

# Linux Fundamentals

# Linux Fundamentals

מאת  
חיים צדוק  
גריגייל בע"מ

**GRIGALE**  
www.grigale.com

Copyright © 2017 by Grigale LTD  
כל הזכויות שמורות לגריגייל בע"מ 2017

## תוכן עניינים

3	תוכן עניינים
6	הכרת מערכת לינוקס
7	מבוא למערכות הפעלה פתוחות
8	מהי מערכת הפעלה פתוחה
8	מהי תכנה חופשית
8	ארגוני קוד פתוח
9	היתרונות של תכנה חופשית
9	היסטוריה של מערכות יוניקס
10	היסטוריה של לינוקס
10	הנחות ידועות בסביבת יוניקס/לינוקס
10	סביבה מרובת משתמשים
10	סביבה מרובת תהליכים
11	מבנה מערכת ההפעלה
12	התחברות ראשונית
12	המשתמש root
12	התחברות למערכת דרך חלון Login
13	מבנה מערכת ההפעלה
14	שכבת הגרעין
14	מודולים
14	שירותי שם
15	הקשר בין התקנים למודולים
15	תפקיד שכבת הגרעין
15	קריאות מערכת
16	המעטפת
17	מערכת הקבצים
18	שכבת ממשק המשתמש החלונאי
18	מערכת ה-X Window
18	Containers
19	התקנת לינוקס
20	MBR – טבלת המחיצות
21	עבודה עם Gparted
21	התקנת לינוקס כמערכת עצמאית
22	התחלה

22	בחירת שם ותכונות
22	ניהול זיכרון
22	דיסק קשיח וירטואלי
22	סוג אחסון דיסק קשיח
22	דיסק קשיח וירטואלי
22	סיכום
23	שימוש בסביבת החלונות
24	שימוש בסביבות עבודה (Workspaces)
25	שימוש בממשק הפקודה
26	למה צריך שורת פקודה
26	השלמה על ידי TAB
26	התחברות דרך ממשק פקודה
28	שימוש במערכת הקבצים
29	מבנה מערכת הקבצים
29	ספריות בעלות משמעות מיוחדת
31	מיקום במערכת הקבצים
31	Inode
34	עבודה עם ספריות
34	עבודה דרך מנהל החלונות
34	עבודה דרך המעטפת
34	הרשאות לספריות
36	עבודה עם קבצים
36	עבודה דרך מנהל החלונות
36	עבודה דרך המעטפת
40	שימוש במעבד התמלילים
41	vi pocket guide
43	המעטפת - The Shell
44	מהי המעטפת
45	תווים מיוחדים במעטפת
46	Redirection – 1 Pipe
46	- Pipe
46	>< Redirection
47	משתני סביבה
47	משתני סביבה מקומיים
48	משתני סביבה גלובליים

48..... יצירת פקודות פנימיות חדשות - Alias

49..... קבצי אתחול

# הכרת מערכת לינוקס

## פרקי הלימוד

מבוא למערכות הפעלה פתוחות  
מבנה מערכת הפעלה פתוחה  
התקנת מערכת הפעלה פתוחה  
שימוש בסביבת החלונות  
שימוש בממשק הפקודה  
שימוש במערכת הקבצים  
שימוש במעבד תמלילים  
מהי המעטפת (סביבת ממשק הפקודה)  
בקרת תהליכים ועבודות  
חיפוש בתוך קבצים ובתוך ספריות  
קריאה וכתיבת סקריפטים  
דחיסה ויצירת קבצי ארכיב  
שליטה מרחוק  
העברת קבצים בין מערכות

## **מבוא למערכות הפעלה פתוחות**

נושאי הלימוד:

- מהי מערכת הפעלה פתוחה
- ארגוני קוד פתוח
- היתרונות של תכנה חופשית
- היסטוריה של מערכות יוניקס
- היסטוריה של לינוקס
- הנחות יסוד ידועות בסביבת יוניקס/לינוקס
- סביבה מרובת משתמשים
- סביבה מרובת תהליכים
- וירטואליזציה
- מחשוב ענן

## מהי מערכת הפעלה פתוחה

מערכת הפעלה פתוחה הינה מערכת הפעלה העומדת בתנאים הבאים:

1. קישוריות - יכולת לתקשר עם מערכות אחרות
2. יבילות – היכולת להעביר נתונים ותוכנות למערכות אחרות
3. בעלת סטנדרטים פתוחים -

1. כתובה לרוב כתוכנה חופשית (קוד פתוח ושימוש חינם)
2. מודולרית - כתובה לרוב במארג שכבות.
3. ניתן להפיץ את מערכת הפעלה עם או ללא שינויים\*

## מהי תכנה חופשית

ניתן להגדיר תכנה חופשית כתכנה המאפשרת למשתמש מספר חריות בסיסיות:

1. החופש להשתמש בתוכנה לכל מטרה.
2. החופש ללמוד את דרכי הפעולה של התוכנה ולהתאים אותה לצרכיו.
3. החופש להפיץ מחדש עותקים של התכנה לאנשים אחרים כדי לסייע להם.
4. החופש לשפר את התכנה ולשחרר את השיפורים לציבור כדי שכל הקהילה תרוויח מכך.

התקן המלא של מערכת הפעלה פתוחה נמצא באתר The Open Group .

## ארגוני קוד פתוח

ארגוני הקוד הפתוח הידועים הינם:

1. פרויקט GNU - פרויקט GNU נוסד ב-1983 על ידי ריצ'רד סטלמן מ-MIT מתוך מטרה לפתח תוכנה חופשית מתוך רצון לאי תלות בשימוש בתוכנות קנייניות.
2. ארגון The Open Group – הנו קונסורציום שנוסד בשנת 1996 על מנת לפתח סטנדרטים פתוחים על ידי יצרני התוכנה. בין החברים בארגון זה: פוג'יטצו, סאן, היטאצ'י, היולט פאקארד, יבם ועוד...
3. המוסד לתכנה חופשית – Free Software Foundation , מלכ"ר שנועד לתמוך בתנועת התכנה החופשית





## היסטוריה של לינוקס

לינוקס הנה מערכת הפעלה שפותחה על ידי לינוס טורוולדס כאשר היה סטודנט למדעי המחשב באוניברסיטת הלסינקי בפינלנד. המרצה שלו – פרופ' טננבאום פיתח את מערכת ההפעלה – מיניקס ולינוס רצה לתרום לפיתוח המערכת ההפעלה אולם לא קיבל אישור לעשות זאת ולכן התחיל לפתח מערכת הפעלה משל עצמו כמערכת שכל מי שרוצה לתרום קוד למערכת ההפעלה יכול לעשות כן.

לינוקס הינה מערכת הפעלה יציבה, עשירה בתכונות ומתקדמת מאוד. המערכת נחשבת מאוד בטוחה. ללינוקס מגוון של יכולות אבטחה המעמידות אותה בתקני צבא ארה"ב המחמירים (תקן C2). המערכת מכילה מגוון אפליקציות משרד ומאפשרת התקנה של עשרות אלפי אפליקציות צד שלישי קוד פתוח או קנייניות.

## הנחות ידועות בסביבת יוניקס/לינוקס

- כל דבר הוא קובץ – גם גישה להתקן תוכנה או התקן חומרה נעשית על ידי שימוש בקבצים. דבר זה יוצר הפשטה מאוד נוחה מבחינת צורת הניהול והתכנות שצריכה להתבצע מול התקן חומרה, צורת התכנות הנה עבודה מול קבצים – פתיחה, קריאה וסגירה של קבצים.

הרבה מסגרות עבודה במערכת ההפעלה מאוגדות כמערכות קבצים.

- כל דבר שרץ במערכת ההפעלה הוא תהליך – כל הפעלה של קובץ בינרי או תסריט (script) במערכת יוצרת תהליך. ישנה טבלת תהליכים המנוהלת על ידי גרעין מערכת ההפעלה וניתן לראות את פרטי התהליך ומידת המשאבים שהוא תופס.
- מערכת מרובת תהליכים – המערכת יכולה לשרת ולהריץ מספר תהליכים במקביל.
- מערכת מרובת משתמשים – המערכת יכולה לשרת מספר משתמשים המתחברים בו זמנית אל המערכת.

## סביבה מרובת משתמשים

סביבה המאפשרת גישה מקבילה למספר משתמשים למשאבים שברשותה

## סביבה מרובת תהליכים

סביבה המאפשרת למספר תהליכים לרוץ בו זמנית על ידי שימוש בחלוקת זמן המעבד לפלחי זמן.

