



תרגיל פיצוח MD5 מבוזר

בתרגיל זה נבנה מערכת לחישוב מבוזר ב-Python ונמצא בעזרתה מה המחרוזת שיצאה את ה-MD5 הבא:

EC9C0F7EDCC18A98B1F31853B1813301

מהו MD5?

<https://he.wikipedia.org/wiki/MD5>

עבודה עם MD5 דרך פייטון:

<https://www.geeksforgeeks.org/md5-hash-python/>

הערה לשם הקלה - המחרוזת היא מספר באורך 10 ספרות... בשלבי פיתוח כדאי לעבוד עם מחרוזות קצרות יותר. לשם כך השתמשו במחרוזות משלכם שתוכלו לייצר בעזרת

<https://www.md5online.org/md5-encrypt.html>

התוכנה תכתב בצורת Client-Server, כאשר יהיה שרת אחד שיחלק את המשימות ומספר Client-ים שיבצעו את החישובים.

ה-Server:

ה-Server יהיה אחראי לחלק את העבודה ל-Client-ים השונים.

הוא ידע אילו אפשרויות נוסו עד עכשיו, ויחלק לכל Client שיבקש ממנו תחום של מספרים שאותו ה-Client צריך לנסות.

ה-Client:

כל Client יהיה אחראי על המחשב שעליו הוא רץ.

אחריות זו תבוא לידי ביטוי על ידי ניצול מירבי של משאבי המחשב שעליו הוא רץ (כך ש-Client שרץ על מחשב עם 4 ליבות יבקש פי 4 יותר עבודה מ-Client הרץ על מחשב בעל ליבה אחת).

בנוסף, מרגע קבלת העבודה, ה-Client ידאג שמספר העבודות שברשותו יתאמו את מספר המעבדים במחשב (כך שהוא באמת ינצל באופן מירבי את יכולות המחשב).

השתמשו במודול ה-threading על מנת לנצל את מירב המשאבים של כל מחשב.

סיום:

לאחר שמצאתם את המחרוזת שייצרה את ה-MD5 יש לשלוח אותה לשרת.

השרת בקבלת המחרוזת צריך להודיע ללקוחות האחרים להפסיק את העבודה ולסיים.

את הפיתוח והבדיקות בצעו על המחשב שלכם (כשהוא גם ה-Client וגם ה-Server) ורק לאחר מכן נשתמש במספר מחשבים על מנת להשיג יותר כוח חישוב.

נקודות מחשבה להמשך (בנוסף למימוש) – איך הייתם ממשיים מצב בו Client-ים חדשים מצטרפים באופן דינאמי

לרשת שלכם, ו-Client-ים קיימים נעלמים בפתאומיות? שאלה זו היא אחת מהבעיות המרכזיות שעומדות בלב החישוב המבוזר הגדול באמת (אצל גוגל, אמזון ומיקרוסופט).

נכתב ע"י © עומר ברק ותומר גלון ונערך מחדש ע"י אלי גולדשטיין