



Projektová dokumentace

DNS resolver

Síťové aplikace a správa sítí

6. listopadu 2020

Jiří Křištof (xkrist2)

Obsah

1	Úvod	1
2	Teoretický základ	1
3	Návrh a implementace	1
3.1	Parsování vstupních argumentů	1
3.2	Módy aplikace	1
3.3	Zpracování souboru filtru	1
3.4	Vytvoření serveru	1
3.5	Chybové stavy	2
4	Překladový systém	2
5	Návod na použití	2
6	Přílohy	2

1 Úvod

Cílem projektu je vytvořit aplikaci dns resolver v jazyce C/C++. Program naslouchá na vybraném portu a filtruje dns dotazy. Program podporuje dotazy typu A pomocí zaslané pomocí UDP protokolu transportní vrstvy. Pro ostatní dotazy je zaslána chybová odpověď. Program načítá z dodaného souboru domény určené k vyfiltrování. Pokud je v dotaze uveden dotaz na doménové jméno (případně i subdoménu), které je uvedeno ve filtru, pak je tento dotaz zahozen. V opačném případě je dotaz dále přeposlán na specifikovaný dns server a zpráva z něj je vrácena původnímu tazateli.

2 Teoretický základ

3 Návrh a implementace

Aplikace je rozdělena do jednotlivých souborů obsahující implementaci tříd, které implementují výsledný systém. Zdrojové soubory se nachází v adresáři `src`.

3.1 Parsování vstupních argumentů

Parsování vstupních argumentů předané programu z příkazového řádku zajišťuje třída `argparse`. Při vytváření instance třídy jsou konstruktoru předány veškeré parametry z příkazové řádky. Po zpracování je možné se pomocí k tomu určených metod dotazovat na potřebné údaje – IP adresu dns serveru, jméno souboru obsahující filtr a případně číslo portu.

V případě, že je programu předáno doménové jméno dns serveru, pak je jeho IP adresa dohledána pomocí funkce `gethostbyname`.

Neplatné vstupní argumenty jsou ignorovány. V případě, že je programu zadáno neplatné číslo portu, pak program využije výchozí hodnotu 53. Program končí v chybovém stavu, pokud nezíská informaci o IP adrese dns serveru a soubor obsahující filtr.

3.2 Módy aplikace

Aplikace může pracovat ve dvou různých režimech. Ve výchozím nastavení pracuje server v tichém režimu – na standardní výstup nejsou vypisovány žádné informace, případné chybové hlášky jsou vypisovány na standardní chybový výstup.

Aplikace může být přepnuta do módu, v němž je uživatel informován o jednotlivých krocích programu zprávami směřovanými na standardní výstup. Mód může být aktivován využitím přepínače `-v`. Výpis informací je implementován třídou `verbose`.

3.3 Zpracování souboru filtru

Zpracování vstupního souboru obsahující filtr je implementováno třídou `filter`. Při vytváření objektu třídy je konstruktoru předán název souboru obsahující filtr. Konstruktor ukládá jednotlivé řádky obsahující doménová jména k vyfiltrování do vektoru textových řetězců.

Při zpracovávání souboru jsou vynechány prázdné řádky a řádkové komentáře začínající symbolem `#`. Před zpracováním každého řádku je prvně provedeno odebrání bílých znaků na začátku a na konci daného řádku.

V případě, že soubor neexistuje, je program ukončen a na standardní chybový výstup je uvedena informace o nastalé chybě.

3.4 Vytvoření serveru

Zajištění síťové komunikace je implementováno pomocí třídy `server`. Konstruktoru objektu třídy `server` jsou předány hodnoty čísla portu, na němž má server čekat na zprávy, a ip adresa dns serveru, na nějž mají být

směřovány dotazy, které nebudou vyfiltrovány. Při instanciaci je detekována verze protokolu IP. Po extrakci dat je vytvořena struktura `sockaddr_in` reprezentující soket serveru. Soket je poté otevřen a připojen k vybranému portu.

Po vytvoření instance třídy `server` je možné spustit pomocí metody `run_server` samotný server, který čeká na dns dotazy, které filtruje a případně přeposílá dál. Metoda také implementuje vytváření odpovědí a případných chybových zpráv.

V případě problému při vytváření serveru či zasílání paketů je program ukončen s určeným chybovým kódem.

3.5 Chybové stavy

Pokud se program dostane do chybového stavu, pak tento stav řeší metody třídy `err_handler`. Třída informuje uživatele o problému. Pokud nelze dále pokračovat a chybový stav ošetřit, pak je program ukončen s chybovým návratovým kódem. Chybové kódy jsou uvedeny v tabulce 1.

4 Překladačový systém

Překladačový systém je vytvořen pomocí utility `make`. Tdrojové soubory je možné přeložit pomocí připojeného souboru `Makefile`. Pro překlad je využíván kompilátor `gcc`. Po překladu je v kořenovém adresáři umístěn spustitelný soubor `dns`.

Pomocí příkazu `make test` je možné spustit testy, které byly využity při testování aplikace.

5 Návod na použití

6 Přílohy

Chybový kód	Popis chyby
0	Program byl ukončen bez detekování chyby
1	Problém při parsování vstupních argumentů
2	Formát IP adresy je neplatný, nebo nebyla získána IP adresa
3	Nedefinovaná hodnota portu, program využije výchozí port 53
4	Program nenalezl uvedený soubor s filtrem
5	Špatná knostrukce souboru s filtrem
6	Chyba při otevírání soketu
7	Chyba při připojení soketu na port
8	Chyba příchozích dat
9	Chyba zaslání dat
10	Chyba zaslání dat – byla zaslána pouze část dat

Tabulka 1: Návratové kódy programu