

## הדגמת CMD ותרגיל

בשיעורים קודמים הכרנו קצת את חלון ה shell של לינוקס. החלון מאפשר לעשות כמעט כל פעולה שאנחנו רוצים במערכת ההפעלה – דרך שורת פקודה אינטראקטיבית (הגדרה נוחה למקרה זה – יכולת ביצוע פעולות ללא שימוש בעכבר...). תוכנות רבות שלהן ממשק גרפי (GUI – Graphical User Interface) מספקות גם ממשק טקסטואלי (CLI – Command Line Interface), על מנת לאפשר עבודה מהירה איתן (והתממשקות עם תוכנות אחרות).

גם ב Windows קיים כלי דומה (אליו נחשפתם בלימודי רשתות). כלי זה נקרא ה- Command Prompt או שורת הפקודה. אנחנו נתייחס אלי בשם הקיצור המוכר – CMD.

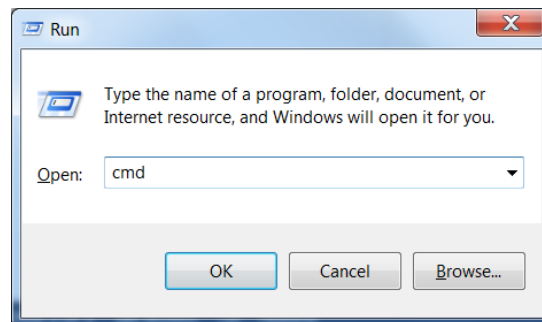
לפני שנתחיל – נציין רק שבשנים האחרונות Microsoft הוציאו כלי חדש ל Windows הנועד לשמש כממשק טקסטואלי והוא PowerShell. הכלי מכיל ממשק עשיר יותר ופקודות מגוונות, אך לא נעסוק בו כרגע.

### רקע

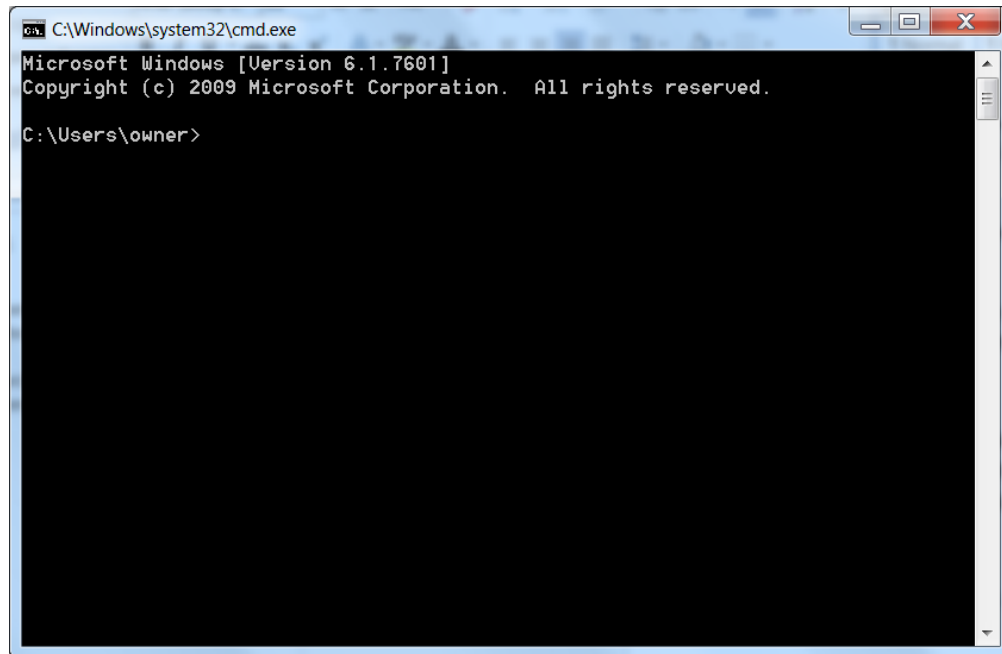
הבסיס לממשק ה CMD הולך שנים אחורה – לתקופת מערכת ההפעלה MS-DOS. מערכת הפעלה זו הגיעה ללא GUI וממשק העבודה איתה היה טקסטואלי לחלוטין. רק באמצע שנות ה 90 התווסף ממשק גרפי (במעה"פ Windows 3.11) שרץ מעל MS-DOS. בהמשך, עם יציאת Windows NT – הפך ה CMD לאפליקציה של מערכת ההפעלה, והממשק הגרפי הפך לממשק העיקרי.