## **מבנה מערכת ההפעלה**

נושאי הלימוד:

- התחברות ראשונית
  - המשתמש root
- התחברות למערכת
- מבנה מערכת ההפעלה
- שכבת הגרעין (Kernel)
  - הגרעין
  - מודולים (Modules)
  - המעטפת (Shell)
- מערכת הקבצים (File System)
- מנהל החלונות – ממשק משתמש גרפי (GUI)
  - שרת ה-X
- מנהל החלונות GNOME
  - אזורי הפעלה (Zones)
  - מעקב דינמי (Dtrace)

## **התחברות ראשונית**

### **המשתמש root**

במערכת ההפעלה קיים משתמש מערכת שנוצר בעת התקנת מערכת ההפעלה ושמו root. משתמש זה הנו המשתמש בעל ההרשאות הגבוהות ביותר ובעל גישה בלתי מוגבלת לכל משאבי המערכת.

מומלץ לא לבצע התחברות למערכת ההפעלה עם משתמש root, אלא רק למטרות תחזוקה במערכת בלבד. ברוב סוגי המערכות הפתוחות המשתמש root הנו משתמש מסוג תפקיד ולא ניתן לבצע התחברות ישירה בעזרתו אלא רק לרשת(מלשון ירושה) ממנו יכולות.

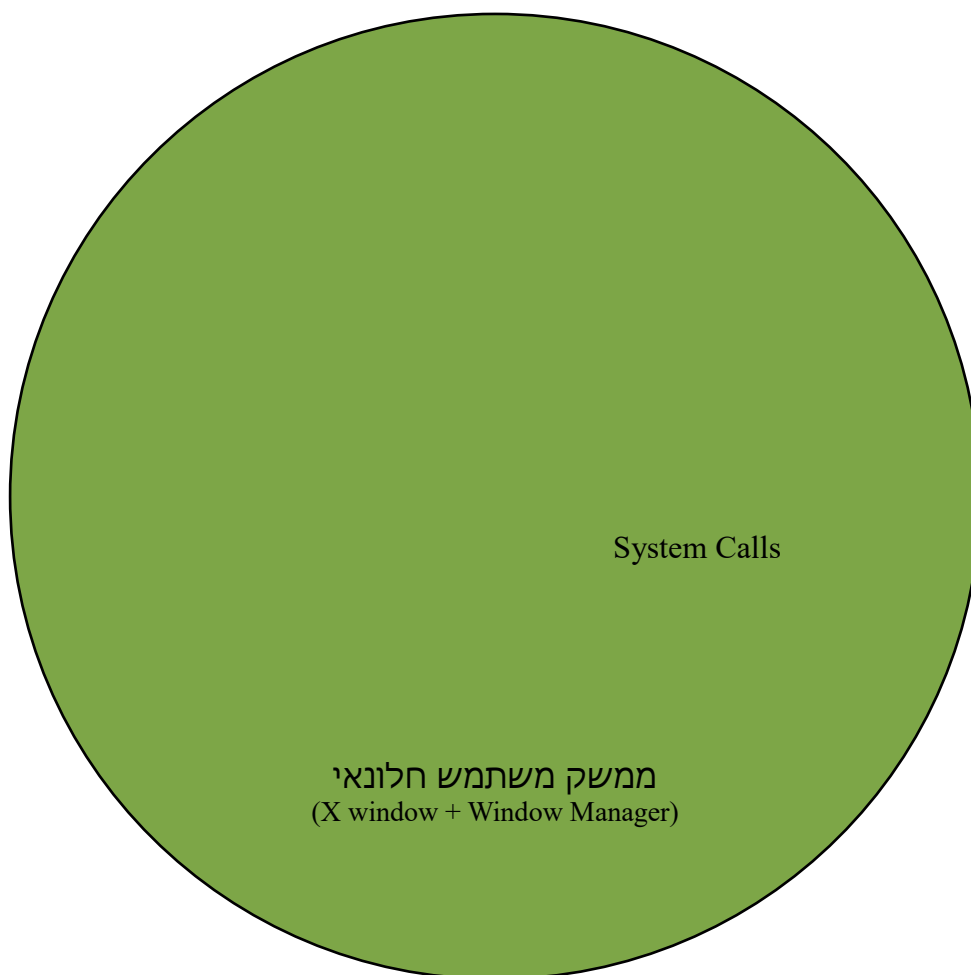
### **התחברות למערכת דרך חלון Login**

על מנת להתחבר למערכת יש להקיש שם משתמש וסיסמה. חשוב להקפיד על אותיות קטנות וגדולות בעת הקשת שם המשתמש והסיסמה. מנהל החלונות ברירת מחדל ברוב המערכות הפתוחות הינו Gnome (אולם גם KDE הנו מנהל חלונות מאוד נפוץ).

## מבנה מערכת ההפעלה

מערכת הפעלה פתוחה בנויה ממספר שכבות תוכנה.

1. שכבת הגרעין
2. שכבת המעטפת
3. שכבת היררכיית הקבצים
4. שכבת ממשק משתמש החלונאית



## שכבת הגרעין

שכבת הגרעין – מכילה את תוכנת הגרעין והמודולים.

הגרעין של לינוקס נקרא Kernel והינו לב מערכת ההפעלה. Kernel מורכב מקובץ ששמו vmlinux והוא נמצא בדרך כלל תחת הספרייה /boot .

נכון להיום לינוקס נפוצה בסביבת אינטל כמערכת הפעלה למחשבים ושרתים ומכונות וירטואליות ובסביבת ARM – כמערכת אנדרואיד.

לינוקס נפוצה גם כמערכת משובצת מחשב על פלטפורמת PowerPC למעשה רוב הראוטרים הביתיים מבוססים לינוקס על פלטפורמה זו. לינוקס נפוצה גם בתחומים אחרים כגון: טלויזיות חכמות, מערכות מדיה ואודיו, מערכות אוויוניקה, מערכות בקרה.

## מודולים

מודול הנו קוד המאפשר להרחיב את שכבת הגרעין. ישנה אפשרות לטעון לגרעין מודולים נוספים המרחיבים את יכולות הגרעין. מודולים אלו יכולים לטפל בהתקנים שונים או להוסיף תכונות אפליקטיביות נוספות שאינן קיימות בגרעין.

מודולים הנם רכיבי תוכנה המקושרים דינמית לגרעין מערכת ההפעלה. מודולים ניתן להריץ לפי דרישה ע"מ לקבל שירות להתקן חומרה או משאב תוכנה במערכת. לדוגמה המודול ( xfs filesystem) , xfs, אחראי על טיפול במערכת הקבצים מסוג xfs. אם במערכת ההפעלה אין מערכת קבצים מסוג xfs זמינה, מודול זה לא ירוץ במסגרת הגרעין.

על מנת לצפות במודולים ניתן להריץ את הפקודה **lsmod**.  
ניתן לראות פרטים של המודול על ידי הרצת הפקודה - **modinfo xfs**

## שירותי שם

שירות שם הנה דרך נפוצה במערכת ההפעלה לתרגם בין כתובות מספריות לשמות ואנו ניתקל בשירות שם במסגרות עבודה רבות במערכת ההפעלה.

## הקשר בין התקנים למודולים

לכל התקן במערכת יש מודול שמייצג את פעילות ההתקן. מספר המודול שמוצג בהתקן נקרא גם מספר ראשי - Major number. אם קיימים כמה התקנים בעלי אותו מספר מודול ראשי הדרך להבדיל ביניהם הנה על ידי מספר משני. ניקח לדוגמה את הדיסק קשיח שלנו:

```
[root@ol74vm1 etc]# ls -l /dev/sda
brw-rw----. 1 root disk 8, 0 Nov 7 06:51 /dev/sda
[root@ol74vm1 etc]# ls -l /dev/sda1
brw-rw----. 1 root disk 8, 1 Nov 7 06:51 /dev/sda1
[root@ol74vm1 etc]# ls -l /dev/sda2
brw-rw----. 1 root disk 8, 2 Nov 7 06:51 /dev/sda2
[root@ol74vm1 etc]#
```

אנו רואים כי מספר המודול (מספר ראשי) שמטפל בהתקן הנו 8. המספר המשני של ההתקן הספציפי עליו אנו מסתכלים הוא 0.

למתקדמים:

הדרך ליצור קבצי התקן במערכת ההפעלה הינה על ידי הפקודה: `mknod`

לדוגמה:

`mknod /dev/haim c 8 0`

## תפקיד שכבת הגרעין

- לטפל במערכות הקבצים השונות – מערכות קבצים
- ניהול התקנים – התקני ערוץ פנימיים וחיצוניים שונים ומגוונים (PCI, FC, USB), התקני אחסון, תקשורות רשת חוטיות/אלחוטיות מגוונות והתקני קלט/פלט.
- טיפול בתהליכים - לספק שירותים ומשאבים לאפליקציות המשתמש במערכת ההפעלה.
- ניהול זכרון – ניהול זכרון ראשי (RAM) וזכרון וירטואלי (SWAP).

