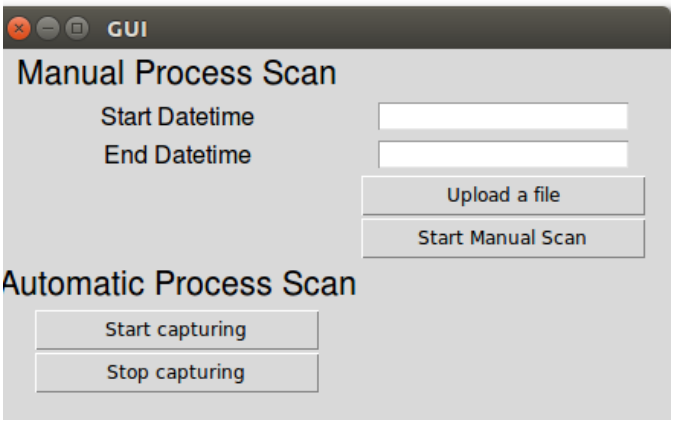
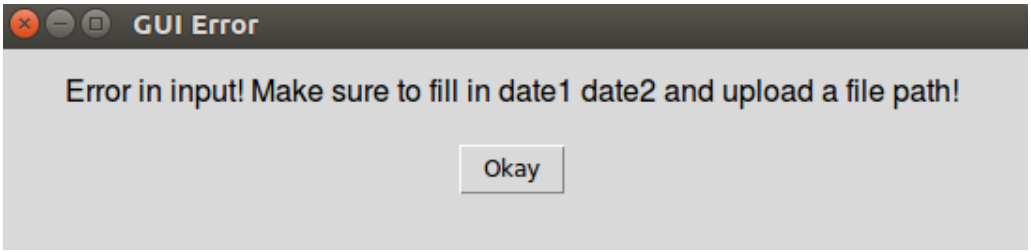
אלכס פישמן

מפרט הפרוייקט

התפריט:



הודעת השגיאה אם לא הוזנו נתונים בשדות:



מבנה התוכנית: התוכנית מורכבת מ4 מחלקות:

1. **GUI:**

- **def browseTxt()** - הפונקציה מיבאת את הבחירה של הקובץ מתפריט המשתמש.
- **def startThread()**- הפונקציה מריצה את המחלקה שאחראית על הסריקה האוטומטית ומריצה אותה בthread נוסף כדי לא לתקוע את הthread הנוכחי שמריץ את התפריט למשתמש.
- **def stopThread()**- הפונקציה עוצרת את הסריקה והורגת את הthread.
- **def popupWindow()** – הפונקציה מופעלת לשם הקפצת הודעה כאשר לא הוזנו נתונים בשדות הקלט.
- **def manualScan()** – הפונקציה בודקת האם הוזנו כל הנתונים הנדרשים בשדות הקלט ושולחת אותם להפעלת הקריאה של הקובץ הנבחר ומשווה בין שני הבלוקים שנמצאו לפי שני הזמנים הכי קרובים שנמצאים.
- **תוכנית ראשית** – מפעילה את כל הפונקציות הנ"ל ומריצה את התפריט למשתמש.

<u>ספריות שנעשה בהם שימוש:</u>

- **import Tkinter** – ספריית הגרפיקה שבעזרתה יצרתי את התפריט למשתמש.
- **import ProcessCapturing , from multiprocessing import Pool** – ספריות התהליכונים שבעזרתן יצרתי תהליכון אסינכרוני בצורה פשוטה מבלי לפקח עליו בעצמי כי הן סיפקו את כל המעטפת הנדרשת.

2. <u>ProcessCapturing</u>:

- **def create_file(name) –** הפונקציה פותחת קובץ לצורך כתיבה כותבת בו את כותרות הטבלה תוך הקפה על רווחים ופורמט מסוים שהחלטתי עליו ואת הזמן הנוכחי (תאריך ושעה).
- **def write_data_to_file(name, procs) –** הפונקציה כותבת לקובץ (שכבר נוצר) את הנתונים שאני מעבירה לה לכתיבה בפורמט שהחלטתי עליו.
- **def write_to_file(name, msg) –** הפונקציה מקבלת את שם הקובץ ואת ההודעה שצריך לכתוב בתוך הקובץ.
- **def if_procs_exist(orig,new) –** הפונקציה בודקת אם הprocess שמופיע בorig מופיע גם בnew ואם הוא לא מופיע (משמע הוא אחד חדש שנוצר או אחד ישן שנעצר) כותבים אותו לקובץ הlog.
- **def scan() -** יוצרת 2 מערכים של processes שרצים ברקע – דגימה מכרגע ודגימה מלפני 5 שניות כותבת בעזרת הפונקציות ל2 קבצי הTXT; אחד את דגימות הprocess-ים כל 5 שניות ולקובץ הLOG את השינויים שנמצאו.
  - ✓ מצורפים 2 הקבצים שנוצרו לדוגמת פלט.

<u>ספריות שנעשה בהם שימוש:</u>

- **import time –** ספריית זמן שבעזרתה עשיתי את המרווח זמנים בין דגימה לדגימה (5 שניות).
- **import datetime –** ספריית הזמן שבעזרתה כתבתי לקובץ את התאריך והשעה העדכניים – ולצורך בדיקה שמרווח ה5 שניות מתקיים.
- **import psutil –** הספרייה שבעזרתה יכולתי לקחת את דגימת הprocess-ים ולקבל את הPID, שם הprocess, הEXE שממנו הprocess הופעל (הpath) והזמן (בשניות) שבו הוא נוצר (התחיל לרוץ).

3. **ProcessLine:** יצרתי מחלקה בשם ()ProcessLine שכל שדה בה משמש לאכסון הערכים בשדות הטבלה שאני מקבלת בתור קובץ הinput.
   - **def __init__(self,sampleTime,row,inputTime_first, inputTime_second)** – הבנאי של המחלקה שמאתחל את השדות.

4. **ManualScan:**
   - **def readFile(filePath,inputTime_first, inputTime_second) –** הפונקציה קוראת את השדות מהקובץ ויוצרת ממנו מערך של אובייקטים מסוג ProcessLine שמוכן לעיבוד מידע בצורה נוחה יותר.
   - **def findMinTimeDiff()** – הפונקציה מחשבת עבור כל שורה את הפרש הזמנים שלה מהזמנים שהכניס המשתמש.
   - **def getBlocksToCompare()** – לאחר שמצאתי את שני הבלוקים אליהם התכוון המשתמש בתוך הקובץ שהוא הזין ובהתאם לשעות שהוא הזין.
   - **def compareTwoBlocks(orig,new) -** הפונקציה משווה בינהם ומוצאת את ההבדלים שבין הקבצים.
   - **def writeBlockToFile(name, procs)** – הפונקציה כותבת את ההבדלים שנמצאו לתוך קובץ הlog.
   - **def manualScaning(filename, date1, date2)** – התוכנית הראשית שמקבלת את הזמנים השזין המשתמש והקובץ שהוא בחר ומפעילה את כל הפונקציות הנ"ל.