

**Bash – 1 מס' 1**

תאריך הגשה: 24.01.24, 23:59

**הקדמה**

בתרגיל זה נבנה script מסוג bash שמטרתו לבדוק אם מישהו שינה לנו משהו בתיקייה נבחרת.

התרגיל מורכב מקובץ יחיד בשם bigBrother.sh, המקבל בקריאה הראשונה פרמטר אחד או רשימה של פרמטרים. הפרמטר הראשון הוא תמיד הכתובת של התיקייה שאחריה אנחנו רוצים לעקוב. אם יש פרמטרים נוספים, הם מהווים רשימה של שמות (לא כתובות) של קבצים ותיקיות שרק אחריהם התוכנה תעקוב. הפרמטרים הללו יינתנו אך ורק בקריאה הראשונה לסקריפט, מקריאה שנייה ומעלה, אין להניח שיינתנו פרמטרים.

דוגמאות לקריאה לסקריפט:

```
> ./bigBrother.sh ~
> ./bigBrother.sh `pwd`
> ./bigBrother.sh abc/ a.txt b directory g.out
```

**תקינות של פרמטרים:**

יש לבדוק שהפרמטר הראשון שמתקבל הוא נתיב תיקני לתיקיה. במידה ולא התקבל פרמטר או התקבל פרמטר לא תיקני - תודפס הודעת שגיאה "Must include valid path" ל-stderr.

**קריאה ראשונה**

בקריאה הראשונה עבור תיקייה מסויימת, יודפס הכיתוב:  
Welcome to the Big Brother

**קריאות נוספות**

בכל קריאה שנייה ומעלה עבור התיקייה שתחת מעקב, הסקריפט משווה בין המצב הקיים בתיקייה למצב התיקייה שהיה בפעם הקודמת בה רץ הסקריפט, ומדפיס את השינויים שחלו:

- קבצים ותיקיות שנמחקו – מודפסים אל stderr.
- קבצים ותיקיות שנוספו – מודפסים אל stdout.

שימו לב – אם הסקריפט יוצר קבצים משלו, הוא אינו אמור להדפיס עדכונים הקשורים לקבצים אלה (ניתן להניח כי לא נמחק או נשנה לכם קבצים שאתם יצרתם). בנוסף, מותר לייצר קבצים אלה אך ורק בתוך התיקייה ממנה נקרא הסקריפט (cwd), או בתיקייה אחריה הוא עוקב.

**קובץ שלא השתנה**

לא יודפס כל מידע בנוגע אליו. ז"א שאם שום דבר לא השתנה מאז הקריאה האחרונה, לא יהיה פלט כלל בעקבות הקריאה לסקריפט. כלומר, אם קובץ נערך (אך לא שינה את שמו), לא יודפס מידע לגביו.

**קובץ שנוסף**

אם הקובץ שנוסף הוא תיקייה, תודפס שורה שתתחיל ב-"Folder added:" ותמשיך בתיאור השם של התיקייה.  
לדוגמא:

Folder added: a

אם הקובץ שנוסף אינו תיקיה, תודפס שורה שתתחיל ב-"File added:" ותמשיך בשם הקובץ הנוצר.

לדוגמא, עבור הקובץ a.txt:  
File added: a.txt

#### קובץ שנמחק

אם הקובץ שנמחק הוא תיקייה, תודפס שורה שתתחיל ב-"Folder deleted" ותמשיך בתיאור השם של התיקייה.

לדוגמא:

Folder deleted: a

אם הקובץ שנוסף אינו תיקיה, תודפס שורה שתתחיל ב-"File deleted" ותמשיך בשם הקובץ הנוצר.

לדוגמא, עבור הקובץ a.txt:

File deleted: a.txt

בפעולת מחיקה, אין צורך להתייחס למבנה הפנימי של תיקיות, אלא רק לתוכן של התיקייה הראשית עבורה הופעל הסקריפט.

#### סדר ההדפסות

יש להדפיס קודם תיקיות ורק אז קבצים. יש להדפיס את התיקיות והקבצים בסדר אלפבתי עולה.

#### הערות:

שימו לב כי במהלך העבודה אנו מתייחסים לקובץ שנמחק במידה ואינו נמצא יותר בתיקייה. כלומר, במידה ונמחק הקובץ a.txt ובמועד מאוחר יותר התווסף קובץ a.txt, אך הסקריפט לא נקרא בין שתי הפעולות, לא נצפה מכם בפלט של התוכנית להדפיס כי a.txt נמחק ו-a.txt התווסף. יש להתייחס לקבצים ותיקיות בצורה שונה, גם אם הם בעלי אותו שם. בנוסף נשים לב כי שינוי שם של קובץ שקול למחיקה והוספה. דגש נוסף הינו שאין להניח כי קובץ ההפעלה של bigBrother יהיו ממוקם בתיקייה עליה נבצע את המעקב.

