תיאור מערכת Ping + Ping and Watchdog timer

סקירת מערכת..................................................................................................................2

פונקציונליות המערכת.......................................................................................................2

איך מריצים......................................................................................................................2

תרשים זרימה לחלק של Ping.............................................................................................3

עץ החלטות למצבים של Ping and Watchdog timer ..........................................................4

הסברים על הקוד (ping.c)........................................................................................... 5-10

**סקירת מערכת**

החלק הראשון של המערכת מאפשר לבצע פקודת Ping לכתובת IP מסוימת על מנת לבדוק תקינות החיבור של נקודות הקצה הזאת לIP שהכנסנו לו.

בשלב הראשון נפתח RAW סוקט לכתובת IP שהוכנס בהרצה של המערכת , בשלב השני נשלח לכתובת IP דרך פרוטוקול ICMP הודעה מסוג ICMP-REQUEST בכדי לבדוק חיבור (הודעה זו מיצג נקראת "פינג") ובחזרה נקבל הודעת ICMP-ECHO-REPLY המכונה "פונג" שבאמצעות נוכל לנתח ולראות את תקינות החיבור שלנו ליעד מסויים ברשת.

בחלק השני של המערכת יש לנו רכיב שמדמה Watchdog timer שמכיל טיימר ומקבל כול פעם שבוצע פקודת פינג והגיע תשובה חזרה הודעה שבוצע פינג בהצלחה ומאפס את הטיימר שלו,

במידה וטיימר מראה שעבר מעל 10 שניות ולא הגיעה הודעת שבוצע פינג בהצלחה ה Watchdog סוגר את התוכנית ומדפיס הודעה שאין חיבור לכתובת יעד.

**פונקציונליות המערכת**

בחלק הראשון למערכת יש פונקציונליות של בדיקת תקינות החיבור בין מקור מסויים ברשת ליעד מסויים ברשת והצגת הנתונים הרלוונטים מתוך הפאקטה של ICMP כדי לנתח ולראות את הפרטים מהודעת ICMP-ECHO-REPLY שקיבלנו.

בחלק השני של המערכת אנחנו בודקים אם יש חיבור יציב ורציף אם כתובת היעד בכול 10 שניות באמצעות טיימר שסופר כול כמה שניות מגיעה חזרה הודעת ICMP-ECHO-REPLY בwatchdog .

במידה ולא הגיע ל watchdog הודעה שהתקבל פינג מעבר 10 שניות הוא שלח הודעה ללקוח שמצבע את הפינג שלא הגיע הודעת פינג מעל 10 שניות באמצעות הודעת "time out " וסוגר את הסוקט מולו.

ובמידה והכול בסדר ומגיע פינג רציף מתחת ל10 שניות כול פעם המערכת רצה בצורה אינסופית.

**איך מריצים**

המערכת רצה על מערכת הפעלה Linux בלבד.

מצורף למערכת קובץ Makefile שבו כתובים פקודות שמקמפלות את הקבצים ומכינים אותם להרצה.

לכן בכדי להריץ את החלק הראשון יש לבצע את השלבים הבאים :

1. להיכנס לטרמינל דרך הנתיק של התקייה ולהריץ את הפקודה maka all.
2. לרשום את הפקודה ./parta <ip addrees> ואיפה שרשום ip address נוסיף את הכתובת IP שאליה נרצה לבצע פקודת Ping.

בכדי להריץ את החלק השני יש לבצע את השלבים הבאים :

1. להיכנס לטרמינל דרך הנתיק של התקייה ולהריץ את הפקודה maka all.
2. לרשום את הפקודה ./watchdog להפעיל את התוכנית של הwatchdog.
3. לרשום את הפקודה ./partb <ip addrees> ואיפה שרשום ip address נוסיף את הכתובת IP שאליה נרצה לבצע פקודת Ping.

**תרשים זרימה לחלק הראשון**

קבלה דרך הmain את כתובת הIP של היעד

יצירת RAW סוקט עם היעד ברשת לפי הIP

שליחת הודעת ICMP REQUEST לכתובת יעד בפונקצית ping

חזרה על התהליך הצורה אינסופית

קבלת הודעת ICMP REPLY מכתובת היעד דרך RAW סוקט בפונקצית listener

נעביר את המידע שקיבלנו לפונקצית display שתדפיס לנו המידע שקיבלנו

נבצע בדיקת checksum לאיתור שגיאות בקבלת הפאקטות

**עץ החלטות של החלק השני**

יצירת RAW סוקט עם היעד ברשת לפי הIP

יצירת חיבור TCP סוקט עם הwatchdog

שליחת הודעת ICMP REQUEST לכתובת יעד בפונקצית ping

קבלת הודעת ICMP REPLY מכתובת היעד דרך RAW סוקט בפונקצית listener ונקבל הודעה מwatchdog בהתאם למצב הטיימר

