

מעבדה 2 – קומפילציה ושימוש מתקדם ב Unix

דוגמא 1: כתיבת קובץ פקודות – shell script

בכל עורך הטקסטים, למשל, בעזרת פקודת pluma נפתח קובץ ונרשום בו את הפקודות של LINUX

```
clear
echo "the script starts now!"
echo "Hi, $USER!"
echo
echo "You can see the file's list of your directory now."
ls -l
echo
echo "I'm going to delete your directory now."
rm my
echo
echo "I'm giving you back your prompt now"
echo
```

שומרים את הקובץ בשם filename.sh, למשל myshell.sh והופכים אותו לביצועי:

`chmod +x myshell.sh`

כדי להריץ את קובץ הפקודות, רושמים :

`sh myshell.sh`

או

`./myshell.sh`

דוגמא 2: כתיבת קובץ פקודות – shell script עם פרמטרים

```
# example of using arguments to a script
echo "My first name is $1"
echo "My surname is $2"
echo "Total number of arguments is $#"
```

נשמור את הנ"ל בקובץ name.sh ונריץ את הקובץ:

`chmod +x name.sh`

`./name.sh Israel Israeli`

ניתוב

הפניית פלט לקובץ (במקום למסך) נעשה ע"י האופרטורים > או >>. האופרטור > גורם לפלט להישמר בתוך קובץ חדש בשם הנתון. כלומר אם היה קובץ בשם הנתון, התוכן שלו יאבד. האופרטור >> גורם לפלט להתווסף לתוכן של קובץ במידה והוא קיים כבר. ואם אין קובץ בשם הנתון, ייוצר קובץ חדש.

למשל `ls -l > filelist.txt`

יוצר קובץ בשם filelist.txt שיכיל את התשובה שהפקודה `ls -l` בדרך כלל שופכת על המסך (רשימת הקבצים בספרייה הנוכחית כולל חלק מהמאפיינים). אם היה כבר קובץ בשם filelist, התוכן הקודם שלו יאבד. לאומת זאת במקרה של הפקודה `ls -l >> filelist.txt`

אם filelist עדיין לא קיים, לא יהיה הבדל בין הפקודה הזו לקודמת, אולם אם קיים קובץ בשם זה, הפלט של הפקודה יתווסף לסוף הקובץ הקיים במקום לדרוס אותו.

קבלת קלט מקובץ במקום מהמקלדת נעשה ע"י האופרטור < בצורה דומה לאופרטור הפלט. לדוגמא, ניצור בספריית Documents קובץ טקסט חדש:

pluma mypath.txt

home/braude/ ונרשום בו את המסלול

הפקודה הבאה: **ls < mypath.txt** תציג על המסך את הספריות את הקבצים שמצאים ב-braude. פקודת ls קיבלה את הפרמטר שלה מתוך הקובץ.

פרמטרים ל-main, קלט ופלט סטנדרטיים

מתכנתים בשפת C יתקלו במוקדם או במאוחר במושגים standard input(stdin), standard output(stdout) ו-standard error(stderr). אלה הם מקור הקלט הסטנדרטי (המקלדת היא ברירת המחדל), יעד הפלט הסטנדרטי ויעד הודעות השגיאה (המסך כברירת המחדל לשניהם). מקור הקלט הסטנדרטי הוא המקור לפקודות scanf, getchar, gets ואחרים ויעד הפלט הסטנדרטי הוא היעד של פקודות printf, putchar, puts ואחרים. ליעד הודעות השגיאה חייבים לציין אותו. למשל,

```
fprintf(stderr )
```

אבל גם לאחרים אפשר לעשות זאת. למשל,

```
fscanf(stdin), fprintf(stdout)
```

אפשר לנתב את הערוצים הללו מחדש לברירות המחדל שלהם (למשל ע"י הפקודות freopen, dup, close). מה שהאופרטורים <, >, >> ו-| עושים, זה למעשה, ניתוב הערוצים הסטנדרטיים הללו ליעדים חדשים, למשל, לקבצים.

דוגמא 3: תכנית עם main שמקבלת פרמטרים.

התכנית מקבלת 3 פרמטרים: שם התכנית, שם קובץ קלט ושם קובץ פלט. תכינו קובץ קלט עם סיומת txt. (רשמו טקסט כלשהו בקובץ זה). אין צורך להכין קובץ הפלט. תריצו את התכנית ב-LINUX ונסו אותה במצבים שונים:

(א) הרצה ללא שמות הקבצים,

(ב) הרצה עם שם קובץ הקלט בלבד

(ג) הרצה עם שמות של שני הקבצים.

