

עבודה בסיסית בלינוקס

אחת המטרות של הקורס הזה היא לאמן אתכם בשימוש במערכת יוניקס - מערכת נפוצה ביותר בעולם התיכנות. אנחנו לא נלמד את הנושא לעומק, נלמד רק כמה פקודות בסיסיות שיאפשרו לכם לבנות ולהריץ תוכנות ב-C++.

לצורך הקורס נשתמש בלינוקס, ובפרט במערכת אובונטו. המערכת אמורה להיות מותקנת על המחשבים במעבדות, אתם יכולים גם להתקין אותה על המחשב הביתי שלכם במקביל ל"חלונות" (יש מדריכים באינטרנט איך לעשות את זה). אחרי שפתחתם את אובונטו, פיתחו חלון-מסוף (terminal) ונסו את הפקודות הבאות:

- `ls` - הצגת הקבצים בתיקה הנוכחית.
- `ls -latr` - וסידור הקבצים לפי זמן העידכון - ייש מדריכים האחרון שלהם (לרוב הפקודות ביוניקס אפשר להוסיף פרמטרים ע"י סימן מינוס ואחריו אות אחת או יותר).
- `cd [dirname]` - הליכה לתיקה אחרת.
- `mkdir [dirname]` - יצירת תיקיה חדשה.
- `cat [filename]` - הדפסת תוכן הקובץ.
- `nano [filename]` - עריכת הקובץ.
- `grep [string]` - חיפוש מחרוזת בקלט התקני (למה זה טוב? נראה בהמשך).
- `source [filename]` - הרצת פקודות מתוך קובץ.
- `top` - צפיה בתהליכים רצים.
- `kill` - הריגת תהליך לפי מספר.
- `killall` - הריגת תהליכים לפי שם.

במערכות יוניקס יש מספר קטן של פקודות בסיסיות, שאפשר לשלב ביניהן על-ידי אופרטורי הכוונת קלט ופלט. למשל:

- העברת קובץ כקלט לפקודה - אופרטור הכוונת הקלט (input redirection):
 - `grep [string] < input.txt`
- שמירת התוצאות של פקודה לתוך קובץ - אופרטור הכוונת הפלט (output redirection):
 - `ls -latr > output.txt`
- העברת הפלט של פקודה אחת לפקודה אחרת - אופרטור הצינור (pipe):
 - `ls -latr | grep bash`

התקנת חבילות

במערכות לינוקס מודרניות ישנן פקודות פשוטות המאפשרות להתקין חבילות חדשות. הפקודות משתנות מהפצה להפצה. בלינוקס, הפקודה המשמשת להתקנה היא `apt`. כדי להשתמש בפקודה הזאת, צריך לקבל "הרשאות משתמש-על" (`super-user`). איך עושים את זה? שתי דרכים:

- כדי להפוך ל"משתמש על", כותבים את הפקודה `su` ומכניסים סיסמה. כדי לצאת ממצב "משתמש על", כותבים `exit`.
- כדי להריץ פקודה אחת כ"משתמש על", כותבים לפני הפקודה `sudo`. זו השיטה המקובלת.

הנה דוגמה לפקודה שמתקינה משחק נחמד ושימושי לשיפור מהירות ההקלדה בשפות שונות:

```
sudo apt install typespeed
```

(למי שאוהב משחקי טקסט: ראו משחקים נוספים כאן
(<https://askubuntu.com/q/11485/42073>).

בניית תוכנת C++ על לינוקס

ישנם כמה חבילות-קומפילציה מקובלות לשפת C++. החבילה הנפוצה ביותר היא כנראה gcc - gnu compiler collection, חבילה הכוללת כמה קומפיילרים שהעיקרי בהם הוא של C, C++. אנחנו נשתמש בחבילה אחרת בשם clang, ובפרט בגירסה 5.0. היתרון העיקרי של חבילה זו הוא שהודעות-השגיאה שלה קלות יותר להבנה - יתרון חשוב מאד כשלומדים שפה חדשה. כדי להתקין אותה על אובונטו, נכתוב:

```
sudo apt install clang++-5.0
```

עכשיו אנחנו יכולים לבנות תוכנות ב-C++. למשל, בתיקיה 1, נכתוב:

```
clang++-5.0 hello.cpp  
ls -latr
```

ונראה קובץ חדש בשם a.out. זה קובץ שאפשר להריץ. כדי להריץ אותו נכתוב:

```
./a.out
```

בניית תוכנה ב-C++ (כמו בסי) מורכבת מארבעה שלבים, כדאי להכיר אותם כדי להבין טוב יותר את הודעות השגיאה שאנחנו מקבלים:

- עיבוד מקדים - preprocess - ביצוע פקודות המתחילות ב-#, כגון הכללת קבצים (#include), הגדרת קבועים (define#) וכו'. ראו דוגמה בתיקיה 3.
- הפיכה לאסמבלר - assemble.
- הפיכה לשפת מכונה - compile.
- קישור כל הקבצים בשפת מכונה עם ספריות חיצוניות (במקרה הצורך) והפיכה לקובץ ריצה - link.

ניתן לבצע כל אחד בנפרד ע"פ העברת הפרמטר המתאים ל-clang, אבל בדרך-כלל מריצים את כולם יחד.

אנחנו נצטרך חבילות נוספות לבניית פרויקטים ובקרת גירסאות:

```
sudo apt install build-essential  
sudo apt install git
```

סיכום: אראל סגל-הלוי.