### פתיחת חלון CMD

הדרך הכי נוחה לפתיחת שורת הפקודה, היא שימוש בשורת ההרצאה – (ניתן לפתוח אותה באמצעות צירוף המקשים WinKey+R), ואז פתיחת cmd



בשלב זה נפתח לנו החלון השחור האהוב



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\owner>
```

כעת, אנו יכולים להריץ פקודות שונות! למתעניינים - ניתן למצוא רשימת פקודות בקישור [הבא](#).

נתחיל בהרצת הפקודה הבסיסית `help`. הפקודה תיתן לנו רשימה (חלקית) של הפקודות הזמינות לנו.

**משימה 1:** הריצו את הפקודות הבאות ושימו לב לפלט שלהן: `HELP`, `DIR`, `TASKLIST`, `SYSTEMINFO`. הסבירו מה עושה כל פקודה.

### פקודות מובנות

**משימה 2:** היכנסו לתיקיה של `C:\Windows\System32`. עבור הפקודות שהרצנו קודם חפשו קובץ הרצה תואם (שימו לב שהסיומת `exe` מסמלת קובץ הרצה – קיצור ל `executable`):

- `tasklist.exe` -
- `systeminfo.exe` -
- `dir.exe` -
- `help.exe` -

האם מצאתם את כולם? מה לא מצאתם?

כעת השתמשו בפקודה *where* בצורה הבאה:

```
where ipconfig
```

מה קיבלתם? מה משמעות הפלט?

**מה ניתן ללמוד ממה שראינו?**

ראינו כי ה"פקודות" אותן אנו מריצים ב CMD ממומשות למעשה כתוכנות נפרדות (עבור כמעט כל אחת היה ניתן למצוא קובץ הרצה תואם עם סיומת exe)

כעת נסו להריץ את הפקודה הבאה:

```
where where
```

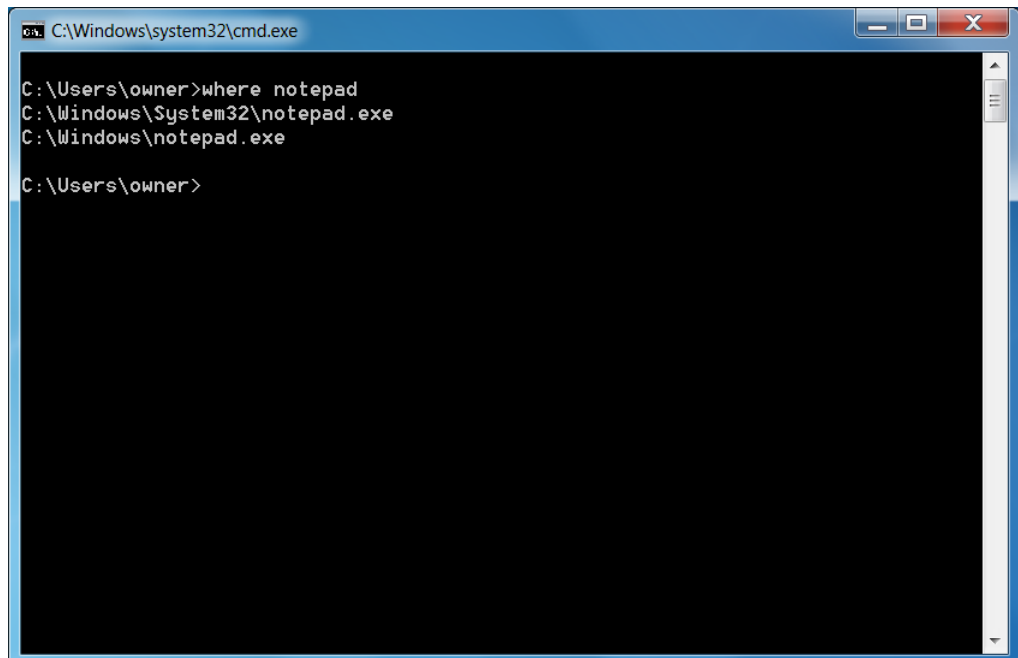
ניתן לראות שגם הפקודה *where* היא תוכנה עצמאית לכל דבר.

**אז למה לא מצאנו למשל את "dir.exe" ?**

ישנן מספר פקודות מובנות שאינן ממומשות כתוכנות חיצוניות. פקודות אלה כוללות בין השאר את *cd, dir, move, copy, del* ועוד. באמצעות פקודת *help* ניתן לקבל רשימה של הפקודות המובנות, אך יש לשים לב כי רשימה זו משלבת בתוכה גם חלק מהפקודות ה"חיצוניות".

## סדר חיפוש תוכנות חיצוניות ומשתנה הסביבה PATH

נתחיל בהרצת הפקודה: *where notepad*



