

# Vysoké učení technické v Brně

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Síťové aplikace  
a  
správa sítí  
2019/2020

Varianta: DNS resolver

# Obsah

Úvod.....	3
DNS servery obecně.....	3
Účely projektu.....	3
Implementace.....	4
Zpracování argumentů.....	4
Prerekvizity.....	4
Komunikace.....	4
Výpis hlavičky a dotazu.....	4
Výpis odpovědí.....	5
Poznámky k implementaci.....	6
Použití a syntaxe programu:.....	6
Souborová struktura.....	6
Hlavičkový soubor.....	6
Odkazy.....	7

# Úvod

Cílem projektu bylo vytvořit aplikaci v jazyce C/C++, která dokáže komunikovat s DNS serverem. Cílem komunikace bylo získat informace od serveru o adrese která byla zadána a následně převést Data ze serveru do prezentovatelné podoby.

## DNS servery obecně

Projekt se zabývá komunikací s DNS serverem a zpracováním odpovědi. DNS servery umožňují oboustraný překlad z domény na ip adresu a obráceně. Tento proces je vyžadován pro jednodušší zpracování domény při přístupu na některou ze známých adres jako například "[www.google.com](http://www.google.com)". Při zadání již zmíněné adresy si počítač zjistí zda sám nezná ip adresu toho serveru aby s ním mohl navázat komunikaci. Při chybějící ip adrese se počítač začne dotazovat DNS serveru který mu byl přidělen manuálně, a nebo jej dostal od místního směrovače. Zařízení poté zašle dotaz na tento DNS server a očekává od něj ip adresu domény na kterou se chce připojit. Po získání domény je uživatel připojen na zmíněnou doménu, ale není mu řečeno že byl připojen pomocí ip adresy[1].

## Účely projektu

Aplikace je navržena tak aby dokázala zpracovat komunikaci se serverem. Při spuštění je nutné zadat adresu DNS serveru a adresu o které chceme získat informace. Zde však schopnosti aplikace nekončí a již zmíněnému dotazu lze přidat příznaky které specifikují co od serveru očekáváme za odpověď. Odpověď je zpracována do čitelné podoby a ty nejpodstatnější informace jsou vypsány na standardní výstup. Tyto informace většinou nevidáme a dokážou nám říct více jak o dotazované adrese tak i o samotném DNS serveru.

# Implementace

## Zpracování argumentů

Program začíná ve funkci `"main()"` kde jsou definice proměnných, které naplní zadané argumenty spuštění. Následuje zpracování argumentů ve funkci `"parseArgs()"`. Nejdůležitější částí této sekce je funkce `"getopt()"`, která nesmírně usnadňuje zpracování argumentů. Nastavují se zde příznaky pro budoucí práci a kontrolují se vstupní parametry. Pro dodatečné zpracování argumentů slouží `"addressToQname()"`, zde se převede adresa na formát zapisatelná do DNS dotazu. Při nastavení příznaku `"-x reverzní dotaz"` se nejdříve převde na reverzní ip formát[2]. V obou případech se adresa převede z tečkovaného formátu `"www.google.com"` na formát vhodný pro komunikaci `"3www6google3com"` kde čísla reprezentují délku následujícího slova.

## Prerekvizity

Dále se přesunme do funkce `"comunication()"`. Před samotnou okmunikací je nutnost nastavit jisté prerekvizity, a to vytvořit buffer který bude reprezentovat celou zaslanou zprávu, získat ip adresu DNS serveru, přičemž v mé implementaci jsem se rozhodl použít `"getaddrinfo()"`, a vytvořit komunikační socket, který zprávu odešle a poté do něj zprávu i přijmeme. V tomto bodě už máme možnost něco naspát DNS serveru, jenže nám chybí nějaká strukturovaná data kterému můžeme poslat.

## Komunikace

Komunikace takového typu vyžaduje přesný formát informací na správném místě. Začínáme zde s naplněním DNS hlavičky která obsahuje základní informace, například že se jedná o dotaz, jaké je id, také jaké informace nás vlastně zajímají. Mezi tyto patří zda vyžadujeme rekurzi, nebo jestli se jedná o reverzní dotaz. Následuje samotný dotaz, jeho jméno, typ a třída. Dotaz nyní finálně odešleme. Čekáme na odpověď serveru, přičemž maximální doba čekání je 5 vteřin.

## Výpis hlavičky a dotazu

Pokud vše proběhlo v pořádku, dostaneme od serveru odpověď. Data v odpovědi však nejsou prezentovatelné a je nutno je naformátovat do čitelné podoby. Funkce `"printHeader()"` a `"printSection()"` zajišťují toto zpracování a výpis. První funkce vypíše hlavičku odpovědi, která obsahuje informace o tom zda byla doména v seznamu autoritativních domén serveru. Infomaci o tom jestli se server rekurzivně dotazoval jiného serveru. A také jestli zpráva musela být zkrácena. Dále je vypsán dotaz, který jsem serveru zapsaly. Objevuje se informace o adrese na kterou jsme se ptali, typ dotazu a třída dotazu.