## קריאות מערכת

גרעין מערכת ההפעלה הנו התוכנה עם העדיפות הגבוהה ביותר במערכת ואחת החשובות בהן. בעת תקלה בגרעין המערכת יפעיל הגרעין את השגרה Panic ויבצע מנגנון של אתחול מהיר ללא יציאה מסודרת. על מנת למדר את הגישה למשאבי הגרעין נקבע כי השיטה בה אפליקציות יפנו אל הגרעין תהיה על ידי שימוש בקריאות מערכת. קריאות מערכת (System Calls) הנה הדרך היחידה בה ניתן לפנות לגרעין המערכת על מנת לבקש שירות.

הפקודה `man syscalls` מסבירה מהן `System calls`. לדוגמה כדי להבין מהי הקריאה `open` ניתן להריץ את הפקודה: `man -s 2 open`

## המעטפת

המעטפת הנה ממשק שורת הפקודה.  
המעטפת הנה תוכנה המכילה מפענח פקודות ותוכנות שירות אינטראקטיביות לשימוש המשתמש. ישנן כמה סוגי תוכנות מעטפת הנמצאות בסביבת סולריס.

**Bourne Shell** - המעטפת המקורית איתה הגיע יוניקס  
**C Shell** - מעטפת עם מספר תוספות כגון היסטוריית פקודות, ניהול עבודות וכו  
**BASH** - מעטפת שהנה פרויקט של ארגון גנו ומכילה מספר גדול של תכונות, מעטפת זו הנה מעטפת ברירת מחדל בהרבה סוגי מערכות פתוחות וגם בלינוקס.

מבנה פקודה:

**<command > [-labcg] [arg1] <arg2|arg3>**  
**switches arguments**

סוף שיעור 1 – 7.11.2017

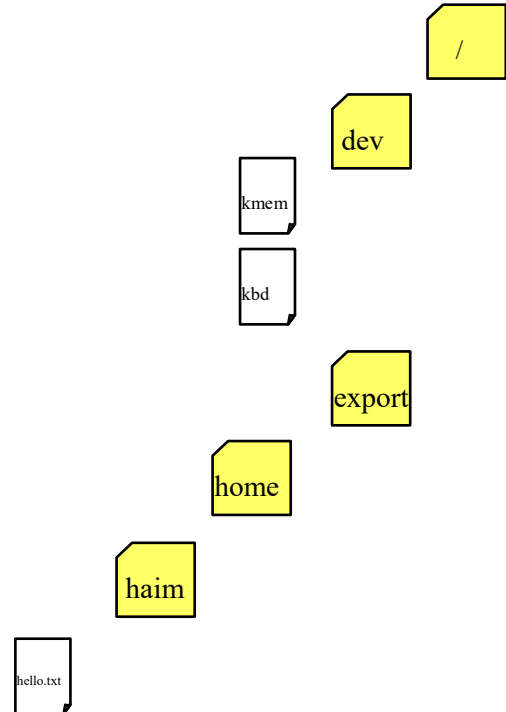


## מערכת הקבצים

מערכת הקבצים הנה מקום בו מאורגנים ספריות וקבצים.

ספריית הקבצים מתחילה בספרייה ראשית ששמה ספריית השורש (root directory) וסימונה במעטפת הינו: /

להלן מבנה אופייני של עץ הקבצים במערכת פתוחה:



כדי לעבור מספרייה אחת לשניה בעץ הספריות יש להשתמש בפקודה `cd`. לדוגמא כדי להגיע לספרייה `haim` בהנחה שאנו בספריית השורש יש לכתוב:

```
cd /export/home/haim
```

כדי ליצור את הקובץ הריק `hello.txt` יש לכתוב:  
**touch /export/home/haim/hello.txt**

כדי לשנות שם של קובץ מהשם `hello.txt` לשם `good.txt` יש לכתוב:

```
mv /export/home/haim/hello.txt /export/home/haim/good.txt
```

כדי למחוק את הקובץ `good.txt` יש לכתוב:

```
rm /export/home/haim/good.txt
```

כזכור, הרבה מסגרות עבודה במערכת ההפעלה משתמשות במערכת קבצים כדרך לארגון נתונים אבסטרקטיים. ניתן לתת כדוגמה את עץ מערכת ההתקנים שנמצא תחת הספרייה `/dev` לדוגמה הדיסק קשיח שלנו במערכת הינו הקובץ `./dev/sda`.

## שכבת ממשק המשתמש החלונאי

שכבת ממשק המשתמש החלונאי במערכות הפתוחות מורכבת מ:

- מערכת ה-X Window
- ממנהל חלונות.

### מערכת ה-X Window

מערכת ה-X Window פותחה באוניברסיטת MIT תחת רישיון קוד פתוח מסוג MIT.

תפקידי מערכת ה-X Window:

זוהי תוכנה ופרוטוקול רשת המספק ממשק משתמש גרפי למחשבים ברשת. התוכנה מספקת שכבת חומרה מופשטת לניהול חומרה ללא תלות בחומרה ייעודית. התוכנה מאפשרת לכל תוכנת חלונות מכל מערכת הפעלה לרוץ על מערכת ה-X window על ידי שימוש במערכת הרשאות וגישה לאותה מערכת X, ללא תלות בסוג מערכת הפעלה או ארכיטקטורת החומרה שמתחת למערכת ההפעלה.

במקור פותחה מערכת ה-X על מנת לספק ממשק משתמש למחשבים רזים (Thin clients), אולם עקב הנפיצות של התוכנה היא התפשטה לשימוש ברוב המערכות הפתוחות.

מערכת GNOME

מערכת GNOME או בשמה המלא: Gnu Network Object Model Environment הנו פרויקט בינלאומי של ארגון גנו שנוצר בשנת 1997. הפרויקט מושתת על הספרייה הגרפית +GTK והוא בנוי ממספר גדול של תת-פרויקטים המספקים יחדיו סביבת שולחן עבודה מלאה.

לדוגמה אחד מתתי הפרויקטים הינו `nautilus` (מנהל קבצים חלונאי) שהנו פרויקט שפותח על ידי חברת הסטרט-אפ `Eazel` בין השנים 1999-2001 ונתרם לקהילה.

## Containers

Containers הנם תכונה ייחודית לסביבת לינוקס. Container הנו אזור הפעלה מבודד להרצת אפליקציות. ניתן לומר כי Container הינו מופע (instance) משוכפל של מערכת ההפעלה עליה הוא רץ אולם בשונה ממכונה וירטואלית הוא אינו מכיל גרעין מערכת הפעלה, מה שהופך אותו למאוד קל משקל ובעל צריכת משאבים מעטה.

## **התקנת לינוקס**

נושאי הלימוד:

- MBR - טבלת המחיצות
- עבודה עם Gparted
- התקנת לינוקס כמערכת עצמאית במחשב
- התקנת לינוקס כמערכת וירטואלית ב-VirtualBox

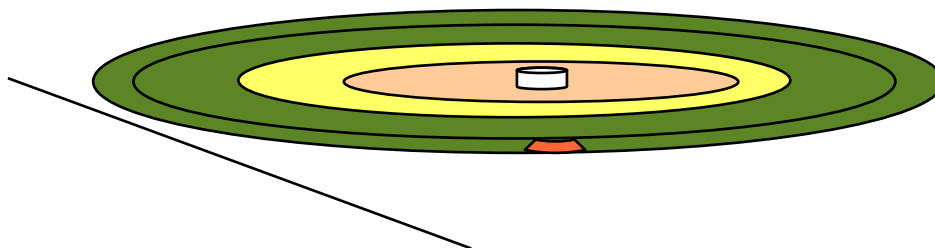
## MBR – טבלת המחיצות

בסביבת ארכיטקטורת X86 ישנה מוסכמה כי בכל דיסק קשיח המחולק למחיצות, סקטור ראשון של הדיסק יכיל את טבלת המחיצות הקיימת באותו דיסק קשיח.

מחיצה פיזית – ניתן להגדיר עד 4 מחיצות פיזיות בדיסק קשיח. מחיצה פיזית מכילה מערכת קבצים אחת תחתיה.

מחיצת הרחבה – על מנת לאפשר הגדרה של יותר מ-4 מחיצות, ניתן להגדיר מחיצת הרחבה (Extended partition) ובתוכה להגדיר מחיצות לוגיות. מחיצת הרחבה אינה מכילה מערכת קבצים תחתיה, אלא מחיצות לוגיות בלבד שבתוכן מערכות קבצים.

מחיצה לוגית – מחיצה הקיימת בתוך מחיצת ההרחבה. ניתן להגדיר כמות בלתי מוגבלת של מחיצות לוגיות, אולם ייתכן כי תהיה מגבלה ברמת מערכת ההפעלה.



