

# אתגר 15

## מערכת קבצים

### מטרה

באתגר זה תבנו מערכת קבצים

אהל השלבים שתעבורו:

מבנה מערכת קבצים	لמידה
להבין איך הסימולטורעובד לבנות מערכת קבצים	הקדמה הסימולטור

### כווית במסמר

### תוכן

מטרה .....	1
נתרגל מיזוגיות החשבות .....	2
לינקים שימושיים .....	2
למידה .....	3
בנייה מערכת קבצים .....	5
זהו!!! .....	7
היעזרו בצליליסט כדי להגשים משימה מושלמת .....	7
נספחים .....	8
דגשים לתוכנות נכון .....	8

## נתרגל מילומניות חשובות

- נבין איך לਮפות מידע לזריכון המשותף.
- נבין איך לשנות מידע בזיכרון משותף.

## لينקים שימושיים

- כדאי לקרוא גם [דגשים לתוכנות נכון](#).
- בסוף יש [צ'קליסט שימושי](#) למקבב אחרי התקדמות בביצוע המשימה.

**בהצלחה يا אלופות ואלופים!**



# למידה

בנייה מערכת קבצים

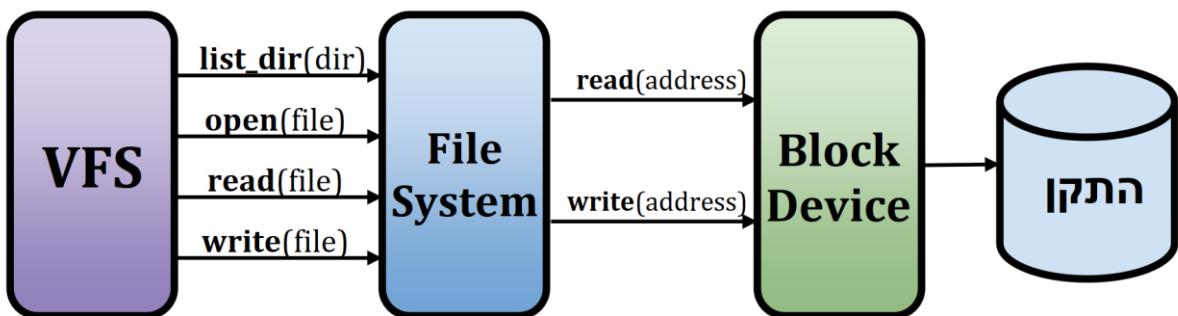
למידה  
רמת קושי: קלה



בתרגיל זה אתם תתנסו בכתיבת מערכת קבצים פשוטה בסביבת Linux, אשר תיראוzelf.

## הקדמה

בעולם כתיקונו, מערכת קבצים הינה רכיב היושב בתוך גרעין מערכת הפעלה וממשקיו מתוארים באופן כללי, באյור הבא:



באשר:

- שכבת ה-VFS הנה השכבה אשר מקבלת הוראות מהמשתמש ע"י קריאות מערכת (syscalls) ומתרגמת את הבקשות של המשתמש לפונקציות של שכבת ה-filesystem. שכבה זו עובדת ברמת הקבצים והתיקיות, ואני מודעת לכתובות בהתקן האחסון בכלל.
- שכבת ה-filesystem מקבלת קריאות המתייחסות לקבצים ותיקיות, ומירגת אותן לבקשת read/write משכבה ה-block device. כאמור, תפקיד שכבה זו הוא לתרגם בקשות ברמת הקבצים לבקשת ברמת כתובות בהתקן האחסון.
- שכבת ה-block device אחראית לדבר עם התקן האחסון הספציפי, שכן התקנים שונים מתקשרים באופן שונה. לדוגמה, גמוך-on-keyboard דיסק קשיח הינט סוגים שונים של התקני אחסון, אך האחד מחובר באמצעות חיבור USB והשני באמצעות SATA, ומכאן שלכל אחד יהיה שכבתה riêngה. שכבה זו עובדת ברמת הכתובות הספציפיות בהתקן האחסון ואני מודעת כלל לקבצים ותיקיות.

## הסימולטור

בתרגיל זה תכתבו מימוש משלכם למערכת קבצים בסיסית אשר תיקרא `myfs`. מכיוון שבתיבת מערכת קבצים בtower גרעין מערכת הפעלה דורשת מיומנויות שעדין לא למדנו בקורס ודרגת מומחיות גבוהה בעבודה עם גרעין מערכת הפעלה לינוקס, סיפקנו לכם סימולטור המדמה את שלוש השכבות הנ"ל, כאשר עלייכם להשלים את שכבת ה-`filesystem`.

הסימולטור נבדל מממשק הפעלה בשני היבטים עיקריים:

- שכבת ה-SFS בסימולטור מתקשרת באופן ישיר עם היוזר באמצעות ממשך שורת פקודה פשוט.
- שכבת ה-Block Device אינה משתמשת בהתקן אחסון אמיתי על מנת לאחסן את המידע, אלא פשוט בקובץ.

