**DevOps קורס**

**Kickoff Meeting – 15/05/2022**

* **הסבר על השבוע השלישי** – הוצגה תוכנית הלימודים ותוכנית הלמידה של התוכנות והכלים שאותם נלמד בהמשך , השבוע מדברים על נושא Between dev and ops
* **Quizzizz** – אפליקציית שאלונים עליה אנחנו נדרשים לענות עד ליום שישי הקרוב , דובר על התפלגות ציונים , נראה כי הממוצע ירד
* **פרויקט** – נראה שהיו הרבה הבדלים באיך להגיש את הפרויקט , בגדול בתוך הקוד עצמו נדרשנו להשתמש בפונקציות שחוסכות זמן ריצה (**Built In**) , כמובן גם הגשה של קוד נקי (לצערי אין לי **BEST PRACTICE** אלא מה שאני חושב שיעיל)

**מפגש חמישי – 30.05.2022**

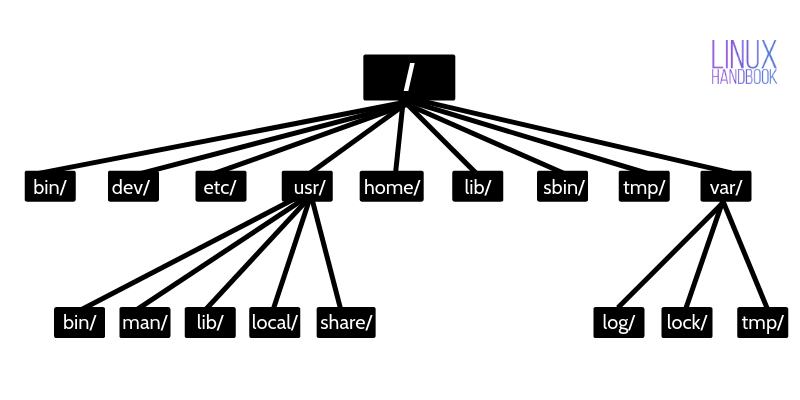
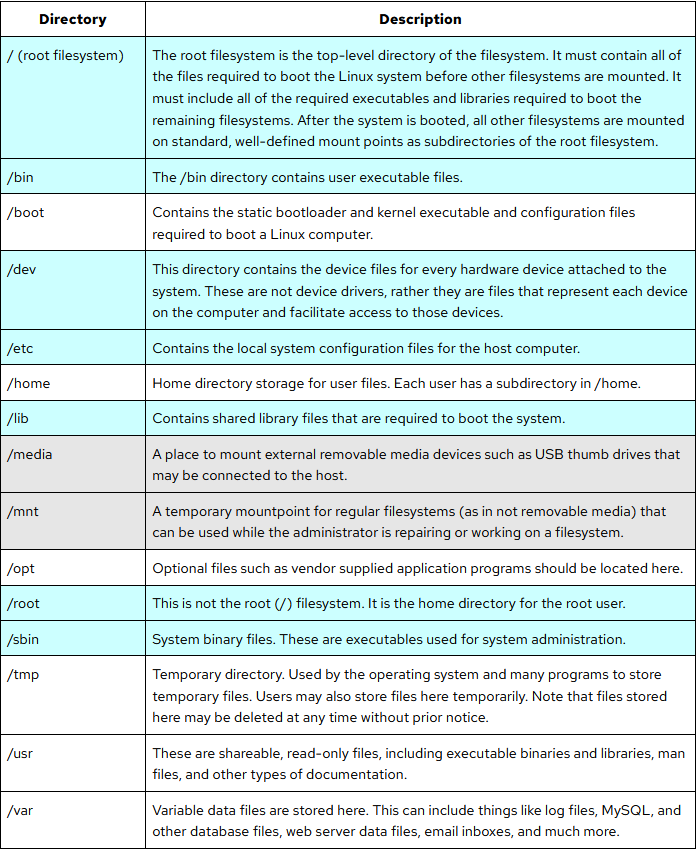
* **אירוח אתרים** – אירוח אתר , הכוונה לאחסון קבצים הכתובים בשפות אתרים כגון **HTML , PHP ,**  אפילו גם ב **Python** , והרצה שלהם כארכיטקטורה מסודרת ומאורגנת בעזרת כלים מסוג **NGINX , APACHE2 , IIS (Internet Information Services)**
* **תהליכים\Process** – כחלק ממערכת ההפעלה , מאחורי כל אפליקציה שרצה במערכת ההפעלה קיים תהליך שרץ ברקע , עם מי שמריץ את התהליך ופרטים נוספים כמו זיכרון שימוש ואחוזי שימוש במשאבי עיבוד ודיסק.
* **קבצים** – קבצים במערכת ההפעלה הם בדרך כלל בינארים (**.EXE או Executable** במערכת ההפעלה **Windows** או קובץ ללא סיומת במערכת ההפעלה **Linux** , בדרך כלל במערכת ההפעלה לינוקס קבצים ללא סיומת יכולים להיות גם סקריפטים וגם קבצים בינארים , הכל תלוי בתוכן הקובץ ולא אפשרי לדעת ללא פתיחת הקובץ או הדפסת תוכן הקובץ על המסך)