Partition 3

## **עבודה עם Gparted**

שלב זה אינו חובה ויש לבצעו רק אם מערכת לינוקס מועמדת להיות מותקנת ביחד עם מערכת הפעלה חלונות ב-Dual boot.

### **התקנת לינוקס כמערכת עצמאית**

1. יש להכניס את התקליטור לינוקס לכוון התקליטורים ולבצע אתחול למחשב.
2. בעת עליית ה-BIOS בחר באופציית אתחול מתקליטור (Boot from CD-ROM).
3. לאחר עליית ההתקנה החלונאית יש לבחור בתפריט מתקדם של ניהול הדיסק ולפנות מקום כך שיותר מקום פנוי להתקנת לינוקס.

## **התקנת לינוקס כמכונה וירטואלית בסביבת VirtualBox**

יש לבצע הורדה של תוכנת VirtualBox מהאתר: [www.virtualbox.org](http://www.virtualbox.org) ולהתקינה על מערכת ההפעלה.

### **הגדרת מכונה וירטואלית מסוג לינוקס:**

#### **התחלה**

הפעל את ממשק המשתמש החלונאי של VirtualBox ובצע את סדר הפעולות הבא:  
לחץ על כפתור New -> לחץ Next

#### **בחירת שם ותכונות**

בתיבת הטקסט Name, הכנס שם למכונה הוירטואלית  
בתיבות הבחירה OS Type בחר לינוקס כסוג מערכת הפעלה, בחר לינוקס 64 ביט כגרסת מערכת הפעלה. או אם קיימת הגרסה המדויקת יש לבחור בה וללחוץ Next.

#### **ניהול זיכרון**

בחר בגודל הזיכרון הרצוי. דאג כי הגודל יהיה לא פחות מ- 2GB (הערך המינימלי האפשרי על מנת לאפשר עבודה עם מערכת החלונות). לחץ Next.

#### **דיסק קשיח וירטואלי**

דאג כי קיים סימון – Boot Hard Disk  
בחר באופציה – Create new hard disk.  
לחץ Next.

#### **סוג אחסון דיסק קשיח**

דאג סוג האחסון יהיה אחסון דינמי – Dynamically expanding storage  
לחץ Next.

#### **דיסק קשיח וירטואלי**

מיקום – קבע מיקום לדיסק הקשיח כרצונך.  
בחר בגודל – 16GB אמור להיות מספיק.  
לחץ Next.

#### **סיכום**

וודא כי הסיכום אכן מתאים לבחירותיך.

## **שימוש בסביבת החלונות**

נושאי הלימוד:

- בחירת אופציות התחברות
- התחברות למערכת
- ניהול קבצים על ידי שימוש בחלונות
- שימוש בסביבות עבודה
- שימוש בעזרה
- התחברות בטוחה במקרה של תקלות
- שינוי סיסמא למשתמש

## **שימוש בסביבות עבודה (Workspaces)**

Gnome מנהל החלונות בסביבה פתוחה מכיל אפשרות לעבוד עם סביבות עבודה. סביבת עבודה הנה שולחן עבודה וירטואלי וניתן להגדיר החל משולחן עבודה 1 או יותר.

ניתן לתת שם לשולחן עבודה וירטואלי על ידי כניסה למנהל הסביבות עבודה.

ניתן לעבור בין שולחנות עבודה על ידי שימוש ב- **Ctrl+Alt** + חץ למעלה או למטה.

התחברות בסביבה בטוחה במקרה שהגדרות מנהל החלונות או שולחן העבודה אינם תקינים, ניתן לבחור באופציית התחברות בטוחה.



## **שימוש בממשק הפקודה**

נושאי הלימוד:

- למה צריך את ממשק הפקודה ?
- השלמה על ידי TAB
- התחברות דרך ממשק הפקודה
- מבנה פקודה – פקודה אופציות וארגומנטים
- פקודות נפוצות
- שימוש בכלי עזר – Manual pages

בפרק זה יילמדו התכנים הבאים:

שימוש בממשק הפקודה. למה צריך את ממשק הפקודה ? כיצד להתחבר דרך ממשק הפקודה, מבנה פקודה – פקודה אופציות וארגומנטים. כיצד להשתמש בכלי עזר – Manual pages

## למה צריך שורת פקודה

שורת פקודה הנה ממשק טקסטואלי עשיר דרכו ניתן לתת הוראות למערכת ההפעלה לביצוע. הפקודות בסביבת מערכות פתוחות הן פקודות אנושיות. שורת הפקודה מאפשרת אוטומציה מלאה ושימוש בקבצי סקריפט כקובצי הוראות לביצוע פעולות בצורה סדרתית.

רוב הפקודות בסביבה פתוחה עומדות תחת תקני POSIX. POSIX הנו אוסף של תקנים המגדירים מהי סביבה פתוחה תקנית. POSIX מגדיר מה יהיה פורמט הפלט של פקודות בסביבה פתוחה. ניתן לקחת לדוגמה את הפקודה `ls -l` הפלט של הפקודה יהיה מורכב מאוסף של שדות קבוע וניתן לבצע פעולות מגוונות על הפלט מבלי לחשוש לאבד מידע או לקבל מידע מוטעה.

המונח מעטפת הינו - shell

## השלמה על ידי TAB

הדרך הכי נפוצה לעבוד מהר ויעיל בסביבות פתוחות הנה על ידי שימוש במקש ה-TAB על מנת להשלים פקודות או שמות קבצים. באם ברצוננו להשלים פקודה או שם של קובץ, ניתן להתחיל להקליד את שם הפקודה או שם הקובץ ולהשלים על ידי TAB. במידה וישנו אוסף פקודות או שמות קבצים מתאים לתחילת המילה שהוקלדה, יש ללחוץ פעמיים על מקש ה-TAB על מנת לקבל את כל אוסף הפקודות או הקבצים המתאימים.

## התחברות דרך ממשק פקודה

כאשר אנו נמצאים במערכות פתוחות ניתן להתחבר למערכות פתוחות מרוחקות אחרות על ידי שימוש בפקודה `ssh`.

`ssh` – הינה סביבת מעטפת מאובטחת (Secure shell). המאפשרת הצפנה של השיחה בין שני המחשבים ואפשרות להתחבר מרוחק בין שני מחשבים על ידי שימוש בהחלפת מפתחות.

שימוש ב- `putty`

`putty` הינה תוכנה קוד פתוח חופשית המאפשרת התחברות למערכות המריצות `ssh` כשירות גישה במערכת ההפעלה. חיפוש קצר בגוגל יאפשר להוריד את תוכנת `putty` ולעבוד אתה בגרסה הדורשת התקנה ( `putty-installer.exe` ) או בגרסה שלא דורשת התקנה ( `putty.exe` ).

מבנה פקודה:

<command> -lacbgs <arg1> <arg2|arg3>  
 switches arguments

פקודות נפוצות

הפקודה	אופציות אפשריות	תאור הפקודה
ls	ls -l צפיה עם יותר פרטים ls -a צפיה בקבצים חבויים ls -d צפיה במידע על ספריה ls -i על ספריה inodes צפיה במידע	צפיה בתוכן הספריה הנוכחית במעטפת.
pwd		המיקום הנוכחי של המעטפת בעץ הקבצים.
less		צפיה בתוכן של קובץ עם דפדוף עם יכולת חזרה לאחור
more		צפיה בתוכן של קובץ עם דפדוף ללא יכולת חזרה לאחור
exit/ Ctrl+D		יציאה מהמעטפת
cat		מציג תוכן של קובץ לפלט
script		מקליטה דו-שיח של פקודות במעטפת
who		מציג מי מחובר למערכת
cd	cd - חזרה למיקום אחרון cd ~ חזרה לספריית הבית שלנו	שינוי מיקום במערכת הקבצים

סוף שיעור 2 – 14.11.2017

## **שימוש במערכת הקבצים**

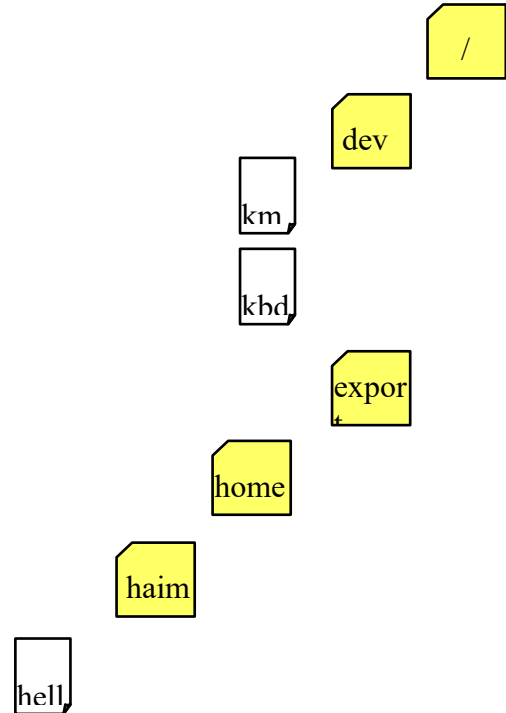
נושאי הלימוד:

