

תרגיל מסכם - עבודה בסביבת לינוקס 24א

אופן הגשת התרגיל:

- בזוגות או יחיד
- לעלות פתרונות ותוצרי קוד למודל
- סרטון הרצה של הסקריפט וקבלת תוצרים

שאלה 1: GIT

בחברת DEVBOX לכל קבוצת פיתוח קיים קובץ אקסל (csv). המכיל משימות שיש לבצע לספרינט הקרוב כל מפתח מקבל אחת לשבועיים אקסל במייל עם הבאגים שהוא צריך לפתור דוגמה לאקסל:

לדוגמא:

BugId	Description	branch	Developer Name	Bug Priority	GITHUB URL
234	Alert should be in red color	BR_1	g.choen	3	...

על מנת לייעל את העבודה בכתיבת תיאור COMMIT ים יותר מפורטים, קיבלת משימה לכתוב סקריפט בשם commit.sh המוסיף את נתוני האקסל ואת הזמן הנוכחי באופן אוטומטי לתיאור ה-COMMIT,

במטרה שכל הפתחים ישתמשו בסקריפט זה (:

יש לכתוב סקריפט בשם **commit.sh** כך שבסיום תיקון הבאג בקוד, המפתח יריץ סקריפט זה

הסקריפט יבדוק מה הענף הנוכחי יחפש באקסל את שם ענף זה

ויבצע את הסטייג' הקומיט וה-PUSH לגיטאהב

כך שתיאור ה-COMMIT בגיט יהיה מבוסס על נתוני האקסל לפי הפורמט הבא:

BugID:CurrrntDateTime:Branch Name:DevName:Priority:Excel Description

- הוסף פרמטר אופציונאלי להערה נוספת עי המפתח

commit.sh {Developer Additional Description}

כך שתיאור ה-COMMIT יהיה בפורמט הבא

BugID:CurrrntDateTime:Branch Name:DevName:Priority:Excel Description:Dev Description

בנה סביבה מתאימה משלך להדגמת הסקריפט (יצירת CSV, הקמת גיטאהב, יצירת ענף, קוד לקובץ לביצוע COMMIT) והרץ את הסקריפט על 2 ענפים נפרדים

ולידציות:

- קובץ האקסל צריך להיות צמוד לסקריפט הרץ
- יש לבדוק קיימות של קובץ האקסל ובדיקת פרמטרים של הסקריפט
- לכתוב אם הייתה שגיאה ב-PUSH

שאלה 2: הורדת תמונות משתמשים לפי נתוני API

<https://regres.in/api/users/{id}> הינו API המאפשר לקבל רשימה של משתמשים ומציאת משתמש לפי id, יש לכתוב סקריפט המקבל 2 פרמטרים:

- שם של תקייה שלתוכה יועתקו התמונות
- מזהה (id) של משתמש אחד או יותר

הסקריפט יוריד את התמונות של המשתמשים לתקייה , דוגמת הרצה:

./UserImageDownloader UsersImages 1 5 8

בדוגמא זו תיווצר תקייה בשם UsersImages ולתוכה יועתקו שלוש תמונות של משתמשים 8,5,1 לפי הכתובת שמצויינות בשדה ה avatar שלהם

בתרגיל זה רצוי להשתמש בספרייה כמו jq: <https://www.baeldung.com/linux/jq-command-json>

דרישות:

- (1) יש להתייחס לכתובת התמונה כך שאיננה ידועה מראש, ויש לקרוא את הכתובת משדה ה avatar של ה-JSON
 - (2) התיקייה תהיה צמודה ל SCRIPT הרץ
 - (3) שם הקובץ יהיה {Id}_{Name}_{LastName}.jpg
 - (4) במידה ויש קבצי עזר נוספים שנוצרים יש להוסיף אותם צמוד לקובץ הסקריפט
 - (5) יש להוסיף קובץ לוג הצמוד לסקריפט הרץ
- יש לכתוב לסוף קובץ הלוג בעיות ולידציה , או בעיות הרצה אם קיימות וכמו כן גם את צעדי הסקריפט העיקריים שמבוצעים
- שם משתמש
 - שם הענף הנוכחי של GIT
 - זמן פעולה (כולל nano sec)
 - פרטי הפעולה

ולידציות:

- יש לבדוק נכונות של פרמטרים לסקריפט: שם של תקייה ולפחות id של משתמש אחד
- במידה ולא קיים שם משתמש ב API יש לכתוב ללוג הודעה מתאימה
- ניתן להשתמש בכל ספרייה (CURL,JQ) או אחרות , במידה ואחת מהספריות הלינוקס שמתשמשים בהם לא מותקנות יש להודיע ללוג ולצאת

שאלה 3: הרצת קוד Python

בתרגיל זה יש לכתוב סקריפט המאפשר להוריד סרטונים מיוטיוב ולנהל playlist של סרטונים אלו, מצורף קובץ פייתון בשם **youtubedownloader.py** המשתמש בספריית **pytube** כדי להוריד סרטוני יוטיוב לקבצי mp3 ו mp4