נבצע בדיקת checksum לאיתור שגיאות בקבלת הפאקטות

נשלח הודעה שיש חיבור מוצלח עם היעד לWatchdog

נעביר את המידע שקיבלנו לפונקצית display שתדפיס לנו המידע שקיבלנו

ההודעה הבאה הגיעה בזמן תקין מתחת ל 10 שניות

שולח הודעה ל listener שהזמן עבר

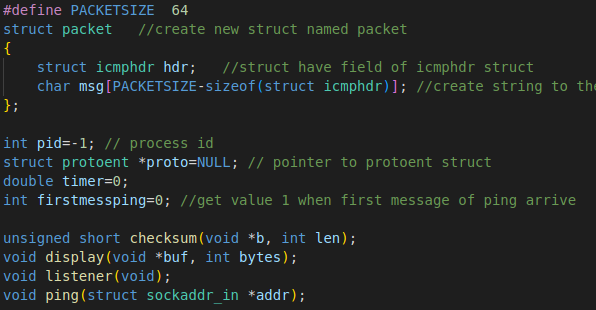
סוגר את הסוקט מול הלקוח ששולח לו הודעות על הפינג

עבר מעל 10 שניות מהודעה להודעה

נריץ אצל ה watchdog טיימר של 10 שניות שסופר אחורה ומתאפס כול פעם ל 10 שניות שמגיעה הודעה שחיבור הצליח מהפינג

**Ping.c**

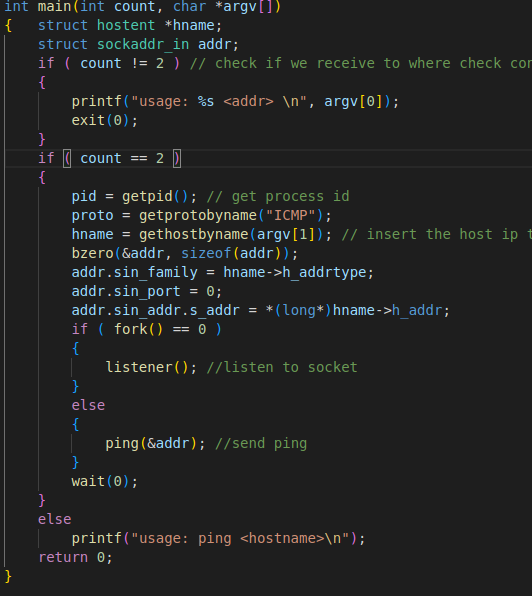
הקוד מבוסס על קוד שנלמד בתרגול .



תחילה הגדרנו את גדול הפאקטה להיות 64 , הגדרנו מבנה בשם packet שמכיל 2 שדות 1. מבנה בשם icmphdr שמכיל את הhader של icmp 2. סטרינג שיכיל את המידע שנרצה לשלוח.

לאחר מכן הצערנו על משתנים גלובליים pid שיכיל את מספר id של התהליך , מבנה protoent שמכיל מידע על פרוטוקולים ,טיימר לחישוב זמני ההגעה של הודעות ה פינג ומשתנה יגיד מתי קיבלנו את ההודעת הפינג הראשונה.

בנוסך יש את החתימות של הפונקציות שהשתמשנו בהם checksum, display, listener, ping .



בפונקציית הmain הגדרנו מצביע למבה hostent בשם hname שמכיל מידע על hostים ומבנה sockaddr\_in בשם addrשמכילה מידע על כתובת אינטרנט.

נבדוק קיבלנו מהמשתמש 2 קלטים שאחד מהם הוא שם הקובץ והשני או הכתובת IP שאליה נבצע את פינג, אם קיבלנו יותר או פחות קלטים נדפיס למשתמש הודעת שגיאה.

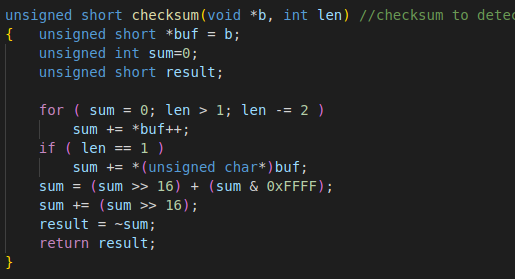
כעת לאחר שבדקו המשתמש הכניס את הכמות הקלטים הנכונה נשמור לתוך משנה pid שהגדרנו כמשתנה גלובליי את ה ID של התהליך שעליו רץ הmain נכניס לporto את שם הפרוטוקול איתו נעבוד (icmp) ול hname את כתובת הIP לביצוע הפינג.