- מבנה מערכת הקבצים
- עבודה עם ספריות
- עבודה דרך מנהל החלונות
- עבודה דרך המעטפת
- עבודה עם קבצים
- עבודה דרך מנהל החלונות
- עבודה דרך המעטפת

## מבנה מערכת הקבצים

מערכת הקבצים הנה מקום בו מאורגנים ספריות וקבצים. ספריית הקבצים מתחילה בספרייה ראשית ששמה ספריית השורש (root directory) וסימונה במעטפת הינו: /

להלן מבנה אופייני של עץ הקבצים במערכת פתוחה:



## ספריות בעלות משמעות מיוחדת

שם הספרייה	תיאור הספרייה	הערות
/	ספרייה הראשית במערכת הקבצים עוגנת תחת המחיצה הראשית	בשונה מסביבת Windows, אין בסביבת לינוקס כוננים כאותיות. כל ספרייה מתחת ל-/ יכולה להיות כונן או סתם ספרייה.
/etc	ספרייה בה שומרים קבצי הגדרות מערכת	קבצים ידועים בספרייה זו הינם קובץ /etc/hosts והקובץ ./etc/services
/dev	ספרייה בה נשמרים קבצי	קובץ התקן לוגי הינו קובץ

קישור לקובץ התקן פיזי והנו בעל שם אינפרומטיבי	התקן לוגיים	
בין היתר מכילה את ספריית bin נשמרים פקודות ידועות לשימוש המשתמש	ספרייה בה נשמרים קבצי עזר לשימוש המשתמש	/usr
בדרך כלל ספרייה זו הינה קיצור דרך לספרייה /usr/bin	ספרייה המכילה קבצים בינריים שהינם פקודות מעטפת ידועות לשימוש המשתמש	/bin
	ספרייה המכילה קבצים בינריים שהינם פקודות ידועות לניהול מערכת ההפעלה	/sbin
	ספרייה בה נמצאים ספריות הבית של המשתמשים	/home
קובץ - /var/log/messages הלוג הכללי במערכת לינוקס קובץ לוג /var/log/mysql.log של MySQL של מסד הנתונים -	ספרייה המיועדת לשמירת קבצי לוגיים	/var
בספרייה זו נמצאים קבצי הגרעין vmlinuz	ספרייה בה נשמרים קבצי העזר לאתחול המערכת	/boot
בספרייה זו נמצאים קבצי .so	ספרייה בה נשמרים ספריות קישור דינמיות וסטטיות	/lib
זוהי ספרייה שדומה ל- Network Neighborhood	ספרייה בה נמצאים שיתופי מערכות קבצים ממערכות אחרות	/net
	ספרייה בה נשמרים חבילות תוכנה חיצוניות	/opt
	ספרייה בה עוגנת מערכת הקבצים הווירטואלית לייצוג תהליכים במערכת ומידע על המערכת.	/proc
	ספריית הבית של המשתמש root	/root
	ספרייה בה נשמר מידע זמני	/tmp
	ספרייה המכילה מערכת קבצים וירטואלית המכילה מידע לשימוש מערכת ההפעלה	/sys

כדי לעבור מספרייה אחת לשנייה בעץ הספריות יש להשתמש בפקודה cd.

לדוגמא כדי להגיע לספרייה haim בהנחה שאנו בספריית השורש יש לכתוב:

```
cd /home/haim  
cd ~haim
```

כדי למצוא את מיקומנו במערכת הקבצים יש לכתוב:

```
pwd
```

כזכור, הרבה מסגרות עבודה במערכת ההפעלה משתמשות במערכת קבצים כדרך לארגון נתונים באותה מסגרת עבודה.

## מיקום במערכת הקבצים

מיקום יחסי – יתחיל מהספרייה הנוכחית  
מיקום אבסולוטי – יתחיל מספריית השורש ( / ).

במערכת פתוחה אין דרך לדעת האם הספרייה שאנו נמצאים בה הינה תחת מחיצה שונה (כמו כונן בסביבת Windows), או בתוך המחיצה בה נמצא - / (ספריית השורש).

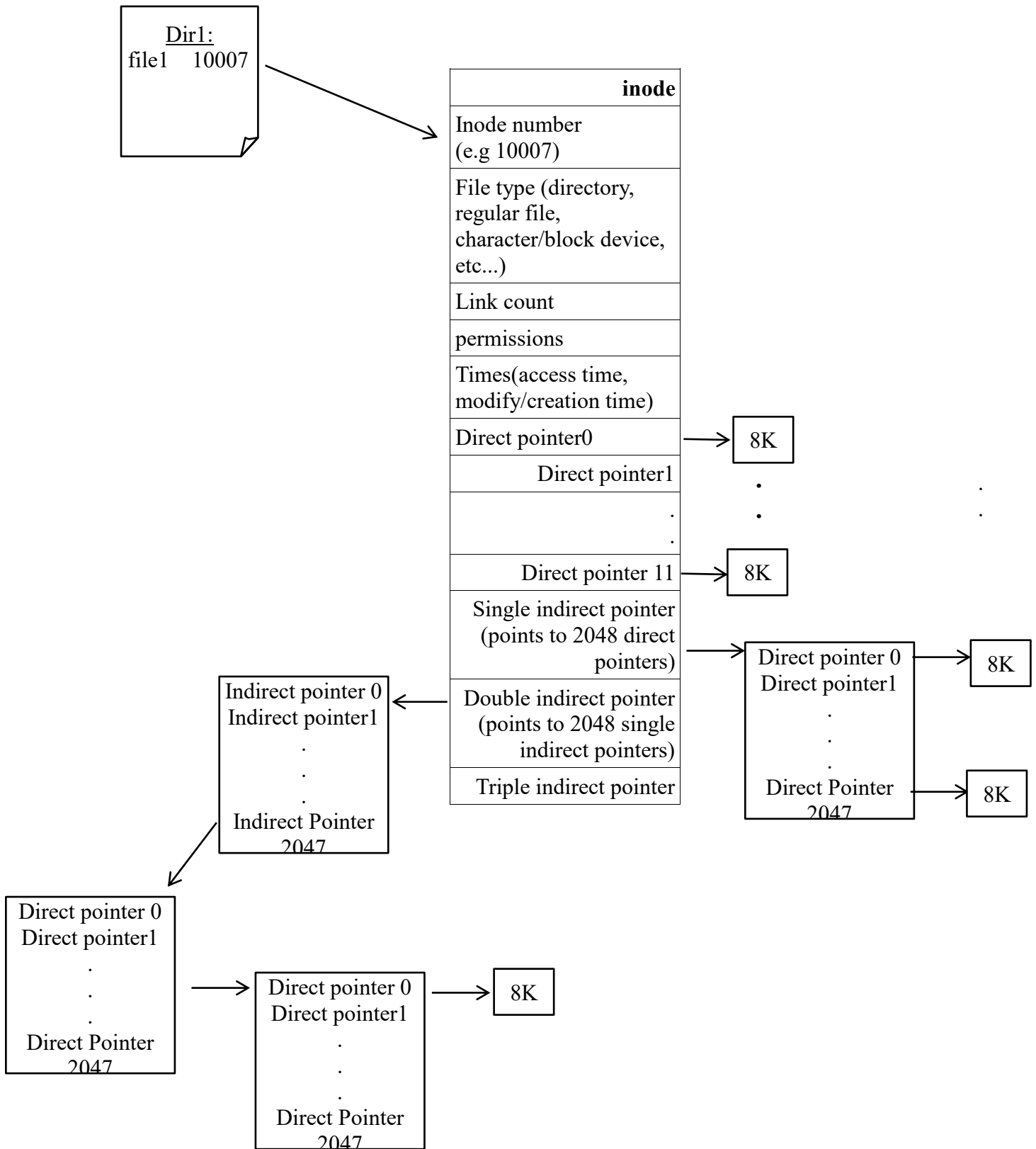
הדרך לראות פירוט של כל המחיצות הנה על ידי הפקודה:

```
df [-h]
```

כל שורת פלט בפקודה זו מתארת מערכת קבצים פיזית (שמאחוריה התקן פיזי) או וירטואלית (שמאחוריה מסגרת עבודה שמוצגת באופן וירטואלי כמערכת קבצים) במערכת.

