



תוכנה ראשונה ב - LEX במערכת הפעלה LINUX

PuTTY →



• כניסה למערכת Linux

10.200.0.17

Login: usrName

Password: usrPassword

- על מנת לכתוב את הטקסטים השונים של תוכנה ב-LINUX משתמשים הפקודה **nano**.

בזמן עריכה, נוכל לשמור את הקובץ הנוכחי על ידי לחיצה על **Ctrl+X** ובחירת שם הקובץ בו נרצה לשמור את המסמך (או **Enter** על מנת לדרוס את המסמך הקיים).

צירוף המקשים Ctrl+X ישמר אותנו על מנת לצאת מ-nano.

במידה ולא שמרנו את המסמך הפעיל תינתן לנו אפשרות לעשות זאת.

על מנת לערוך קבצים נכתוב בשורה הפקודה:

nano my_first_file.l או **nano my_first_file.lex**

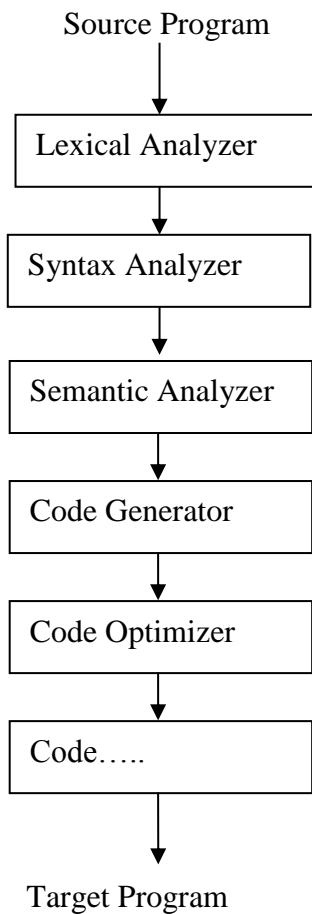
- כדי להציג את רשימת קבצים בספרייה הנוכחית נשמך בפקודה: **ls**.
- כדי להציג את רשימת קבצים בספרייה מסוימת נשמך בפקודה: **ls name_directory**.
- ליצירת ספרייה חדשה בתוך ספרייה נוכחית (תת-ספרייה) נשמך: **mkdir name_dir**.
- להחלפת ספרייה נוכחית נשמך: **cd /usr/sbin**.

חשוב לזכור כי אחרי שם פקודה (**cd**) חייבים לשים רווח לפני שם ספרייה, אליה רוצים לעבור.

שימוש בפקודה ללא פרמטרים תעשה העברה לספרייה ראשית של משתמש: **cd**

חזרה לספריית אם (צעד אחד אחורה): **cd ..**

כדי להבין את נושאי המעבדות בקורס זה, נדון קצת בשלבי הקומפילציה (compilation phases):



שלושת השלבים הראשונים מוכרים לנו משפת הדיבור. אנו מזהים במשפט מלים בודדות שהן אכן מלים בשפה שלנו. אנו אח"כ מבצעים ניתוח דקדוקי-תחבירי של המשפט ולבסוף מחפשים את הסמנטיקה של המשפט. כך גם עובד המהדר. השלבים הבאים רלוונטיים כמובן רק לשפות תכנות ולמהדר ולא לנו בני האדם.

יצירת מנתח לקסיקלי בעזרת Lex

מבנה תוכנית

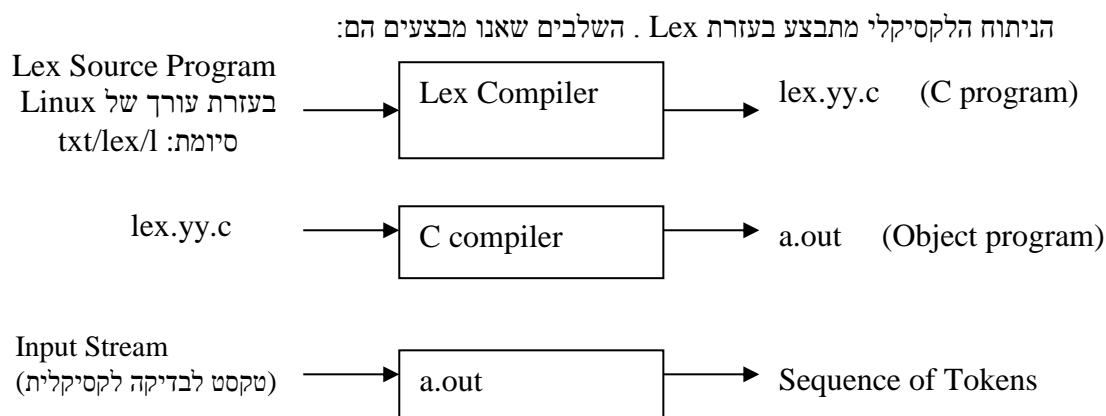
זהו מבנה של תוכנית ב-Lex:

```

{ definitions }
%%
{ rules }
%%
{ user subroutines }
  
```

כפי שאנו רואים תוכנית Lex מורכבת מ-3 חלקים: הגדרות, כללים ופרוצדורות. הסימן %% מפריד בין החלקים. החלק היחיד שהוא הכרחי הוא החלק של הכללים. בהגדרות אנו נצהיר על משתנים וקבועים שנשתמש בהם בכללים, אם יש כאלה משתנים וקבועים. בפרוצדורות נגדיר פרוצדורות בהן אנו צריכים להשתמש בכללים, אם יש כאלה פרוצדורות.

שילבי הרצת תוכנית Lex



לאחר שנכנסו לשרת ה-Linux:

1. ניצור קובץ עם סיומת txt (למשל בעזרת nano). נבצע לו קומפילציה של lex על ידי הקשה של שם קובץ: **lex my_first_lex.lex**
 2. כעת נבצע קומפילציה של C לקובץ lex.yy.c שנוצר מהשלב הקודם על ידי הקשה של **lex.yy.c -o Filename -ll .cc** הקומפיילר של C ייצור קובץ בשם Filename עם סיומת .out
 3. נבנה קובץ של טקסט לניתוח לקסיקלי בעזרת עורך. כעת נריץ את הקובץ עם סיומת out שקיבלנו ונשלח אליו את קובץ הטקסט לניתוח לקסיקלי, כלומר נקיש: **./a.out**
- המחרוזת **yytext** היא מחרוזת מיוחדת המכילה את המחרוזת שזוהתה בביטוי הרגולרי וניתן להשתמש בה בקטע התוכנית עבור הביטוי הרגולרי.
 - המשתנה המספרי **yyleng** מכיל תמיד את אורך **yytext**.
 - נשים לב כי תוכן **yytext** ו-**yyleng** משתנה אוטומטית במהלך הניתוח הלקסיקלי.