```
C:\Windows\system32\cmd.exe
C:\Users\owner>where notepad
C:\Windows\System32\notepad.exe
C:\Windows\notepad.exe
C:\Users\owner>
```

נשים לב כי קיבלנו הפעם 2 שורות פלט עבור הפקודה. למעשה – נמצאו שני קבצי הרצה עם השם *notepad*.

כעת – נבצע ניסוי קטן. היכנסו לתיקייה C:\Windows\System32 והעתיקו את notepad.exe לתוך שולחן העבודה. הריצו שנית את הפקודה *where notepad* – האם הפלט השתנה?

עתה נשאלת השאלה – באילו תיקיות בוחר *where* לחפש את התוכנות? (שימו לב ש-*where* מדמה את החיפוש ש-*cmd* עושה ברגע שאנו מקישים פקודה שאינה מובנית). לשם כך קיים משתנה סביבה בשם PATH המכיל את רשימת התיקיות בהן יש לחפש קבצי הרצה. משתנה זה מכיל מחרוזת ארוכה של תיקיות המופרדות זו מזו באמצעות ";" (נקודה פסיק).

להצגת ערך משתנה הסביבה נשתמש בפקודה המוכרת לנו *set* בצורה הבאה:

```
set PATH
```

נשים לב שקיבלנו רשימה ארוכה של תיקיות שונות ובפרט C:\Windows\System32 - ו-C:\Windows\ שבהן מצאנו את notepad.exe. במידה ונקליד עכשיו *notepad* ב CMD - סדר החיפוש של התוכנה יהיה לפי סדר התיקיות במשתנה PATH, והקובץ ההרצה הראשון שימצא תואם – יורץ.

- להרחבה - קראו על משתנה הסביבה PATHEXT והמשמעות שלו (הדבר יעזור בתרגיל המצורף)

PATH, יחד עם מספר משתנים אחרים, הם משתנים הנוצרים באופן אוטומטי לכל תהליך. במידה ונרצה לשנות ערך של משתנה מסוג זה – לא נוכל להסתפק בפקודה *set VARIABLE=value* אותה למדנו שכן היא תשנה את ערך המשתנה רק בתהליך שלנו. אם ניצור חלון CMD חדש – ערך המשתנה יהיה זהה לערכו כשיצרנו את חלון ה CMD הראשון.

ניתן לשנות את הערכים של משתנים אלו דרך הגדרות המערכת. הסבר להגעה להגדרות אלה ניתן למצוא למשל [כאן](#).

## תרגיל

בתרגיל זה נממש את פקודת ls ל CMD (פקודה ls בלינוקס פועלת בדומה לפקודה dir ב CMD).

הפקודה צריכה להחזיר על רשימת הקבצים והתיקיות בתיקה שממנה היא הורצה.

**בנוסף:** הוספת פרמטר של תיקיה. במידה ופרמטר זה אינו ריק – תוחזר רשימת הקבצים הנמצאים בתיקה שצוינה. (כלומר, הרצאה של הפקודה `ls c:\` תחזיר את רשימת הקבצים והתיקיות הנמצאים בתיקיית האב של כונן C)

המימוש יתבצע בצורה הבאה:

- נממש בפיתון את `ls.py` שיציג את רשימת הקבצים בתיקה הנוכחית.
- מכיוון שקבצי פיתון אינם נחשבים לקבצי הרצה, נצטרך לייצר קובץ הרצה. קובץ זה יהיה קובץ Batch פשוט שיריץ באמצעות פיתון את `ls.py` (ניתן לחפש על קבצי batch באינטרנט).
  - קובץ זה אמור להכיל שורה אחת בלבד – הדומה לשורה אותה היינו כותבים ב CMD כדי להריץ את `ls`.
  - שימו לב שבשביל החלק של הבונוס, נצטרך לתמוך גם בקבלת פרמטרים לקובץ ה batch דרך ה CMD, כך שהם יועברו לקוד הפייתון שכתבנו.
- לסיום, נדאג ששני הקבצים אותם יצרנו יהיו בתיקה אחת, ונוסיף את תיקיה זו למשתנה ה PATH.
  - היזהרו כשאתם עורכים את משתנה זה. שיבוש שלו עלול לפגוע בפעולת תוכנות במחשב.
  - הוסיפו את התיקייה שלכם בסוף רשימת התיקיות, על ידי תוספת של נקודה-פסיק ואחריו ציון הנתיב המלא לתיקה.

בהצלחה!