הורידו את הקובץ המצורף לtower מكونת הלינוקס שלכם. הקובץ הוא מסובgzar-tar-זהו קובץ דחוס (בדומה לפורמט קובץ המוכר מווינדוס). על מנת לפתוח אותו, אתם יכולים להשתמש בפקודה:

```
$tar xf <tar.gz file path>
```

באשר תעשו זאת, תיווצר תיקיה בשם `fs`. על מנת לkomfl את הפרויקט, בנסו לתיקיה זו והקישו את הפקודה `make`, אשר תקמפל ותיצור את הקובץ הבינארי `fs/myfs`. זהו הסימולטור.

ניתן להריץ את הסימולטור באופן הבא:

```
$ ./bin/myfs <some file>
```

באשר הפקטור `some file` מתייחס לשם הקובץ בו יאחסן המידע של שכבת ה-Block Device. אם תעבירו שם של קובץ שלא קיים, הסימולטור ייצור את הקובץ.

## בנייה מערכת קבצים

בנייה מערכת קבצים

רמת קושי: קשה מאד

למידה

### תיכנות – הגשת הפרויקט שקיבלתם לאחר השלמתו



1. הריצו את הסימולטור על קבצים שונים והבינו כיצד לعباد איתו (מומלץ להשתמש בפקודה `help`). כמו כן, מהר מאוד תבחןנו כי כמעט כל פעולה שתבצעו תביא להודעת השגיאה "not implemented".
2. עברו בקצרה על קבצי הקוד של הסימולטור והבינו באופן כללי את תפקוד כל שכבה.
3. השלימו את המימוש בקובץ `myfs.cpp`.

להלן דוגמת ריצה של סימולטור תקין:

```
$ ./bin/myfs /tmp/some_new_file
Did not find myfs instance on blkdev
Creating...
Finished!
Welcome to myfs
To get help, please type 'help' on the prompt
below.
myfs$ touch file_a
myfs$ edit file_a
Enter new file content
This is some file content
myfs$ ls
file_a 26
myfs$ touch file_b
myfs$ ls
file_a 26
file_b 0
myfs$ cat file_a
This is some file content
myfs$ exit
```

כמו כן, וודאו שהמידע אכן נשמר בין הרצאות שונות של הסימולטור:

```
$ ./bin/myfs /tmp/some_new_file
Welcome to myfs
To get help, please type 'help' on the prompt
below.
myfs$ ls
file_a 26
file_b 0
myfs$ cat file_a
This is some file content
myfs$ exit
```

## הערות והקלות

1. שימושו לב: על מנת להשלים את התרגיל אתם נדרש לעורר רק את הקבצים `h/cpp/myfs.h`, אין צורך לשנות חלקו הקוד אחריהם. כמו כן, אין צורך להקדיש זמן רב להבנת השכבות השונות של הסימולטור.
2. על מנת להקל על המימוש, אין צורך לתמוך בתיקיות. אם המשתמש מבקש ליצור תיקיה, מותר לבם להציג `implemented not`. כמו כן, אתם יכולים להניח כי כל הנתיבים (`paths`) שתקבלו הם מהצורה "filename" ואינם כוללים תיקיות בכלל.
3. מותר להניח כי כל שמות הקבצים במערכת `fsy` הם באורך של לכל היותר 10 תווים.
4. מומלץ בחום לבצע על המציג בשיעור `filesystem`, ובאופן ספציפי על החלק של `ext`, לפחות על מנת לקבל רעיונות למימוש. למשל, ניתן להקדיש חלק מתחילת ה-`Block Device` לטבלה אשר מסמנת איפה כל קובץ מתחליל ואיפה הוא נגמר.

## בונוס

הוסיפו תמיכה בתיקיות

## זהו!!!

היעזרו בצל' קליסט כדי להגיש משימה מושלמת

### משימה 2 – הגשת הפרויקט

- יצירת קובץ
- הצגת רשימת הקבצים
- עריכת קובץ
- הצגת מידע בקובץ
- בונוס – תמיינה בתיקיות

סיימתם? עבשו!  
רוכדים



## **נספחים**

### **דgesים לתוכנות נכון**

- כדאי ל亙טף כל מספר שורות קוד ולא לחכות לסוף! הרובה יותר קל לתקן מאשר אין הרבה שגיאות קומפיילציה. בנוסף קל יותר להבין מאיפה השגיאות נובעות.
- כדאי לבתוב פונקציה ולבדק אותה לפני שתת媚ם ממשיכים לפונקציה הבאה. חישבו על מקרים קצה ונסו לראות מה קורה.
- בכל פעם שתת媚ם מתקנים משהו, זכרו שיכל להיות שפגעתם במשהו אחר. לכן עלייכם לבדוק שוב מהתחלה.