נאפס את המנה addr ונכניס לתוכו את ה הIP שאליה נבצע פינג ,את הפורט וסוג הIP 4V או 6V.

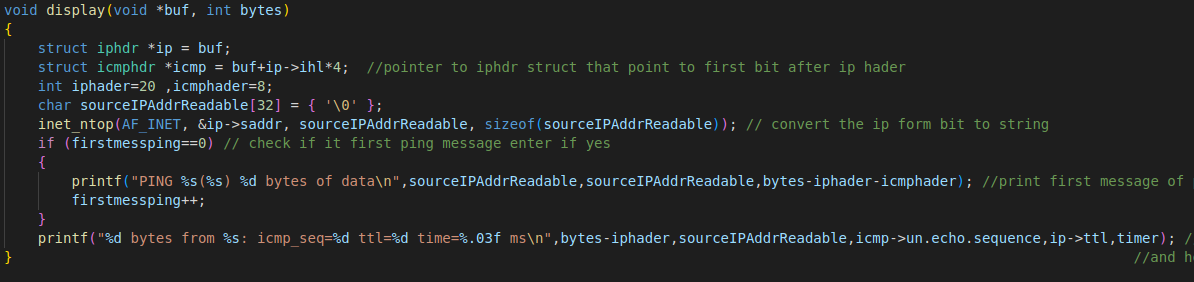
ניצור תהליך חדש באמצעות הפונקציה fork שמחזירה 0 אם שנוצר תהליך חדש.

אם fork תחזיר 0 נקרא לפונקציה listener שמקשיבה ומחכה לתגובות להודעות icmp שנשלחו.

אחרת נקרא לפונקציה ping תשלח לIP שהמשתמש נתן הודעות icmp לבדיקת תקינות החיבור.



פונקצית checksum זהו פונקצייה שבאמצעות נמצא שגיאות וביטים שהגיעו לא כמו שציפינו שיגעו ובכך נהייה בבקרה על המידע שמגיע.



פונקציית display זהו פונקציה שבאמצעות נדפיס את המידע שנרצה להציג למשתמש כחלק מפקודת הפינג שהוא ביקש.

מידע זה מגיע מהודעת ה icmp שמשם נשלוף את המידע.

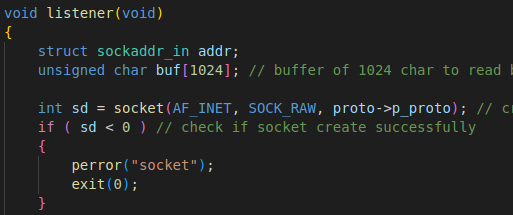
נגדיר 2 מצביעים 1 מסוג iphdr שיצביע על buf שקיבלנו ,ומצביע מסוג icmphdr שיצביע על hader של הודעת ה icmp .

נגדיר 2 משתנים שיכילו את כמות ביטים של הhader של פרוטוקול IP ושל הhader של הicmp.

ניצור סטרינג שיכיל את כתובת הIP של היעד ונכניס את הIP לתוכה באמצעות פונקציית inet\_ntop שממירה את הכתובת מביטים לסטרינג ושם בתוך המשתנה שהגדרנו.

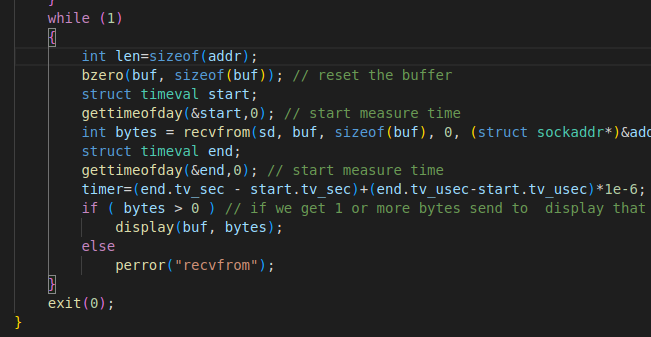
נבדוק שזה ההודעה הראשונה שקיבלנו באמצעות משתנה firstmessping בכדי להדפיס את הפרטים הראשונים על היעד ואז נעלה את משתנה ל1.

לאחר מכן נדפיס את כמות הביטים שקיבלנו ,מספר הפינג ,ה ttl שלו , ומשך הזמן שלקח לתשובה להגיע .



פונקצית listener זוהי הפונקציה שמחכה לקבל את התשובות להודעות icmp שנשלחות אל היעד.

תחילה נגדיר מבנה שיכיל את הכתובת הIP של היעד וbuffer בגדול 1024 ביטים שיכיל את המידע שיגיע.

