

הוספת תכונות לתכנית shell

- בתיקיה "קטעי קוד" במודל ישנם שלושה קבצים לתכנית shell:
- shell1.c - מדפיס סמן ומריץ פקודות עם ארגומנטים.
 - shell2.c - מוסיף ניתוב לקובץ.
 - shell3.c - מוסיף pipe.

נקמפל את התכנית השנייה:

```
gcc -o myshell shell2.c
```

נריץ את התכנית:

```
./myshell
```

נוכל לראות שהתכנית מבצעת פקודות:

```
hello: ls -l
```

וגם מבצעת פקודות ברקע:

```
hello: ls -l &
```

וגם מנתבת פלט לקובץ:

```
hello: ls -l > file
```

נשים לב שחלקי הפקודה מופרדים על ידי התוו רווח.

עליכם להוסיף את התכונות הבאות (אפשר להוסיף פונקציות ל-main):

1. ניתוב כתיבה ל- stderr

```
hello: ls -l nofile 2> mylog
```

הוספה לקובץ קיים ע"י >>

```
hello: ls -l >> mylog
```

כמו בתכנית shell רגילה, אם הקובץ לא קיים, ייצור אותו.

פקודות מובנות ב-shell (מקומם לפני fork()) ויש לבצע אחריהם :continue):

2. פקודה לשינוי הסמן:

```
hello: prompt = myprompt
```

(הפקודה מכילה שלוש מלים שמופרדות בשני רווחים)

3. פקודת echo שמדפיסה את הארגומנטים:

```
hello: echo abc xyz
```

ידפיס

```
abc xyz
```

הפקודה תתמוך בניתוב הפלט (>, >>)

4. הפקודה

```
hello: echo $?
```

תדפיס את הסטטוס של הפקודה האחרונה שהתבצעה.

5. פקודה שמשנה את תיקיית העבודה הנוכחית של ה-shell:

```
hello: cd mydir
```

6. פקודה שחוזרת על הפקודה האחרונה:

```
hello: !!
```

(שני סימני קריאה במלה הראשונה של הפקודה)

7. פקודה ליציאה מה-shell:

```
hello: quit
```

8. אם המשתמש הקליד Control-C, התכנית לא תסיים אלא תדפיס את ההודעה:
You typed Control-C!

במידה וה-SHELL מריץ תהליך אחר, התהליך יזרק ע"י המערכת
(default behavior)

9. אפשרות לשרשר כמה פקודות ב-pipe.

עבור כל פקודה ב-pipe יש צורך בהקצאה דינמית של argv

10. הוספת משתנים ל-shell:

```
hello: $person = David
```

```
hello: echo person
```

```
person
```

```
hello: echo $person
```

```
David
```

11. פקודת read:

```
hello: echo Enter a string
read name
hello
echo $name
hello
```

12. תמיכה בבקרת זרימה, כלומר IF/ELSE. לדוגמה:

```
if date | grep Fri
then
    echo "Shabat Shalom"
else
    echo "Hard way to go"
fi
```

הקלדת ה תנאי תתבצע שורה שורה, כמו שמופיע כאן.

לאחר שהוספתם את התכונות, נא הריצו את הפקודות הבאות:

```
./shell
hello: date >> myfile
hello: cat myfile
hello: date -u >> myfile
hello: cat myfile
hello: wc -l < myfile
hello: prompt = hi:
hi: mkdir mydir
hi: cd mydir
hi: pwd
hi: touch file1 file2 file3
hi: ls
hi: !!
hi: echo abc xyz
hi: ls
hi: echo $?
hi: ls no_such_file
hi: echo $?
hi: ls no_such_file 2> file
hi: Control-C
hi: cat > colors.txt
blue
black
```

```

red
red
green
blue
green
red
red
blue
Control-D
hi: cat colors.txt
hi: cat colors.txt | cat | cat | cat
hi: sort colors.txt | uniq -c | sort -r | head -3
hi: quit

```

הערות טכניות:

כל הסעיפים הם חובה לביצועה, אך ללא סעיפים 9 ו 12 המטלה לא תתקבל (כלומר ציון 0), הורדת נקודות על כל סעיף אחר היא - 10 נקודות. כמו כן חובה להגיש MAKE תקין (אחרת, הציון הוא 0)

המטלה תכתב ב ספת ++C/C

דוגמת ההרצה היא לשם דוגמה בלבד. בבדיקה תתכן הרצה של פקודות נוספות, בסדר שונה, וגם כאלה שחוזרות על עצמן. יתרה על כך, עליכם **לצרף קובץ טקסט**, המכיל דוגמאות מהבדיקות שלכם.

המטלה עלולה להיבדק באופן אוטומאטי, ולכן יש **להקפיד על ה סינטקס**. קובץ ההרצה יקרה myshell

יש להגיש את המטלה כולל הנספחים (make, readme) בקובץ ZIP ששמו יהיה הת.ז. של המגיש. אם ישנם שני מגישים אזי הקובץ יקרה תז_תז. (15 נקודות אם לא).

איחור בהגשת המטלה ללא סיבה מוצדקת (מילואים/מחלה) מוריד 10 נקודות ליום. במידה ולאחד משני השוטפים יש סיבה מוצדקת, ולשני לא, ההורדה תיהי 5 נקודות ליום.

יש להיזהר בזמן העברת הקבצים בין מערכות הפעלה שונות (וחלונות במיוחד), יש לוודה שלקווצים שמות תקינים. ללא .txt , גרשיים, או כל ארתיפקט אחר. במידה והוגשו הקבצים בשמות שגויים, ההורדה היא 20 נקודות.