

ZADÁNÍ:

Úkolem projektu je:

[1] Nastudovat si detaily protokolu DNS, systému DNS obecně a uvést informace relevantní pro řešení projektu v projektové dokumentaci (až 6 bodů).

[2] Naprogramovat C/C++ nástroj, který se za pomoci síťové knihovny BSD sockets dotazuje systému DNS a realizuje překlad doménových jmen a IP adres (až 14 bodů).

KONVENCE ODEVZDÁVANÉHO ZIP ARCHIVU xlogin00.zip

- dokumentace.pdf - výstup zadání [1]
- README - krátký textový popis programu s případnými rozšířeními/omezeními
- Makefile - pro překlad zdrojového kódu zadáním make
- *.c, *.cpp, *.cc, *.h - zdrojové a hlavičkové soubory výstup zadání [2]

UPŘESNĚNÍ ZADÁNÍ:

Ad [1]

Dokumentace bude obsahovat titulní stranu, obsah a logicky strukturovaný text, v rámci kterého se objeví: uvedení do problematiky systému DNS, přehled nastudovaných informací z literatury (formát DNS dotazu, odpovědi, způsob komprese zpráv, rekurzivní/iterativní způsob dotazování), návrh aplikace, popis významných a zajímavých pasáží implementace (řešení timeoutu), návod pro použití programu, popis případných rozšíření nad rámec zadání a použitá literatura.

Při psaní dokumentace se obecně řiďte

- fakultními pokyny pro psaní práce (http://www.fit.vutbr.cz/info/szz/psani_textu.php.cs) a
- pravidly pro bibliografické citace (http://www.fit.vutbr.cz/info/szz/bib_citace.php.cs).

Ad [2]

Konvence jména aplikace a použití:

`./ipk-lookup [-h]`

`./ipk-lookup -s server [-T timeout] [-t type] [-i] name`

- h (help) - volitelný parametr, při jeho zadání se vypíše nápověda a program se ukončí.
- s (server) - povinný parametr, DNS server (IPv4 adresa), na který se budou odesílat dotazy.
- T (timeout) - volitelný parametr, timeout (v sekundách) pro dotaz, výchozí hodnota 5 sekund.
- t (type) - volitelný parametr, typ dotazovaného záznamu: A (výchozí), AAAA, NS, PTR, CNAME.
- i (iterative) - volitelný parametr, vynucení iterativního způsobu rezoluce, viz dále.
- name - překládané doménové jméno, v případě parametru -t PTR program na vstupu naopak očekává IPv4 nebo IPv6 adresu.

Ve výchozím stavu aplikace pokládá pouze rekurzivní DNS dotazy. V případě vynucení iterativního dotazu (parametr -i) slouží zadaný server (parametr -s, respektive -p), pouze k nalezení některého z kořenových DNS serverů, respektive ke zjištění jeho IP adresy. Následně se už aplikace postupně dotazuje pouze takto nalezených (kořenových) serverů, na které byla vždy delegována správa části

DNS prostoru jmen, a to až do získání požadované odpovědi, zjištění, že záznam neexistuje, případně vypršení timeoutu.

Formát výstupu: Na standardní výstup vypište výsledek překladu, respektive dílčí odpovědi (následování CNAME, použité NS a A záznamy při iterativní rezoluci). Jednotlivé záznamy vypisujte v následujícím tvaru:

```
[ ]+ IN [ ]+ [ ]+ \n
```

Blíže viz příklady použití. V případě chyby vypište na standardní chybový výstup uživatelsky srozumitelnou chybovou zprávu a program ukončete.

Návratová hodnota: Program vrací hodnotu 0 v případě úspěchu, hodnotu 1 v případě neúspěchu (neexistující záznam, vypršení timeoutu) a hodnotu 2 pro chybně zadané parametry při spuštění.

Další doporučení/omezení/pokyny:

- Implementovaná konzolová aplikace bude povinně vypracována v jazyce C/C++, využít však můžete libovolné a v systému dostupné standardní knihovny.
- V projektu se očekává vytvoření DNS dotazu, jeho odeslání na server a zpracování odpovědi zcela ve vlastní režii, pouze s využitím BSD socketů. Z tohoto důvodu je zakázáno použití funkcí `getaddrinfo()`, `gethostbyname()` a podobných, které řeší překlad doménových jmen.
- Uvažujte pouze třídu IN (Internet) a pouze IPv4 a UDP komunikaci s DNS servery.
- Pro snadné parsování vstupních parametrů doporučuji použít funkci `getopt()`, pro načtení, výpis IPv4 nebo IPv6 adres potom funkce: `inet_ntop()`, `inet_pton()`, případně `inet_aton()`, `inet_ntoa()`.
- Aplikace v žádném případě nesmí skončit s chybou `SEGMENTATION FAULT` ani jiným násilným systémovým ukončením (např. dělení nulou).
- Zdrojový kód vytvořeného programu by měl být řádně okomentovaný. Pokud přejímáte krátké pasáže zdrojového kódu z různých tutoriálů či příkladů z Internetu (ne mezi však mezi sebou), je povinné správně vyznačit tyto části a uvést jejich původní autory dle licenčních podmínek, kterými se distribuce daných zdrojových kódů řídí. V případě nedodržení bude na projekt nahlíženo jako na plagiát!
- Inspirujte se a vyzkoušejte si práci s nástroji `nslookup` a `dig` (parametr `+trace`). Zachyťte si daný provoz pomocí `Wireshark`.
- Jako referenční stroj pro překlad a otestování využijte školní server `merlin.fit.vutbr.cz`. Výsledky vaší implementace by však měly být co možná nejvíce multiplatformní a přenositelné i mezi různými unix-like operačními systémy.
- Pokud v projektu nestihnete implementovat všechny požadované vlastnosti, je nutné veškerá omezení jasně uvést v dokumentaci a v souboru `README` - na dokumentovanou chybu se rozhodně nahlíží v lepším světle než na nedokumentovanou.

PŘÍKLADY POUŽITÍ:

```
~$ ./ipk-lookup -s 8.8.8.8 www.fit.vutbr.cz  
www.fit.vutbr.cz. IN A 147.229.9.23
```

(exit code 0)

```
~$ ./ipk-lookup -s 8.8.8.8 -t AAAA -i www.fit.vutbr.cz.  
. IN NS j.root-servers.net.  
j.root-servers.net. IN A 192.58.128.30  
cz. IN NS c.ns.nic.cz.  
c.ns.nic.cz. IN A 194.0.14.1  
vutbr.cz. IN NS pipit.cis.vutbr.cz.  
pipit.cis.vutbr.cz. IN A 77.93.219.110  
fit.vutbr.cz. IN NS guta.fit.vutbr.cz.  
guta.fit.vutbr.cz. IN A 147.229.9.11  
www.fit.vutbr.cz. IN AAAA 2001:67c:1220:809::93e5:917  
(exit code 0)
```

```
~$ ./ipk-lookup -s 8.8.8.8 www4.fit.vutbr.cz.  
www4.fit.vutbr.cz. IN CNAME tereza.fit.vutbr.cz.  
tereza.fit.vutbr.cz. IN A 147.229.9.22  
(exit code 0)
```

```
~$ ./ipk-lookup -s 8.8.8.8 -t CNAME www4.fit.vutbr.cz.  
www4.fit.vutbr.cz. IN CNAME tereza.fit.vutbr.cz.  
(exit code 0)
```

```
~$ ./ipk-lookup -s 8.8.8.8 -t AAAA www4.fit.vutbr.cz.  
www4.fit.vutbr.cz. IN CNAME tereza.fit.vutbr.cz.  
(exit code 1)
```

```
~$ ./ipk-lookup -s 8.8.8.8 -t PTR 2001:67c:1220:8b0::93e5:b013  
3.1.0.b.5.e.3.9.0.0.0.0.0.0.0.0.b.8.0.0.2.2.1.c.7.6.0.1.0.0.2.ip6.arpa. IN PTR merlin6.fit.vutbr.cz.  
(exit code 0)
```

```
~$ ./ipk-lookup -s 8.8.8.8 -t PTR -i 147.229.13.238  
. IN NS m.root-servers.net.  
m.root-servers.net. IN A 202.12.27.33  
in-addr.arpa. IN NS b.in-addr-servers.arpa.  
b.in-addr-servers.arpa. IN A 199.253.183.183  
147.in-addr.arpa. IN NS r.arin.net.  
net. IN NS j.gtld-servers.net.  
j.gtld-servers.net. IN A 192.48.79.30  
arin.net. IN NS ns1.arin.net.  
ns1.arin.net. IN A 199.212.0.108  
r.arin.net. IN A 199.180.180.63  
229.147.in-addr.arpa. IN NS ns.ces.net.  
ces.net. IN NS nsa.ces.net.  
nsa.ces.net. IN A 195.113.144.205
```

ns.ces.net. IN A 195.113.144.233
13.229.147.in-addr.arpa. IN NS rhino.cis.vutbr.cz.
rhino.cis.vutbr.cz. IN A 147.229.3.10
238.13.229.147.in-addr.arpa. IN PTR pckucerajan.fit.vutbr.cz.
(exit code 0)

LITERATURA:

- P. Mockapetris, RFC 1034: Domain names - concepts and facilities,
<https://tools.ietf.org/html/rfc1034>.
- P. Mockapetris, RFC 1035: Domain names - implementation and specification,
<https://tools.ietf.org/html/rfc1035>.
- S. Thomson, et al., RFC 3596: DNS Extensions to Support IP Version 6,
<https://tools.ietf.org/html/rfc3596>.