הקבצים כתובים בשפת מכונה , הקבצים הם בדרך כלל תוצר של איזשהו תהליך פיתוחי.  
הקבצים הללו

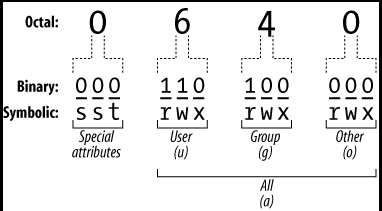
* **Package Management** – ניהול גרסאות אפליקציה \ ספריות , הדוגמא הכי טובה לניהול ספריות או אפליקציות היא מערכת **apt-get** באובונטו , היא מאפשרת התקנה של אפליקציות על ידי משיכה שלהם ממאגר מרוחק (**Repository**) של מתחזקי המערכת ההפעלה , דבר דומה קיים בתכנות לשפות כמו **Python** וגם **Node** , ניתן להשתמש ב**Python PIP** או **NPM** כדי להוריד ספריות נוספות שניתן יהיה לעבוד עמם בקוד.

**HOWEVER** – ייתכן וכחלק מהתהליך הפיתוחי ייגרמו מצבים של תלות , או **Dependencies**. זאת אומרת , מצבים בהם אפליקציה לא יכולה לרוץ במערכת ההפעלה ללא התקנה של ספריות או קמפול של הספריות האלה בתוך הקוד , במשחקי מחשב התופעה נפוצה מאוד , המשתמש נדרש להוריד תוכנה בשם **DirectX** כדי שהמשחק יוכל לרוץ (אפליקציה) ומבחינת תהליכי פיתוח , דוגמא לכך יכולה להיות התקנה של ספריה על מנת שהקוד ירוץ , ב**Python** ניתן להגדיר קובץ מסוג **requirements.txt** עם תוכן של שמות הספריות הנדרשות על מנת שהקוד יוריד את הספריות לבד וירוץ.

* **מערכת ההפעלה LINUX** – מערכת ההפעלה לינוקס נוצרה על ידי **Linus Torvalds** בליווי **Richard Stallman** , מבחינה היסטורית **Linux** לא היתה ידועה בשמה אז אלא בתור מערכת הפעלה בשם **Unix** , לאחר פיתוח מערכת ההפעלה **Unix** שנבנתה בתצורת **GNU** (**GNU is not unix**) וכמובן עם רישיון הטוען כי אין להשתמש במערכת ההפעלה או בקוד הפתוח שלה בתור מוצר פרטי אלא "הכל שייך לכולם" , נלקח ה **Kernel** של מערכת ההפעלה של **Unix** ובעזרתו נבנו מערכות הפעלה שונות , אחת מהן היא **FreeBSD** (מערכת הפעלה **Unix** מבוססת **Kernel** של לינוקס) של אוניברסיטת ברקלי בקליפורניה , מטרתה היתה לימוד ומחקר ופיתוח על בסיס **Unix** , בעבר מערכת ההפעלה של **Microsoft** נקראה **MS-DOS** או **DOS** בקיצור , בזכות הפיתוח של ה**Kernel** של **Linux** , **Microsoft** התחילה להשתמש ב**Kernel** של **Linux** בתוך מערכת ההפעלה שלה , מה שאיפשר לה להתקדם טכנולוגית ולשחרר את מערכת ההפעלה **Windows 1.0** שעם הזמן השתדרגה לגרסאות כמו **Windows 95 , Windows XP** וכו...

