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

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

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

1. פקודה לשינוי הסמן:

hello: prompt = myprompt

(הפקודה מכילה שלוש מלים שמופרדות בשני רווחים)

1. פקודת echo שמדפיסה את הארגומנטים:

hello: echo abc xyz

ידפיס

abc xyz

1. הפקודה

hello: echo $?

תדפיס את הסטטוס של הפקודה האחרונה שהתבצעה.

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

hello: cd mydir

1. \*פקודה שחוזרת על הפקודה האחרונה:

hello: !!

(שני סימני קריאה במלה הראשונה של הפקודה)

1. פקודה ליציאה מה- shell:

hello: quit

1. אם המשתמש הקליד Control-C, התכנית לא תסיים אלא תדפיס את ההודעה:

You typed Control-C!

1. \*אפשרות לשרשר כמה פקודות ב- pipe.

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

(או להקצות מערך בגודל 10).

1. הוספת משתנים ל- shell:

hello: $person = David

hello: echo person

person

hello: echo $person

David

1. פקודת read:

hello: echo Enter a string

read name

hello

echo $name

hello

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

Technical Notes:

* Submission day – up to 08/07/22 11:59pm
* Put your results in ZIP (not rar/7z/tar etc.) and name it by your ID. (-15 points if not)
* You can sibmit in pairs, name id ID\_ID (-15 points if not)
* Failing to submit working Make may lower your point dramaticaly, down to Zero !
* If you are late, you can submit up to 10/07/22 11:59pm. 10 points loose for each day.
* Be aware of linux/windows files transfer. Linux is CaseSensitive, while Windows is not
* Don’t wait for the last minute, as “My VM is died”, “My cat swallowed the mouse” and other subterfuges will not be accepted
* The task should be implementing using C language
* Beware of wrong input !
* Note that the script for the test may be in different order, and may use general Linux commands, like ECHO