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#define MAXLINE 100
```

```
void Error(char* msg){fprintf(stderr, "Error:%s\n", msg); exit(1);}
```

```
int main(int argc, char* argv[]){
```

```
    FILE *in,*out;
```

```
    char buff[MAXLINE];
```

```
    if(argc>3) Error("Bad number of parameters\n");
```

```
    if(argc==3){
```

```
        if((out=fopen(argv[2],"w"))==NULL) Error("Can't open output file\n");
```

```
    }
```

```
    else
```

```
        out=stdout;
```

```
    if(argc>1){
```

```
        if((in=fopen(argv[1],"r"))==NULL) Error("Can't open input file\n");
```

```
    }
```

```

else
    in=stdin;
while(fgets(buff, MAXLINE, in)!=NULL)
    fputs(buff, out);
if(out!=stdout)
    fclose(out);
if(in!=stdin)
    fclose(in);
return 0;
}

```

תרגילים:

תרגיל 1:

עליך לכתוב shell script אשר מקבל כפרמטרים שם של תיקייה (התיקייה לא קיימת) ושם של קובץ (קיים). הסקריפט יוצר תיקייה בשם שהתקבל ויוצר קובץ חדש בשם lab2File.txt ומעתיק את תוכנו של הקובץ שהתקבל לקובץ החדש, לאחר מכן יש להוריד את הרשאות הכתיבה לקובץ lab2File.txt. הסקריפט נדרש להדפיס הודעות למסך בהתחלה ובסוף.

תרגיל 2:

שלב 1:

כתבו תכנית בשפת C המקבלת רשימת מספרים שלמים ומדפיסה את הממוצע שלהם. שמרו את התכנית שלכם בקובץ avgNum.c וקמפלו את התכנית בעזרת:

```
cc avgNum.c -o avgNum
```

התוכנית תקבל את הקלט **משורת הפקודה ולא** ע"י scanf. הריצו את התכנית ובדקו תקינות לגבי קלטים שונים.

לדוגמא, בהרצה: avgNum 1 2 3 יודפס למסך: 2. בהרצה: avgNum 50 10 20 90 יודפס למסך: 42.5. בהרצה: avgNum 10 10 2 3 יודפס למסך: 6.25. * יש לדאוג לשימוש ב exit 1 אם יש בעיה...

שלב 2:

כתבו shell script (קובץ עם סיומת .sh) המקמפל ומריץ את הקוד שכתבתם. הסקריפט יריץ את הקוד עם ארבעה מספרים: 1,2,3,4.

שלב 3:

הריצו את ה shell script שכתבתם.

תרגיל 3:

שלב 1: כתבו תכנית בשפת C בשם catByAt.c המקבלת מחרוזת ללא רווחים **משורת הפקודה**. התוכנית תדפיס למסך את תתי-המחרוזות במחרוזת הנתונה המופרדות על ידי @ באותיות גדולות. לדוגמא, בהרצה: catByAt Israel1990@gmail@com יודפס למסך:

ISRAEL1990

EMAIL

COM

ניתן להשתמש בפונקציית מהסיפרייה string.h.
קמפלו את התכנית בעזרת:

```
cc catByAt.c -o catByAt
```

התוכנית תקבל את הקלט **משורת הפקודה ולא** ע"י scanf.
הריצו את התכנית ובדקו תקינות לגבי קלטים שונים.
** יש לדאוג לשימוש ב exit 1 אם יש בעיה...

שלב 2:

כתבו shell script (קובץ עם סיומת .sh) המקמפל ומריץ את הקוד שכתבתם. הסקריפט יריץ את הקוד עם מחרוזת קלט משורת הפקודה

שלב 3:

הריצו את ה shell script שכתבתם.

הוראות הגשה:

יש להגיש קובץ WORD עם שמות שני בני הזוג ופתרון 3 התרגילים הנ"ל.
לכל תרגיל, יש להעתיק (לא לצלם) את הקוד הנדרש. יש לכתוב קוד מסודר ומתועד.
בשאלות 2,3 יש להוסיף תצלום מסך ההרצה (בטרמינל בסנטוס) עם קלטים כרצונכם.
ההגשה בזוגות. שני בני הזוג נדרשים להגיש.

בהצלחה!!!