## Inode

inode הינו מבנה נתונים, היחידה ממנה בנוי כל קובץ או ספרייה במערכת הקבצים. ה-inode שומר אינפורמציה על נתונים של הקובץ על ידי שימוש בשדות שמתארים את: מספר ה-inode, סוג הקובץ (ספרייה, קובץ רגיל, קובץ קישור או קובץ התקן), הרשאות, בעלויות, גודל, זמן עדכון, כמות קישורים של ה-inode לטבלאות ספריות ופוינטרים לבלוקים המכילים את המידע שבקובץ. כמות הקישורים של ה-inode יכולה לגדול אם יוצרים קישור חזק (hard link) בין שם של קובץ נוסף במערכת הקבצים לקובץ קיים, על ידי הפקודה ln. כאשר כמות הקישורים הינה אפס – מועברים כל הבלוקים של הקובץ לרשימת הבלוקים החופשיים לשימוש והקובץ נמחק מהמערכת. (זה בדרך כלל קורה על ידי הפקודה rm).





לדוגמא הפקודה הבאה מייצרת קישור חזק של שם קובץ f2 לאותו inode ש-f1 מחובר אליו.  
`ln f1 f2`

לדוגמא הפקודה הבאה מייצרת קישור חלש (symbolic link) של קובץ f4 לשם הקובץ f3  
`ln -s f3 f4`

שימו לב כי סוג הקובץ f4 הינו | – link.  
אם נמחק את הקובץ f3 הקובץ f4 יהפוך להיות קובץ מתנדנד (dangling) ללא בסיס מתחתיו.

# עבודה עם ספריות

## עבודה דרך מנהל החלונות

השימוש עם ספריות במנהל החלונות מאוד דומה לשימוש במערכת חלונות סטנדרטית.

## עבודה דרך המעטפת

משמעות של המושגים . ו- ..

## הרשאות לספריות

הרשאות לספריות מורכבת משלשה קבוצות של הרשאות כדי לראות הרשאות של ספריה ניתן לבצע את הפקודה:

`ls -dl dir1`

<b>rwx</b>	<b>rwx</b>	<b>rwx</b>
u-user	g-group	o-other
קבוצת הרשאות בעלים	קבוצת הרשאות קבוצה	קבוצת הרשאות לאחרים

משמעות הסימולים:

**r** – צפיה בתוכן ספרייה

**w** – מחיקה או הוספת קבצים

**x** – גישה לתוך הספרייה

**u** – המשתמש הבעלים של הקובץ

**g** – הקבוצה שהיא הבעלים של הקובץ

**o** – אחרים שאינם שייכים לקבוצה או שאינם הבעלים של הקובץ

**a** – כולם (גם המשתמש גם הקבוצה וגם אחרים)

222°

הרשאות שאינן בתוקף מסומנות בסימן מקף (-)

<b>rwx</b>	<b>rwx</b>	<b>r---</b>
<b>421</b>	<b>421</b>	<b>4</b>

## עבודה עם מספרים

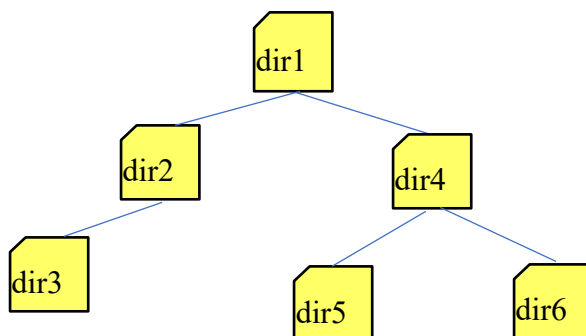
במקרה זה ישנן הרשאות מלאות לבעלים ולקבוצה אולם אין הרשאות מחיקה או הוספת

קבצים וגישה לספרייה לאחרים.

טבלת פקודות עיקריות לטיפול בספריות

הערות	תאור הפקודה	אופציות אפשריות	הפקודה
מקבילה לפקודה dir בסביבת דוס	צפיה בתוכן הספרייה הנוכחית במעטפת.	-l - מציג בצורה מפורטת	ls
	המיקום הנוכחי של המעטפת בעץ הקבצים.		pwd
	יצירת ספרייה	מאפשר יצירת -p נתיב ספריות	mkdir <new_dir>
	1. מאפשרת שינוי שם 2. מאפשרת הזזה של ספרייה למיקום אחר בעץ הקבצים 3. מאפשרת שינוי שם והזזה	1. mv a/d a/am 2. mv a/am a/b 3. mv a/am a/b/d	mv <src> <dst>
	העתקה		cp <src> <dst>
	מחיקת ספרייה ריקה בלבד		rmdir <empty-dir>
	מחיקת ספרייה וכל התוכן שלה		rm -r <dir>
+ הוספת הרשאה - הורדת הרשאה	שינוי הרשאות לספרייה	chmod u+r d1 chmod u-r d2 chmod g+w chmod g-w chmod o-x chmod o+x chmod a+x chmod ug+rw	chmod

כיצד ליצור את עץ קבצים הבא על ידי פקודה אחת של mkdir.?



תשובה:

mkdir -p dir1/dir2/dir3 dir1/dir4/dir5 dir1/dir4/dir6

# עבודה עם קבצים

## עבודה דרך מנהל החלונות

השימוש עם קבצים במנהל החלונות מאוד דומה לשימוש במערכת חלונות סטנדרטית.

## עבודה דרך המעטפת

הרשאות לקבצים מורכבת משלשה קבוצות של הרשאות כדי לראות הרשאות של קובץ ניתן לבצע את הפקודה:

`ls -l file1`

<b>rwx</b>	<b>rwx</b>	<b>rwx</b>
u-user	g-group	o-other
קבוצת הרשאות בעלים	קבוצת הרשאות קבוצה	קבוצת הרשאות לאחרים

r – צפיה בתוכן הקובץ  
w – שינוי תוכן הקובץ  
x – הרשאת ריצה  
u – המשתמש הבעלים של הקובץ  
g – הקבוצה שהיא הבעלים של הקובץ  
o – אחרים שאינם שייכים לקבוצה או שאינם הבעלים של הקובץ  
a – כולם (גם המשתמש גם הקבוצה וגם אחרים)

הרשאות שאינן בתוקף מסומנות בסימן מקף (-)

`rwx rwx r--`

במקרה זה ישנן הרשאות מלאות לבעלים ולקבוצה אולם אין הרשאות שינוי תוכן קובץ או הרשאת ריצה לאחרים.

## טבלת פקודות עיקריות לטיפול בקבצים

הפקודה	אופציות אפשריות	תאור הפקודה	הערות
<b>ls</b>	-l - צפיה מפורטת -a - מראה גם קבצים חבויים	צפיה בתוכן הספרייה הנוכחית במעטפת.	קובץ חבוי - קובץ שמתחיל בתו נקודה. -la - מראה גם מפורט וקבצים חבויים
<b>pwd</b>		המיקום הנוכחי של המעטפת בעץ הקבצים.	
<b>cd</b>	<b>cd -</b> - get back to the last visited directory.	מעבר לספרייה אחרת	מיקום יחסי - יתחיל מהספרייה הנוכחית מיקום אבסולוטי - יתחיל מספריית השורש ( / ).
<b>touch</b>		יצירת קובץ ריק	נועד בדרך כלל ליצירת קבצי דגל או לשינוי מועד עדכון קובץ
<b>mv &lt;src&gt; &lt;dst&gt;</b>	1. mv a/f1 a/f2 2. mv a/f1 a/dir1/ 3. mv a/f2 a/b/f3	1. מאפשרת שינוי שם 2. מאפשרת הזזה של קובץ למיקום אחר בעץ הקבצים 3. מאפשרת שינוי שם והזזה	
<b>cp &lt;src&gt; &lt;dst&gt;</b>		העתקת קובץ	לדוגמא: cp f1 f2
<b>rm -r &lt;dir&gt;</b>		מחיקת קובץ	
<b>chmod</b>	chmod u+r f1 chmod u-r f2	שינוי הרשאות לקובץ	+ הוספת הרשאה - הורדת הרשאה
<b>chown user f1</b> <b>chown user:group f1</b>	chown haim:proj a f1	שינוי בעלות משתמש או בעלות משתמש ובעלות קבוצה	
<b>chgrp proj b f1</b>		שינוי בעלות קבוצה בלבד	
<b>ln [-s] src dst</b>	-s - יצירת קישור סימבולי (symbolic link)	יצירת קובץ קישור (קיצור דרך)	

צורה נוספת לסימון הרשאות:

סימון הרשאות עם מספרים אוקטליים (מספרים בבסיס שמונה מספרים המיוצגים ע"י 0-7)

המספר	ההרשאה
0	---
1	--x
2	-w-
3	-wx
4	r--
5	r-x
6	rw-
7	rwx

המספרים 1,2,3 אינם נמצאים בשימוש לייצוג הרשאות. שינוי הרשאות לקובץ או ספרייה:

```
chmod 640 f1
chmod 750 dir1
```

הרשאות ברירת מחדל – UMASK  
 - umask - הפקודה מראה מה הם הרשאות ברירת מחדל (ללא פרמטרים)  
 - umask - מגדירה אילו הרשאות יוצרו בעת יצירת קובץ אם מספקים פרמטר אחד.