**כיצד בנויה מערכת ההפעלה לינוקס** – מערכת ההפעלה **Linux** בנויה בתצורת עץ , אנחנו מכירים נתיבים ב **Windows** בשמם המלא , קודם כל יוצא הכונן -> לאחר מכן יוצג נתיב בכונן וכו וכו , הכוונה -> **C:\Users\Ben\Desktop\\*Some Directory\*** , ב **Linux** הכל עובד קצת אחרת , יש לשים לב כי ב**Windows** הסימון הוא סלאש הפוך (**Backslash , \**) וב**Linux** הסימון הוא סלאש רגיל -> / , ולכן לעיתים כאשר מפתחים מוצרים על **Windows** או על **Linux** , יש לשים לב לצורת כתיבה לנתיבים.  
ב**Linux** סימון נתיבים מתחיל כך לדוגמא -> **/home/Ben/Desktop/\*Some Directory\***  
  
באופן כללי , תצורת עץ הספריות במערכת ההפעלה **Linux** מתבטא כך.  
  
  
  
**נתיב ה - /** - תחת נתיב זה בעצם "מתחילה" כל מערכת ההפעלה (לא ה**Kernel**) , נתיב זה נקרא **root** (קיים גם משתמש במערכת בשם **root** עם הרשאות גבוהות , בהמשך הסיכום ידובר על המשתמש הזה) , נתיב זה מכיל תחתיו נתיבים נוספים אשר לכל אחד מהם תפקיד מיוחד במערכת ההפעלה.  
  
**נתיב /bin** – בנתיב זה קיימים קבצים בינארים של המערכת , בינהם פונקציות כמו **ls** או **mount** אשר נדרשות לטובת עליית מערכת ההפעלה , או לשימוש בתיקון המערכת (**Kernel Panic** וכו)  
  
**נתיב /boot** – מכיל קבצים והגדרות הנדרשים לריצת מערכת ההפעלה   
  
**נתיב /dev** – בנתיב זה קיימים קבצים של ההתקנים המחוברים וממופים אל מערכת ההפעלה , בנתיב זה ניתן למצוא התקנים כמו דיסקים , התקני שמע התקני צריבה וכו...  
  
**נתיב /etc** - מכיל קבצים שבהם קיימים הגדרות למערכת ההפעלה , בדרך כלל קיים הבדל בקבצים בנתיב זה בין אם מערכת ההפעלה מותקנת על **Container** או על מכונה וירטואלית.  
  
**נתיב /home** – בנתיב זה קיימים הגדרות עבור המשתמשים במערכת ההפעלה , נתיב דומה ניתן למצוא ב**Windows** תחת (לרוב) **C:\Users**   
תחת נתיב זה קיימים נתיבים כמו **/home/\*user\*/Desktop** או **/home/\*user\*/Downloads** וכו.  
  
**נתיב /lib** – מכיל קבצים וספריות הנדרשים לטובת עליית מערכת ההפעלה , בשונה מנתיב **/boot** , נתיב **/lib** מכיל קיצור דרך (הפניה) לנתיב **/usr** (נתיב הגדרות משתמש , לא מדובר על **/home**) ותחתיו קבצים ונתיבים עבור מערכות שונות במערכת ההפעלה , כמו מערכת ניהול האפליקציות **apt** או **yum** (תלוי **Distro**)

**נתיב /media** – בנתיב זה קיימים קבצים או נתיבים (גם הפניות) עבור מכשירים נתיקים במערכת ההפעלה , כמו **DOK (**דיסקונקי) או דיסקים קשיחים נתיקים  
  
**נתיב /mnt** – תחת נתיב זה קיימים הפניות ונתיבים אל דיסקים או ספריות מקושרות במערכת ההפעלה , תחת **/mnt** ניתן למצוא את הדיסק שעליו מותקנת מערכת ההפעלה , ניתן לראות פרטים נוספים על היכן מותקנת מערכת ההפעלה בעזרת הפקודה **df** (או **df -h** לטובת פורמט קריא יותר)  
  
**נתיב /opt** – תחת נתיב זה קיימים קבצים ואפליקציות אשר אינן נדרשות לקיום מערכת ההפעלה , בין היתר אפליקציות שהמשתמש התקין ידנית בנתיב זה.  
  
