מבוא לתקשורת מחשבים - תרגיל מעשי #2

מגישים: 322522244, 313479263

מסמך דוקומנטציה - nsclient

בתרגיל זה נרצה לממש DNS client. התוכנה תקבל כארגומנט את כתובת הIP של שרת הDNS ולאחר מכן תפתח ממשק משתמש בו המשתמש מזין שם domain (string). על התוכנה לבדוק את תקינות הקלט ולהחזיר למשתמש את כתובת הIPV4 של הdomain. בדומה לgethostbyname(), התכנית ממלאה את struct hostent remoteHost, וממנה מחלצת את כתובת הIP הנחוצה.

**main.c:**

בחלק זה מתבצעת ההתקשרות הראשונית עם שרת הDNS. זאת באמצעות פתיחה של socket, והתחברות על ידי פונקציית socket\_connect().

לאחר מכן התכנה תפתח את ממשק המשתמש על מנת שהמשתמש יוכל להזין את שם הdomain עבורו המשתמש רוצה לקבל את כתובת הIP.

validateHost(hostname): לאחר קבלת המחרוזת מהמשתמש, התכנית תבצע בדיקת תקינות לשם domain, עבור המוסכמות שנקבעו.

dnsQuery(hostname): לאחר הבדיקה, התכנית תכניס את שם הdomain לפונקציה זו על מנת לקבל ממנה את struct hostent remoteHost.

assertDnsQueryResultIsValid(remoteHost, hostname): פונקציה עבור דיבאג. היא מקבלת את המבנה שקיבלנו ובודקת האם הוא תואמת למבנה שהיינו מקבלים עם הפונקציה gethostbyname(). במידה והמבנים לא תואמים, היא מפילה את התכנית.

**socket\_utils.c:**

socket\_initialize(WSADATA\* wsaData): פונקציה זו מאתחלת את הsocket באמצעות WSAStartup().

socket\_connect(SOCKET\* sock, const char\* dest, const u\_short port): פונקציה זו מקבלת את הsocket ואת הport ואת היעד ומתחברת לשרת הDNS. בנוסף, נשתמש בsetsockopt() על מנת להחיל TIMOUT על זמן קבלת התגובה מהשרת כפי שהוגדר לנו בתרגיל.

**dns\_utils.c:**

בקובץ זה נבצע את שליחת השאלה לשרת הDNS וקבלת התשובה ממנו. בפרויקט זה אנו מניחים שישנה שאלה אחת ותשובה אחת בלבד. שליחת השאלה, query (string), תתבצע בפורמט RFC-1035. לשם כך בנינו את struct dns\_header\_t, בו יוחזרו לנו מספר השאלות והתשובות וRCODE אותו נצטרך לבדוק. את struct question\_t שמכיל את שם הדומיין מקודד על פי הפורמט, את סוג הquery ואת הclass שלו. בנוסף struct response\_t המכילה את השדות אותם על התכנית לקבל מהשרת על מנת למלא את remoteHost כראוי.

struct hostent\* dnsQuery(const char\* hostname): פונקציה זו מקבלת את שם הדומיין ומחזירה את struct remoteHost לאחר הזנת כל הנתונים למבנה. על ידי השימוש בפונקציות הבאות:

createDnsQueryBuf(hostname, &sizeof\_query,&sizeof\_qname) : מקבלת את שם הדומיין ומחזירה את הquery, את גודלו ואת גודל שם הדומיין לאחר קידוד על פי פורמט RFC-1035. בפונקציה זו, מתבצעת ההשמה של dns\_header\_t על פי המוסכמות בתרגיל, והשמת question\_t, כאשר שדה qname מתקבל מפונקציית change\_question\_name() המקבלת את שם הדומיין ומקודדת אותו.

לאחר בניית הquery, הפונקציה dnsQuery() שולחת את הבאפר לDNS ומקבלת ממנו תגובה ושולחת את התגובה הזו לפונקציה הבאה:

parseDnsResponseBuf(response, sizeof\_response,sizeof\_qname): פונקציה זו מקבלת את התגובה שנשלחה מהשרת ומפסקת אותו ויוצרת את מבנה remoteHost.

פונקציה זו עובדת עם char\* reader. מחרוזת זו הינה העתק של response. במהלך הקוד, הפונקציה תמלא את המבנים על ידי הקצאת reader עם גודל המבנה ואז תקדם את המחרוזת בגודל המבנה הנתון על מנת למלא את המבנה הבא. כך למשל, מושם dns\_header\_t\* dns ולאחר מכן reader מקודם בגודל ההאדר כדי שנוכל למלא גם את מבנה question\_t\* ques. מבנה dns נכנס לפונקציית parseDnsHeaderFromResponse() לבדיקת ההאדר המוחזר; כלומר בדיקת הRCODE.

לאחר מבנה ques, reader מגיע לanswer בתוך תגובת השרת ומופעלת הפונקציה read\_qname\_wrapper(). פונקציה זו מקבלת את התגובה וגודלה, את הreader ופוינטר לremoteHost אותו הפונקציה ממלאה (את שדה h\_name) ומחזירה האם היא עברה בהצלחה. פונקציה זו בודקת את הביטים הראשונים של שדה name, אם מדובר ב'11', היא מחלצת את השם המקודד מתוך האופסט המוקצה בשדה name. אחרת, היא מחלצת את השם המקודד מqname ומקדמת בהתאם. בנוסף, השם המקודד נשלח לפונקציית read\_qname(), ושם שם הדומיין מקודד חזרה לפורמט הנקודות, שם מושם בפועל שדה h\_name בremoteHos.

על מנת למלא את h\_addr\_list בremotHost מתבצעת אלוקציה לh\_addr\_list אותו נמלא באמצעות reader. לאחר מכן נשלח לh\_addr\_list פוינטר לרשימה זו של הכתובות ולבסוף הפונקציה מחזירה את struct remotHost.

מבנה remoteHost חוזר לdnsQuery() שבודק אותו ומחזיר אותו לmain.