א) הרץ את קוד הפייתון ושליחת פרמטרים מהטרמינל

- התקן את ספריית **pytube** של פייתון והרץ את הקוד על מנת להוריד קבצי וידאו ואודיו לפי כתובות יוטיוב
- שנה/עדכן את הקוד פייתון כך שניתן יהיה להריץ אותו מהטרמינל עי שליחת הפרמטרים הבאים
`url, folderpath, filename, isaudio, resolution`
לדוגמא:

youtubedownloader.py `https://www.youtube.com/watch?v=IJYn09tuPw4` ./Eurovision24
EdenGolan_Hurricane.mp4 false 720p

ב) פקודה להורדת סרטון לתקיית **playlist**

כתוב `bashscript` המקבל את הדגלים והפרמטרים הבאים ומפעיל את קוד הפייתון :

`./youtuber.sh --url {youtube url} --filename {file name} --folderPath {folder path} --resolution {resolution}`

- יש להריץ את קוד הפייתון בסביבה סגורה **venv**, הסקריפט יאתחל סביבה חדשה, אם לא קיימת ויתקין לסביבה זו את ספריית הפייתון הדרושה (**pytube**), לפי קובץ **requirements.txt**.
- אם שם הקובץ **filename** מסתיים ב **mp3** זהו קובץ אודיו ואם הוא מסתיים ב **mp4** זהו קובץ וידאו, במידה והקובץ הוא אודיו אין צורך בדגל ה `resolution`
- במידה ומסלול התקייה **folderPath** לא קיים יש ליצור תקייה זו
- שם ה- **playlist** יהיה לפי שם התקייה הסופית במסלול ה-**folderPath** בה יישמרו כל הקבצים שהורדו
- ניתן להוריד וידאו או אודיו לכל מסלול בלינוקס לכן יש לשמור את כל מסלולי התקיות ושמות ה **playlist** בהתאמה בקובץ טקסט-עזר המשמש כאינדקס לכל ה **playlist** קובץ זה יהיה צמוד לסקריפט
- אם בהורדת קובץ ה **playlist** כבר קיים אין צורך להוסיף אותו לאינדקס רק לתקיית ה **playlist**
- הרץ את קוד הפייתון כתהליך רקע ב **BASHSCRIPT** והדפסת אנימצית התקדמות שתבחר כל עוד ה **PROCESS** של הפייתון חי ועדיין מתבצעת ההורדה [demo](#)
- בדוק תקינות וקיימות פרמטרים והודע למשתמש על בעיות
- תמוך במנגנון קיצור דגלים למשל **-url** יהיה **-u**
- הוסף דרך הסקריפט מספר קבצי מדיה ב **playlist** שונים

ג) פקודה לצפייה בסרטונים

על מנת לנגן את קבצי המדיה של playlist מסויים ברצף, יש צורך לייצר קובץ מסוג **m3u**, קובץ זה הוא בפורמט טקסט פשוט כך שכל שורה בקובץ הינה מסלול לקובץ המדיה, קובץ מסוג זה הינו סטנדרט ליצירת playlist, ברוב הנגנים

ניתן להשתמש בספריית לינוקס כדוגמת **mpv** או **mplayer** המאפשרות לצפות בקבצי מדיה דרך פקודות הטרמינל בלבד, לדוגמא הפקודה הבאה תפתח נגן וידאו ישירות מהטרמינל:

mpv ./chilout.m3u

הוסף לסקריפט הקיים דגל בשם **--play** המאפשר לקבל את שם ה playlist ולבצע ניגון ברצף של הקבצים ברשימה זו:

. /youtuber -play {playlist name}

- התקן את **mpv** כחבילת לינוקס
- הוסף לסקריפט דגל בשם **play** המקבל שם של playlist
- הסקריפט יחפש באינדקס את שם ה playlist, ואת מסלול התקיה המתאים ויודא שתקיה זו קיימת ויש לה קבצי מדיה מתאימים
- לפני כל ניגון של הקבצים, הסקריפט ייצר באופן דינאמי קובץ חדש בפורמט הבא: **{name of the playlist}_playlist.m3u**
- ובו רשימה של כל קבצי המדיה כך שכל שורה תציג את המסלול של קובץ המדיה
- קובץ ה **m3u** יישמר צמוד לסקריפט הרץ ואם קיים הוא יידרס
- הרחב את הפעלת הנגן עבור רשימה של playlist כך שהקובץ שיוצר יאפשר ניגון של מספר רשימות **{name of playlist 1}_{name of playlist 2}_playlist.m3u**
- בצע הפעלת הנגן

ד) פקודה להצגת playlist

. /youtuber -list {playlistname1 playlistname2 ...}

- הוסף דגל לסקריפט,
- מקבל אחד או יותר שמות של playlist
- יש להציג את שם ה Playlist מסלול תקיית ה- playlist וכל הקבצים בתקייית ה playlist
- אם ערך הדגל יהיה ריק יודפסו כל ה playlist ים והקבצים שלהם
- בדיקת נכונות פרמטרים

