

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 במערכת-ההפעלה לינוקס, ובפרט בגירסה שלה שנקראת אובונטו (Ubuntu).

התקנת לינוקס על המחשב שלכם

- יש כמה דרכים להתקין אותה על המחשב שלכם - יש באינטרנט מדריכים לגבי כל אחת מהאפשרויות:
1. התקנה על מחשב חדש כמערכת-הפעלה יחידה;
 2. התקנה על מחשב שכבר מותקנת עליו מערכת חלונות, בתצורת dual-boot;
 3. התקנת תת-מערכת בתוך חלונות - Windows Subsystem Linux. ניתן למצוא בחנות האפליקציות של חלונות החל מגרסה 10 של חלונות. בדקתי אותה והיא מאד נוחה.
 4. עבודה בסביבת-פיתוח אונליין, ללא כל התקנה על המחשב שלכם, למשל באתר <http://c9.io>.

אם לא הצלחתם להתקין על המחשב שלכם, אתם יכולים לעבוד במעבדות-המחשבים שבאוניברסיטה; המערכת אמורה להיות מותקנת שם.

פקודות בסיסיות

אחרי שפתחתם את אובונטו, פיתחו חלון-מסוף (terminal) ונסו את הפקודות הבאות:

- `ls` - הצגת הקבצים בתיקה הנוכחית.
- `ls -latr` - אותו דבר, עם יותר פירוט לגבי כל קובץ, וסידור הקבצים לפי זמן העידכון - `ls -latr` - 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, ++C. אנחנו נשתמש בחבילה אחרת בשם clang, ובפרט בגירסה 5. 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, בתיקה 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 בתיקה 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.

לצורך בניית פרויקטים נצטרך גם את החבילה הבאה:

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

בקרת גירסאות – גיט

ניתן לעבוד עם גיט (git) ישירות משורת-הפקודה של אובונטו. קודם-כל צריך להתקין:

```
sudo apt install git
```

אם התקנתם אובונטו ישירות על המחשב שלכם (לא דרך Windows Subsystem Linux), אתם יכולים להתקין גם ממשק-משתמש גרפי:

```
sudo apt install git-gui
```

כדי לשבט מאגר (למשל את המאגר של הקורס) יש לכתוב:

```
git clone https://github.com/erelsgl/ariel-cpp-5779.git
```

כדי להיכנס לתיקיה:

```
cd ariel-cpp-5779
```

אם התקנתם git-gui, אתם יכולים לכתוב בתוך התיקיה:

```
git gui
```

ומכאן והלאה לעבוד עם ממשק גרפי.

אם לא התקנתם git-gui, אתם יכולים לעבוד ישירות מהמסוף:

כדי למשוך עדכונים מהשרת:

```
git pull
```

כדי להוסיף קובץ שעדכנתם לרשימת הקבצים המוגשים:

```
git add <filename>
```

כדי להגיש את השינויים שלכם:

```
git commit
```

כדי לדחוף את השינויים שהגשתם לשרת:

```
git push
```

סיכום: אראל סגל-הלוי.