ניצור סוקט מסוג raw סוקט לחיבור עם היעד באמצעות פונקצית sock שמחזירה ערך שלם , אם חזר ערך קטן מ 0 או ש0 סימן שהייתה בעיה עם יצירת הסוקט לכן נדפיס הודעת שגיאה.

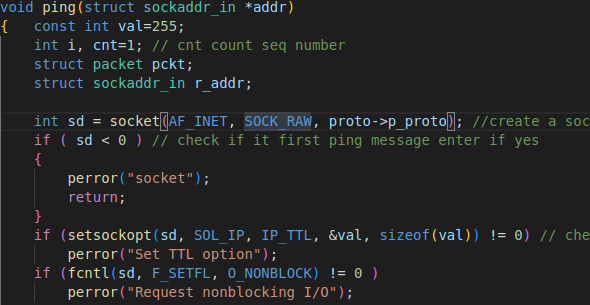
לאחר מכן נרוץ בלולאה אינסופית כיון שלא נגביל את כמות הפעמים של ביצוע פקודת ה פינג, נגיר משתנה שיכיל את גודל המבנה addr שמכיל את הכתובת יעד ונאפס את הbuffer .

ניצור משתנה מסוג timeval שישמש למדידת הזמן השלקח התשובה להגיע.

נמדוד את הזמן העכשיו באמצעות פקודת gettimeofday , נקבל את מידע מהסוקט באמצעות הפקודה recvfrom שכותבת את המידע שהתקבל לbuffer , ונמדוד שוב את הזמן כדי לחשב את משך הזמן שעבר.

נבצע החסרה של הזמן סיום מהזמן ההתחלה ונכניס הזמן למשתנה timer .

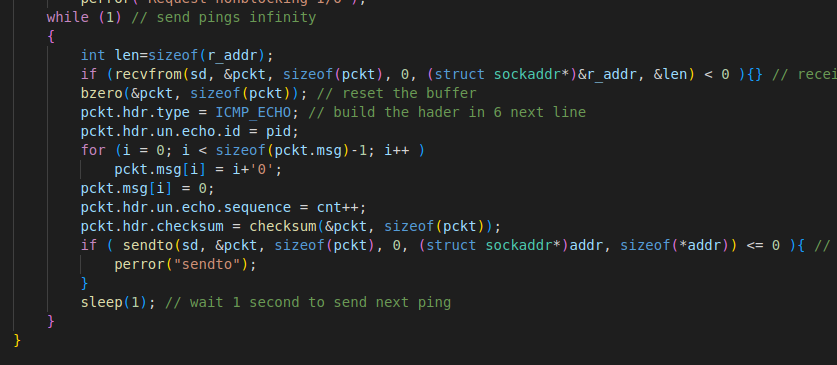
ולבסוף נבדוק אם התקבל מידע ואם כן נשלח אותו לdisplay להדפסה ואם לא קיבלנו מידע נדפיס הודעת שגיאה בהתאם.



פונקצית ping זוהי הפונקציה ששולח את ההודעות ה icmp ליעד לבדיקת תקינות החיבור.

נגדיר מספר משתנים cnt לספירת מספר ההודעות שנשלחו , פאקטה באמצעות מבנה pckt וכתובת IP של היעד באמצעות המבנה sockaddr\_in .

ניצור סוקט מסוג raw סוקט באמצעות פונקצית sock שמחזירה ערך שלם , אם חזר ערך קטן מ 0 או ש 0 סימן שהייתה בעיה עם יצירת הסוקט לכן נדפיס הודעת שגיאה.

נבדוק שהסוג פנוי באמצעות הפונקציה setsockop שמחזירה ערך שלם אם חזר ערך שונה מ 0 סימן שהיתה שגיאה ולכן נדפיס הודעה בהתאם.

לאחר מכן נרוץ בלולאה אינסופית כיון שלא נגביל את כמות הפעמים של ביצוע פקודת ה פינג,

נשמור במשנה את גודל המבנה שמכיל את הכותב של היעד, נאפס את הפאקטה באמצעות bzero .

ונכניס נתונים לפאקטה icmp\_echo שזה שלחת פינג pid זה ID של התהליך .

ואז נכניס את הנתונים של הdata של הודעת ה icmp בלולאה ,מכניס את מספר הפינג שנשלח (cnt ) ונאפס את הchecksum .

נשלח את הבקשה לכתובת היעד באמצעות פונקצית sendto שמחזירה ערך שלם אם חזר ערך קטן או שווה ל 0 סימן שהייתה שגיאה לכן נדפיס הודעת שגיאה בהתאם.

ולבסוף באמצעות פנקציית sleep נמתין בשליחה לשליחה שניה אחת כדי לא להציף את היעד בהודעות icmp .