#### הנחיות הגשה

- ההגשה ביחידים בלבד. עבודה שתוגש בפורמט אחר, לא תיבדק.
- הגישו קובץ אחד בשם bigBrother.sh
- התרגיל יעבור בדיקה אוטומטית. כדי שהתוכנית תעבור בשלום את הבדיקה האוטומטית, אין לשנות את השם של הקובץ, ואין לשנות את הפורמט של ההדפסות שניתנו – הן מבחינת כמות enters, upper/lower case ורווחים. אם ההדפסה אינה מדוייקת כנדרש, הבדיקה תיכשל.
- התרגיל יעבור בדיקות קפדניות למציאת העתקות, וסטודנטים שימצאו כי העתיקו יענשו בחומרה. אנא המנעו מהעתקות.
- זכרו – מבחינת bash, אותיות קטנות אינן זהות לאותיות גדולות ( $a \neq A$ ). שימרו על אחידות בין העבודה שלכם למסמך זה, ועל אחידות בתוך העבודה עצמה.
- מותר לכם ליצור קבצים נוספים מתוך הסקריפט (לדוגמא בעזרת touch), אך אין לצרף קבצים נוספים לעבודה, ואין לתת שמות יצירתיים בעברית לקבצים שלכם.
- אין צורך לבדוק תקינות של הקלט איפה שלא מוגדר (לא תיבדקו על המקרה של מתן פרמטרים נוספים לתוכנית, לדוגמא).
- את העבודות יש להגיש דרך אתה הקורס (moodle) במקום המיועד לכך.
- יש לקרוא את מדריך ההגשות בקפידה ולעבור על הצ'קליסט לפני הגשת העבודה.

- שאלות לגבי העבודה ניתן יהיה לשאול בפורום מיוחד שיפתח באתר הקורס (moodle).

בהצלחה !!!

## דוגמאות הרצה

מצורפת תיקייה עם דוגמאות ל-output ול-error של קריאות לסקריפט בהתאם למצבים המתוארים. שימו לב, מדובר בדוגמאות שלא עוברות על כל המקרים האפשריים, יש לבצע בדיקות מקיפות לפי הוראות העבודה.

בכל קריאה לסקריפט שלכם אחרי שעשיתם את הפעולות הרלוונטיות, אתם אמורים לקבל בדיוק את ה-output וה-error שמופיעים בתיקייה הרלוונטית עבור המקרה. כל מקרה מסומן במספר מזהה, בתיקייה זו יהיו הקבצים אליכם להשוות. בפעם הראשונה שאתם קוראים לסקריפט, עליכם להשוות לקבצים עם השם "1". בקריאה השנייה, לקבצים עם השם "2". בקריאה השלישית לקבצים עם השם "3" וכך הלאה. השוו את ה-output שלכם עם קובץ out ואת ה-error שלכם עם קובץ err. כל הקריאות מניחות שהתיקייה קיימת ותיקנית אלא אם נאמר אחרת.

### יצירה ומחיקה של קובץ (1)

1. נקרא לסקריפט עם התיקייה tracked
2. נוסף קובץ a.txt ונריץ שוב את הסקריפט.
3. נמחק את a.txt ונריץ שוב את הסקריפט.

### יצירה ומחיקה של תיקייה (2)

1. נקרא לסקריפט עם התיקייה tracked
2. נוסף תיקייה a ונריץ שוב את הסקריפט.
3. נמחק את התיקייה a ונריץ שוב את הסקריפט.

### קריאה לסקריפט עם תיקייה לא קיימת (3)

1. נקרא לסקריפט עם התיקייה abc שלא קיימת.

### קריאה עם `pwd` (4)

1. נקרא לסקריפט עם התיקייה `pwd`
2. נוסף תיקייה בשם a וקובץ בשם b.txt ונריץ שוב את הסקריפט.
3. נמחק את התיקייה a והקובץ b.txt ונריץ שוב את הסקריפט.

### הוספת קבצים מרשימת הקבצים (5)

1. נקרא לסקריפט עם הארגומנטים tracked/ a.txt b.txt
2. נוסף קובץ a.txt ונריץ שוב את הסקריפט.
3. נוסף קובץ b.txt ונריץ שוב את הסקריפט.

### הוספת קבצים שלא מרשימת הקבצים (6)

1. נקרא לסקריפט עם הארגומנטים tracked/ a.txt b.txt
2. נוסף קובץ c.txt ונריץ שוב את הסקריפט.
3. נוסף קובץ d.txt ונריץ שוב את הסקריפט.

### הוספת תיקיות שברשימה והסרתן (7)

1. נקרא לסקריפט עם הארגומנטים tracked a b
2. נוסף את התיקיות a ו-b ונריץ שוב את הסקריפט.
3. נמחק את התיקיות a ו-b ונריץ שוב את הסקריפט.