**נתיב /root** – כפי שאמרתי קודם , הנתיב הזה אינו מכיל את כל התיקיות הנ"ל אלא הוא עצמו משמש כנתיב ה"בית" של המשתמש **root** במערכת ההפעלה **Linux** , בדרך כלל הנתיב הזה מגיע ריק לאחר התקנה.  
  
**נתיב /sbin** – בדומה לנתיב **/bin** , בנתיב זה נמצאים קבצים וספריות עבור מערכת ההפעלה ברמת הרשאות גבוהות (**Super User**) , בנתיב זה קיימים גם אפליקציות וקבצים אשר משתמש ללא הרשאות במערכת (בדרך כלל **sudo**) לא יכול להריץ.  
  
**נתיב /tmp** – בנתיב זה נמצאים קבצים זמניים של המערכת , לוגים של הורדות או כל מיני ג'אנק שמערכת ההפעלה כותבת לנתיב הנ"ל , לעיתים גם הורדות.  
הדבר קיים גם ב**Windows** וגם ב**Linux**  
  
**נתיב /usr** – בנתיב זה קיימים קבצים ואפליקציות במצב קריאה בלבד , כגון **man** או מסמכים רלוונטים נוספים.  
  
**נתיב /var** – בנתיב זה נמצאים לוגים של המערכת כגון **Syslog** או לוגים של אפליקציות צד ג' כמו **MySQL**

* **הרשאות במערכת ההפעלה Linux** – במערכת ההפעלה **Linux** ניהול הרשאות מתבצע על ידי הפקודות **Chown** ו **Chmod** , מטרת הפקודה **Chown** היא החלפת בעלות על קובץ מסוים , לדוגמא , לעיתים אנו משתמשים בפקודה **sudo** לטובת עריכה או יצירה של קובץ , מה שמשאיר את הקובץ עם בעלות של משתמש **root** במערכת , לעיתים הדבר מונע מאיתנו לכתוב או לערוך את הקובץ הזה.  
    
  מקרה נוסף של **sudo** , לעיתים אנחנו נדרשים להחליף הרשאות עבור קובץ , לדוגמא איננו מעוניינים שהקובץ יוכל לרוץ (במידה ומדובר בסקריפט מבחינת אבטחת מידע) או במידה ואנו פשוט מעוניינים לשנות הרשאות לקובץ כדי שהוא יכול לרוץ רק על ידי משתמשים בקבוצה ספציפית.  
    
    
    
  כפי שניתן לראות באיור , קובץ מחולק ל4 קבוצות הרשאות  
    
  בקבוצה הראשונה מוגדרים הרשאות מיוחדות – זאת אומרת **chmod 200** (משתמשים במספרים בדרך כלל בפקודה זו , ניתן לבצע גם קיצורים בעזרת **u + x** וכו)  
    
  בקבוצה השניה מוגדרים הרשאות עבור משתמש , לדוגמא **Ben** , בדרך כלל משתמש שמעוניין לפתוח או לערוך קובץ לא נדרש לשנות את הבעלים של הקובץ (**Chown**) אלא נדרש להגדיר לו הרשאה בעזרת **chmod** ולשייך את המשתמש לקבוצה ספציפית , לכל מקרה באופן כללי משתמש יוכל לצפות בקובץ אך לא לערוך אותו.  
    
  בקבוצה השלישית מוגדרים הרשאות עבור קבוצות , כ – **Best Practice** בעבודה במערכת הפעלה **Linux** מוסיפים משתמשים אל קבוצות ומגדירים הרשאות לקבצים על פי קבוצות , כך שמשתמש שאינו נמצא בקבוצה מסוימת לא יוכל לערוך קבצים שאינו נדרש לערוך , כמו מסמכים של עובדים נוספים או אפילו קבצי מערכת.  
    
  בקבוצה הרביעית אין משמעות בדרך כלל , היות ומבצעים ניהול הרשאות על ידי קבוצות.