## Výpis odpovědí

Data se třídí na "Answer", "Authority", "Additional". Všechny tři sekce používají tutéž funkci pro zpracování dat a výpis "printSection()". Pro každý s těchto záznamů se načítají data pomocí "readAnswer()", která postupně jde v datech a zapisuje informace do předchystaných struktur. V této funkci je hlavní logika zpracování dat odpovědi. Data nemusí být ucelená a často se stává že se ve správě skáče na jinou pozici. Pro tento problém byla potřeba implementovat komplexní ukládání pozic ukazatele pomocí záložky a odskakování posle zapsané hodnoty. Po dočtení jednoho názvu je nutné doménu převést do dormálního tečkovaného zápisu. Načtou se informace o názvu, typu, třídě, době platnosti odpovědi a data. Všechny tyto informace jsou vypsány na standartní výstup a program je ukončen.

# Poznámky k implementaci

Program vyžaduje aby argument s adresou byl zadán jako poslední. Dále bezprostředně po příznaku "-s" musí následovat adresa serveru. Podobně bezprostředně po "-p" musí následovat číslo portu. Přidal jsem možnost příznaku "-h", který nespustí program a pouze vypíše instrukce pro spuštění programu. Program podporuje výpisy typu A, NS, CNAME, PTR, MX a AAAA. Podle mého hledání jsou tyto nejpoužívanější typy a ostatní jsou buď raritní nebo zastaralé. Pro třídy je možné vypsát IN, CS, CH a HS. Program byl úspěšně přeložen a testován na VUT Fit serveru merlin i eva.

## Použití a syntaxe programu:

Po překladu se vygeneruje soubor "dns", který lze spustit pomocí:

```
./dns [-h] [-r] [-x] [-6] -s server [-p port] adresa
```

-s server: Jedná se o ip adresu nebo doménu DNS serveru na který bude dotaz zaslán. Po příznaku musí ihned následovat server.

adresa: Dotazovaná adresa na kterou se ptáme, může to být doména či ip adresa viz níže. Adresa musí být zadána jako poslední prvek.

-h: Vypíše na standartní výstup nápovědu pro použití programu.

-r: Rekurze serveru je vyžadována. Pokud server může, tak zpracovává dotaz rekurzivně, čímž se ptá ostatních serverů na stejný dotaz, pokud sám nezná odpověď.

-x: Dotaz je reverzní. Namísto argumentu adresy je nutné zadat ip adresu na kterou se ptáme a server nám dá odpověď ve formátu domény.

-6: Dotaz je typu AAAA. Tím změním v otázce typ ip adresy na kterou se dotazujeme z ipv4 na ipv6.

-p port: Dotaz bude zaslán na číslo daného portu. Pokud server na tomto portu naslouchá, tak komunikace probíhá identicky jen s jiným portem.

Výchozí port je 53. Po příznaku musí ihned následovat číslo portu. Příznaky v "[]" jsou volitelné.

## Souborová struktura

Program se skládá pouze ze dvou souborů a to z "dns.cpp" a hlavičkového souboru "dns.h". Původní návrh aplikace byl rozdělen do více souborů, ale v souvislosti s tím že spustitelný kód má pod 500 řádků a to s komentáři, rozhodl jsem se vše vložit do jednoho souboru. Déle projekt obsahuje "Makefile" pro přeložení projektu, soubor "test.sh", což je jednoduchý testovací soubor, soubor "README" se souhrnem informací o programu a dokumentaci, kterou právě dočítáte.

## Hlavičkový soubor

Obsahuje definice struktur, které usnadnili práci se zapisováním a čtením dat. Některé proměnné v hlavičkovém souboru jsou redundantní a jsou pouze na udržení přehlednosti. Hlavičkový soubor bylo snadné sestavit, díky dostatečné manuálové dokumentaci[4] s doplňkovými informacemi z internetu[5]. Jediné věc se kterou jsem měl v této části problém byla proměnná "flags" v DNS hlavičce, která obsahovala příznaky dotazu a odpovědi. Díky pochopení problematiky, a uvědomnění si, že se jedná o hexadecimální číslo, jsem dokázal nastavovat a číst příznaky pomocí maskování.

# Odkazy

- [1] Domain Name System  
[https://cs.wikipedia.org/wiki/Domain\\_Name\\_System](https://cs.wikipedia.org/wiki/Domain_Name_System)
- [2] Reverse DNS lookup  
[https://en.wikipedia.org/wiki/Reverse\\_DNS\\_lookup](https://en.wikipedia.org/wiki/Reverse_DNS_lookup)
- [3] List of DNS record types  
[https://en.wikipedia.org/wiki/List\\_of\\_DNS\\_record\\_types](https://en.wikipedia.org/wiki/List_of_DNS_record_types)
- [4] RFC: DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION  
<https://tools.ietf.org/html/rfc1035#page-12>
- [5] Deep dive into DNS messages:  
<https://amriunix.com/post/deep-dive-into-dns-messages/>