כיצד נקבעים הרשאות -

קביעת הרשאות לקובץ:  
 מפחיתים את המסיכה מ- 666

קביעת הרשאות לספרייה:  
 מפחיתים את המסיכה מ- 777  
 לדוגמא:  
 umask 027

- 777 בבסיס הרשאות של ספרייה

-

- 027 מסכה  
 - 750 הרשאות אפקטיביות

- 666 בבסיס הרשאות של קובץ

-

- 027 מסכה  
 - 640 הרשאות אפקטיביות

כאשר המסיכה היא 0 – יש הרשאות גבוהות יותר.  
כאשר המסיכה היא 7 – אין הרשאות.  
ספרת המאות ב-UMASK אחראית על הבעלים של הקובץ (Owner)  
ספרת העשרות ב-UMASK אחראית על הקבוצה של הקובץ (Group)  
ספרת האחדות ב-UMASK אחראית על אחר (Other)

**sticky bit** – איסור למחוק קובץ או ספרייה שאינם שייכים לי למרות שיש הרשאות בספרייה  
chmod +t dir1  
**setgid** – כל קובץ שיווצר, ייוצר עם בעלות קבוצה כפי שנקבעה בספרייה  
chmog g+s dir1  
**setuid** – ריצת הקובץ מתבצעת עם הרשאות של הבעלים  
chmod u+s script.sh

## שימוש במעבד התמלילים

נושאים

○ שימוש במעבד תמלילים VI

VI הינו אחד מתוכנות מעבדי התמלילים הותיקות הקיימות.

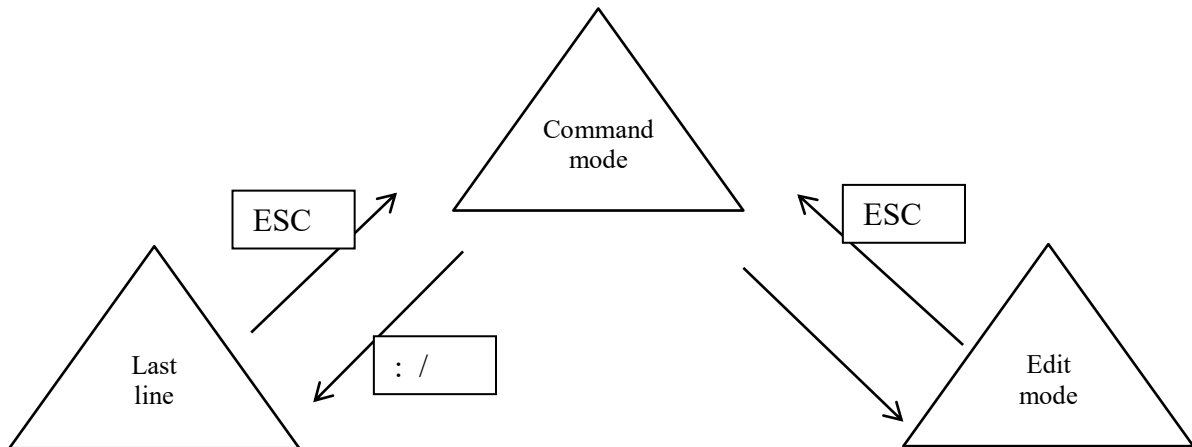
VI בנוי מ-3 מצבים:

- **מצב פקודה** – מצב זה הינו המצב ההתחלתי שאליו נגיע כשנפתח VI, על מנת לחזור למצב הפקודה יש להקיש על מקש ה-ESC.  
במצב זה ניתן לבצע פעולות על טקסט קיים ללא עריכתו. כגון מחיקת שורה אחת או יותר, ביצוע Undo, מחיקת תוכן, העתקה הדבקה ועוד...  
להלן חלק מהפעולות האפשריות:
  - dd – מחיקת שורה קיימת
  - Ndd – כאשר N הוא מספר, אומר למחוק N שורות מהשורה הנוכחית והלאה.
  - D – מחיקת כל התוים מהתו הנוכחי ועד סוף השורה.
  - dG – מחיקת תוכן מהמיקום הנוכחי ועד סוף הקובץ.
  - u – ביצוע Undo לפעולה האחרונה
  - yy – ביצוע העתקה של השורה הנוכחית
  - p – ביצוע הדבקה של השורה שהועתקה לתוך השורה שבמיקום הנוכחי.
  - P – ביצוע העתקה שורה אחת מעל המיקום הנוכחי.
- **מצב עריכה** – למצב זה ניתן להגיע ממצב הפקודה על ידי הקשת אחד מהתוים הבאים:
  - i - הכנסת תוים מהתו הנוכחי.
  - a – הכנסת תוים תו אחד אחרי המיקום הנוכחי.
  - I – הכנסת תוים מתחילת השורה הנוכחית.
  - A – הכנסת תוים מסוף השורה הנוכחית.
  - o – הכנסת תוים שורה אחת מתחת המיקום הנוכחי.
  - O – הכנסת תוים שורה אחת מעל למיקום הנוכחי.
- **מצב סוף שורה** – למצב זה ניתן להגיע ממצב הפקודה על ידי הקשת התו נקודתיים (:).  
במצב זה ניתן לבצע פעולות חיפוש והחלפה בקובץ, פעולות של שמירה ויציאה, הוספת תוכן קובץ אחר לקובץ הקיים והחלפת מצבי צפיה (כגון הוספת/הסרת מספר השורה בתחילת כל שורה, הצפנת הקובץ. להלן חלק מהפעולות האפשריות:
  - w: – שמירה
  - wq!: – שמירה ויציאה
  - q!: – יציאה בלי שמירה
  - g/new/old:%: – החלפת המילה old במילה new בכל הקובץ



# Vi pocket guide

[Version 1.04, 08-05-09]



## handle files and exit

**:q** quit editor  
**:q!** force quit editor  
**:w** write to disk  
**:wq** write to disk and exit  
**:wq!** Force write to disk and exit  
**:n <file>** open another <file> for editing  
**:n #** switch between files

## Search functions

**/exp** search for exp  
**?exp** search backward for exp  
**n** repeat search forward  
**N** repeat search backward  
**:set ic** search ignore case sensitive  
**:set noic** search should be case sensitive

**:%s/OLD/NEW/g** search and replace globally the string OLD with the string NEW

## Control edit session

**:set nu** show line numbers  
**:set nonu** hide line numbers

## Encryption

**:X** ask for encryption key  
(in order to open an encrypted file use vi -x)  
please note: only null encryption key returns the file to be decrypted !

## Alternate arrow keys:

(if regular arrow keys do not work).

## Relevant switches:

**-x** Allow encryption/decryption of a file using a specific key(which should be entered interactively).  
**-r** Allow recovering of a file after an editor or system crash.

## Cursor movement

**l** move cursor right  
**h** move cursor left  
**j** move cursor down  
**k** move cursor up  
**0** move to start of line (zero)  
**\$** move to end of line  
**G** move to last line in the file

## Delete text

**x** delete a character  
**dw** delete one word  
**dd** delete one line  
**D** delete till end of line  
**d0** delete to start of line  
**dG** delete to end of file  
**dIG** delete from beginning of file to cursor

## Copy and insert text

**yy** yank 1 line  
**2yy** yank two lines  
**p** paste below cursor  
**P** paste above cursor

## Undo and Redo commands

**u** undo last change  
**U** restore current line  
**.** repeat last change

## Add/Append text

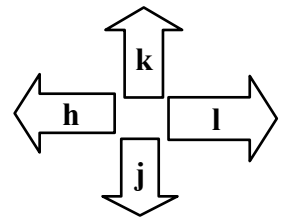
**a** append text after cursor  
**A** append text at end of line  
**i** insert before cursor  
**I** insert a at begin of line  
**5i** insert text 5 times

## Add lines

**o** add line below cursor  
**O** add line above cursor

## Change text

**s** change one character  
**cw** change one word ahead  
**r** replace one character



סיימו ב-5.12.2017

## ***The Shell - המעטפת***

נושאי לימוד:

- מהי המעטפת ואיך מגדירים הגדרות
- תוים מיוחדים במעטפת
- משתני סביבה מקומיים וגלובליים
- Alias – קיצורי פקודות
- Initialization files – אתחול

## מהי המעטפת

המעטפת היא מפענח פקודות. למעטפת יש יכולת לפענח פקודות פנימיות – אלו פקודות שמוגדרות בתוך המעטפת ופקודות חיצוניות – אלו פקודות שאינן מוגדרות בתוך המעטפת, אלא כקבצי סקריפט או קבצים בינריים הנשלחים לריצה על ידי המעטפת.

