sa spúšťa s 2 povinnými argumentami `-q` a `-w` a jedným voliteľným argumentom `-d`.

*./isa-tazatel -q www.fit.vutbr.cz -d dns.google.com -w whois.ripe.net*

- `-q <IP|hostname>`
- `-w <IP|hostname WHOIS serveru>`, ktorý bude dotazován
- `-d <IP>`, ktorý bude dotazovaný, nepovinný argument pričom implicitne sa používa DNS resolver v operačnom systéme

## 5. Testovanie

Po testovaní v kóde som stále nechal kontrolné výpisy ktoré je možné zapnúť vymazaním `#define NDEBUB` ktorý sa nachádza v súbore `functions.h`. Program som testoval primárne s argumentami ktoré sú uvedené v jednotlivých spustení s prekladom v sekcii 4.1. Všetky servery až na `whois.iana.org` (ďalej „iana“) vždy odpovedali a ja som tak dokázal vypísať aspoň dáke informácie od nich ktoré boli potrebné. Iana posielala niekedy balast pred jednotlivými informáciami vtedy na štandardný výstup je odoslaná správa o možnom nerozoznaní odpovede regulárnym výrazom. Svoje výstupy programu som porovnával s online whois vyhľadávačom <http://www.dnslookup.online/>.

## 6. Bonusy

Ako bonus som implementoval prácu s DNS resolverom kde si užívateľ za pomoci prepínača môže vybrať DNS resolver. Na tento bonus využívam svoju funkciu ktorú som zmenil už vyššie, [`set\_dns\_server\(\)`](#). Na základe toho či bol zadaný prepínač `-d`, funkcia vyhodnotí či sa bude brať implicitný DNS resolver alebo užívateľ zadal ktorý sa má preferovať. Zadaná IP adresa sa zapíše do `_res` štruktúry na prvé miesto, `_res` štruktúra ktorá sa pred tým inicializuje a nastaví sa počet IP adres v tejto štruktúre na 1.

## 7. Záver

Podarilo sa mi naimplementovať skoro celé zadanie s jedným bonusom. Pri koncovom testovaní projektu som nezaznamenal žiadnu inú chybu ako tú ktorú som nevedel vyriešiť. Tou chybou je nemožnosť programu vypísať záznam PTR, program nie je schopný vojsť do možnosti switchu pre PTR záznam, z dôvodu ošetrenia podmienok pred tým. Na tomto projekte som strávil desiatky hodín zo začiatku som sa v tom hľadal no čím viac večerov som nad tým strávil tým ma to aj viac bavilo. Zdokonalil som si svoje programovanie najmä v jazyku C a o niečo som si zlepšil aj C++ a vytváranie viacerých súbor. Lepšie som pochopil ako pracovať s Wiresharkom a debugovacím prostredím gdb. Taktiež som sa naučil pracovať s knižnicou pre regulárne výrazy a spracovávať argumenty za pomoci funkcie `getopt`.

## 8. Literatúra

- [1] P. Matoušek, Sítové služby a jejich architektúra, VUTIUM, 2014.
- [2] „Cboard,“ 3 Máj 2012. [Online]. Available: <https://cboard.cprogramming.com/c-programming/169902-getnameinfo-example-problem.html>.
- [3] „Github,“ 24 Október 2018. [Online]. Available: <https://github.com/angrave/SystemProgramming/wiki/Networking,-Part-2:-Using-getaddrinfo>.
- [4] „Stackoverflow,“ 18 Júl 2018. [Online]. Available: <https://stackoverflow.com/questions/51401982/dns-retrieving-host-ip-address-using-resolv-h>.