

# מערכות הפעלה - תרגיל בית 0

אוניברסיטת חיפה

סמסטר אביב תשפ"ד

## התקנת מערכת ההפעלה והמכונה הוירטואלית

עקבו אחר השלבים הבאים:

1. Download VMware Player from [here](#).
2. Install it on your machine.
3. Download the zip file of the Red Hat Linux **Linux-cs234123** from [here](#).
4. Unzip the file to some folder (named **Linux-cs**, for example).
5. Open the VMware Player.
6. Go to Player→File→Open, browse to the **Linux-cs** folder and select **Linuxcs234123.vmx**.
7. Double click on the newly added icon in the VMware Player to start the VM.

⊕ While starting the VM, always select Red Hat **Custom** version.

⊕ Login: username=root, password=234120.

## תיאור התרגיל

בתרגיל זה עליכם לכתוב תוכנית המממשת מחשבון פשוט הכולל ארבע פעולות חשבון (+, -, ×, ÷) על מספרים טבעיים (integers) בעלי ספרה אחת בלבד.

⊕ המחשבון יקבל קלט מה-input stream (יש להשתמש בפונקציה fgets).

⊕ אם הרישא של הקלט היא מהצורה

whitespace a whitespace op whitespace b

אז התוכנית תדפיס ל-output stream את תוצאת החישוב (כאשר  $a, b$  הם מספרים כמתואר לעיל,  $op \in \{+, -, *, /\}$  ו-whitespace מייצג מספר אי-שלילי של רווחים), דוגמאות בהמשך.

⊕ כל עוד קיימת לקלט רישא המייצגת ביטוי תקין, הסיפא המתאימה יכולה להיות כל מחרוזת, ותוצאת הקלט צריכה להיות מחושבת ע"פ הרישא המתאימה.

⊗ למשל, הקלט הבא הינו תקין, והערך המודפס צריך להיות 11.

5	+	6	something	1	-	5
---	---	---	-----------	---	---	---

⊕ התוכנית תסיים את ריצתה כאשר תתקבל המחרוזת "exit".

⊕ התוכנית תדפיס ל-output stream את המחרוזת "error" כאשר הקלט אינו תואם את התבנית המתוארת לעיל, או כאשר יש ניסיון לחלוקה ב-0.

## דוגמאות

מספר דוגמאות בטבלה 1.

קלט	פלט
0 + 0	0
1 - 8	-7
9 * 2	18
3/9	0
5 / 0	error
asdf 3 + 2	error
4 - 4 234s	0
2 * 23	4
6 + 4 + 3 + 2	10
8 * * 9	error
6 + 6	12

טבלה 1: דוגמאות קלט-פלט.

## קומפילציה והרצה

כתבו את התוכנית באמצעות עורך הטקסט המועדף עליכם. כעת, כדי להדר ולהריץ את התוכנית, בצעו את השלבים הבאים:

1. העתיקו את התוכנית שכתבתם אל המכונה הוירטואלית. בכדי לעשות זאת, עליכם ליצור Shared Folder שתאפשר לכם לגשת לקבצי העבודה גם מה-VM. היעזרו בסרטון [זה](#) כדי לעשות זאת.

2. הדרו את התוכנית שכתבתם - על שם הקובץ להיות calc.c, הדרו אותו באמצעות הרצת הפקודה הבאה ב-terminal בתוך ה-VM:

```
gcc -Werror -std=c99 calc.c -o calc
```

הסבר על הפקודה:

⊕ gcc: המהדר של C, built in בלינוקס.

⊕ Werror:- דגל שגורם לקומפיילר להתייחס לכל warning כשגיאה.

⊕ -std=c99: דגל שמציין קומפילציה לפי סטנדרט C99.

⊕ calc.c: שם קובץ המקור שמקומפל.

⊕ calc -o: לאחר הקומפילציה, ה-ELF שיווצר ייקרא calc.

3. הריצו את התוכנית באמצעות הפקודה הבאה ב-terminal של ה-VM:

```
./calc
```

לנוחיותכם, הדגמה של שלבים 2 ו-3 נמצאת בלינק [זה](#).

## בדיקה

בדקו את תקינות התוכנית שלכם ע"י קבצי קלט ופלט ידועים מראש. סופקו לכם שני זוגות של קבצי קלט ופלט לדוגמא (`test{1,2}.in, out`). בדקו את תקינות התוכנית שלכם ע"י הכוונת ערוצי הקלט/פלט כך שיעשו שימוש בקבצי הבדיקה:

1. השתמשו ב-`test1.in` כקלט לתוכנית, ושמרו את הפלט בקובץ `mytest1.out` באמצעות הפקודה הבאה:

```
./calc < test1.in > mytest1.out
```

2. השוו בין הפלט שלכם לפלט שסופק לכם באמצעות הפקודה הבאה:

```
diff test1.out mytest1.out
```

3. אם הפקודה רצה ללא פלט למסך, אז הפלטים זהים.

בצעו בדיקה זו עבור כל קובץ בדיקה. לנוחיותכם, הדגמת השלבים נמצאת בלינק [זה](#).

## הגשה

הגשה במודל, לפי הפורמט הבא:

1. עליכם ליצור קובץ zip (השתמשו ב-zip או gzip בלבד) בשם hw0\_id1\_id2.zip, כאשר id1, id2 הם מספרי תעודות הזהות של המגישים.

2. קובץ ה-zip מכיל אך ורק את הקבצים הבאים, ללא תתי-ספריות.

calc.c ⊕

⊕ submitters.txt, שמכיל את מספרי הזהות והשמות של מגישי התרגיל, מופרדים ע"י פסיק. למשל:

Bill Gates,bill@microsoft.com,123456789
Linus Torvalds,linus@gmail.com,234567890

3. צרו את קובץ ה-zip באמצעות הפקודה

```
zip hw0_id1_id2.zip calc.c submitters.txt
```

4. הגישו את ה-zip דרך המודל.

לנוחיותכם, הדגמה של שלבים 1-3 בלינק [זה](#).