**פקודות פנימיות** – while, for, if alias, echo, history, pwd ועוד...

**פקודות חיצוניות** – יש משתנה סביבה PATH שמורה למעטפת היכן מיקום הספריות בהן לחפש פקודות חיצוניות.

לפעמים היינה קיימות פקודות שהן גם פנימיות וגם חיצוניות, המעטפת תעדיף כברירת מחדל להריץ פקודה פנימית במידה וקיימת פקודה חיצונית עם אותו שם.

כדי להריץ פקודה חיצונית יש לכתוב את הנתיב המלא של הפקודה.

ניתן ליצור פקודות חדשות על ידי הפקודה alias.

ישנו סט הגדרות שקובע את התנהגות המעטפת ניתן לראות זאת ע"י פקודה set -o

noclobber – מונע דריסת קבצים בעת יצירת קבצים חדשים

history – מאפשר שמירת היסטוריית שימוש בפקודות

emacs – דפוס עבודה של מעבד תמלילים מסוג emacs.

כדי לצפות בסט של התנהגויות הקיימות:

set -o

כדי להפעיל התנהגות משתמשים בפקודה:

set -o noclobber

כדי לכבות התנהגות משתמשים בפקודה:

set +o noclobber

פקודות פנימיות מיוחדות -

שם הפקודה	תאור הפקודה
history	מציגה היסטוריה של כל הפקודות שבוצעו
pwd	מציגה את מיקום המעטפת במערכת הקבצים
export	מציג ומאפשר הגדרת משתני סביבה הגלובליים המוגדרים במעטפת
set	מציג ומאפשר הגדרת משתני סביבה מקומיים המוגדרים

## תווים מיוחדים במעטפת

הערות	דוגמא	המשמעות	התו
		מחליף 0 תווים או יותר מכל סוג	*
		מחליף תו אחד בלבד	?
	<code>cd /export/home/haim/dir1\ /pictures ls -l my\ hello.txt</code>	המשך שורה או ביטול רווח כתו מפריד במעטפת או ביטול משמעות של תו מיוחד	\
	<code>ls [a-z] [0-9] [abcd]</code>	הגדרת קבוצה שבנויה מטווח או מתווים בודדים	[]
	<code>ls; date; cal</code>	מפריד בין פקודות בתוך שורה	;
	<code>ssh mysql1 '(cd /var/log ;cat mysqld.log)'</code>	יצירת בלוק פקודות אחד	'()'
	<code>mkdir `date +%d.%m.%Y`</code>	מחליף את פלט הפקודה כחלק מהקלט של המעטפת.	`גרש מוטה
	<code>mv hello.txt 'My hello.txt'</code>	מתייחס לקבוצת תווים כמחרוזת אחת (גם אם יש רווחים באמצע).	'גרש ישר
	<code>cd ~</code>	מייצג את ספריית הבית	~ טילדה

	ls -l   more הפלט של הפקודה ls הופך להיות קלט של הפקודה more	מאפשר להעביר קלט של פקודה אחת להיות פלט של פקודה אחרת	pipe
	# this text will not be performed.	מסמן את התוים הבאים אחריו כהערה	# רשתית

סיימנו בתאריך 19.12.2017

## Redirection – 1 Pipe

במערכות הפעלה פתוחות ניתן להעביר פלט וקלט בין פקודות ובין פקודות לקבצים. לכל תהליך שרץ ישנם 3 קבצים - קלט, פלט ושגיאה סטנדרטיים.

- קובץ קלט 0 – stdin – standard input
- קובץ פלט 1 – stdout – standard output
- קובץ שגיאה 2 – stderr – standard error

### | - Pipe

התו - | מאפשר להעביר פלט של פקודה להיות קלט של הפקודה מימין לה. לדוגמא:

מאפשר לראות פלט של הפקודה ls עם צפיה בתוים מיוחדים

```
ls -la | cat -vet
```

מאפשר לראות פלט של הפקודה ls עם דפדוף

```
ls -la | more
```

### >< Redirection

התו < מאפשר להעביר קובץ כקלט של פקודה לדוגמא:

מאפשר לשלוח מייל כאשר תוכן המייל הינו קלט מקובץ

```
mail user@example.com < hello.txt
```

התו > מאפשר להעביר פלט של הפקודה ls לתוך הקובץ output.txt

```
ls -la > output.txt
```

הפקודה tee מאפשרת לשלב בין העברת פלט של פקודה לקובץ וכקלט לפקודה אחרת

```
ls -l | tee ls.txt | more
```

## משתני סביבה

משתנים שנמצאים במעטפת ומוגדרים לאורך חיי תהליך המעטפת שם משתנה הינו case sensitive כלומר רגיש לאותיות קטנות או גדולות. כדי לראות את משתני הסביבה נבצע את הפקודות: set-l env משתני סביבה בדר"כ יופיעו עם אותיות גדולות (Capital letters).

## משתני סביבה מקומיים

משתנה סביבה מקומי הינו משתנה המכיל ערך במעטפת. טווח ההכרה של משתנה סביבה מקומי הינו רק במעטפת בה הוא הוגדר. (ולא עבור תהליכים בנים של המעטפת).

כדי לצפות במשתני סביבה מקומיים

set  
 כדי להגדיר משתנה סביבה מקומי  
 MYDIR='content of the variable'

כדי להכניס פלט של הפקודה date לתוך משתנה -

MYDIR=`date +%d.%m.%Y`

כדי לראות תוכן של משתנה סביבה מקומי

echo \$MYDIR

echo \$PATH

PATH=\$PATH:/usr/sfw/bin:.

PATH=./usr/sfw/bin:\$PATH

טבלת משתנים ידועים

דוגמא	תיאור המשתנה	המשתנה
	נתיב חיפוש לפקודות חיצוניות התו : מפריד בין נתיבים	PATH
<pre>export PS1='\u@\h: \w] #' שם המשתמש \u - שם המחשב \h - שם הספרייה \w -</pre>	מחרוזת תחילית לשורת פקודה (Prompt)	PS1

## סיימנו ב - 9.1.2018

אם בטעות מחקנו את המשתנה סביבה PATH ניתן להריץ פקודות על ידי מתן מסלול מלא של הפקודה.

לדוגמא הרצת הפקודה :ls  
 /usr/bin/ls

תרגול:

1. צרו ספרייה בשם bin בתוך ספריית הבית שלכם.  
הפקודה:

2. העתיקו את הפקודה /bin/usr/ping לתוך הספרייה bin שייצרתם.  
הפקודה:

3. הוסיפו את הספרייה ~/bin להיות הספרייה הראשונה במשתנה הסביבה PATH מבלי למחוק את תוכן PATH שהיה קודם.  
הפקודה:

## **משתני סביבה גלובליים**

משתנה סביבה גלובלי הנו משתנה סביבה מקומי שאנו מייצאים החוצה על ידי הפקודה  
.export

כדי להפוך משתנה סביבה מקומי לגלובלי נריץ את הפקודה:  
export MYDIR  
export MYVAR=hello

טווח ההכרה של משתנה סביבה גלובלי הנו למעטפת ולכל התהליכים הבנים שלה.  
(התהליכים בנים יורשים את כל המשתנים הגלובליים שקיימים במעטפת).

## **Alias – יצירת פקודות פנימיות חדשות**

alias מאפשרת ליצור פקודה חדשה שהיא אוסף של פקודה חיצונית או פנימית אחת או יותר.  
יצירת פקודות חדשות על ידי הפקודה alias

כדי לצפות איזה פקודות מוחלפות על ידי הפקודה alias נכתוב:  
alias

ניצור פקודה ll המריצה ls -l על ידי alias לדוגמא:  
יש לתחום פקודות המכילות רווח בגרש ישר.

```
alias ll='ls -l'
```

ניצור פקודה rm שמריצה rm -i לדוגמא:

```
alias rm='rm -i'
```



## קבצי אתחול

שם המעטפת	שם הקובץ אתחול	מתי יקרא הקובץ
Bash	~/.bashrc	בעת פתיחת מעטפת חדשה במנהל החלונות.
	~/.bash_profile	בעת ביצוע התחברות מרחוק למערכת.

דוגמא לקובץ אתחול:

```
vi .bashrc
```

```
export PS1='\u@\h: \w] #'  
export MAIL=haim.tzadok@grigale.com  
alias rm='rm -i'  
alias dir=ls
```

טעינת קבצי אתחול  
את קובץ האתחול המעטפת טוענת בעת עלייה של המעטפת.  
ניתן לטעון מעטפת על ידי שימוש בפקודה source או על ידי שימוש בפקודה נקודה (.)

דוגמא לטעינת קובץ אתחול:

```
. .bashrc
```

או

```
source .bashrc
```