

תרגיל 4 C בסביבת יוניקס

שאלה (1)

עברו על טבלת ה- file descriptors וכתבו לקובץ בשם descriptor_file את רשימת ה- descriptors הפנויים והתפוסים הכתיבה לקובץ תתבצע ע"י הפקודה `printf`.

ההדפסה של ה- descriptors התפוסים תהיה בעזרת שני סיגנאלים שתדרסו, על רשומה/שורה זוגית תפוסה יהיה כתוב `sigINT true` על רשומה/שורה אי-זוגית תפוסה יהיה כתוב `sigSEGV true`.
על אלו הפנויים יהיה כתוב בשורה מספר ה- descriptor ו- `false`.
על מנת לבדוק את התכנית צרו קובץ אחד ב- shell ואת הקובץ השני תצרו ב- C. תפתחו את שני הקבצים את הראשון שפתחתם תסגרו ואז תדפיסו לקובץ את הטבלא.

useful function: `fcntl()` `signal()` `kill()`

```
sigINT true
sigSEGV true
sigINT true
4 false
sigINT true
```

שאלה (2)

כתבו בפסודו-קוד (לשפה פרוצדוראלית רגילה) תכנית המריצה שלושה תהליכים עם שני pipes ביניהם לדוגמא:

`Propt>abc | xyz | ABC`

שאלה (3)

צרו תהליך היוצר קובץ חדש עם הרשאות של **קריאה וכתיבה בלבד**. תהליך זה יוצר 3 תהליכונים (threads); התהליכון הראשון כותב לקובץ 10 פעמים 1 בשורה אחת ובסוף השורה כותב את מספר התהליכון (thread ID) ויורד שורה, השני עושה את אותו הדבר (על אותו הקובץ) אבל עם המספר 2 והשלישי אותו דבר (על אותו הקובץ) עם המספר 3.

שאלה (4)

תעשו את אותו הדבר כמו שאלה 3 אבל שתמיד יהיה סדר בין השורות ובתוך השורות, ז"א שתמיד השורה הראשונה הרביעית וכו' תהיה עם המספר 1 ורק עם המספר 1, שהתהליכון השני לא יוכל פתאום לכתוב באמצע השורה הראשונה 2. בנוסף שהתהליכון הראשון לא יכתוב שתי שורות רצופות. בפתרון שאלה זו עליכם להשתמש במנעול (mutex).

שאלה (5)

כמו שאלה 4 אבל להשתמש בסמפור (semaphore).

שאלה (6)

בדקו האם אתם יכולים לעשות כמו שאלה 4 אבל להשתמש רק עם סיגנאלים כגון `sigwait`.
אם כן ממשו התכנית.

שאלה (7)

בדקו האם אתם יכולים לעשות כמו שאלה 4 אבל להשתמש רק בפונקציה `pthread_yield`.
אם כן ממשו התכנית.

שאלה (8)

בדקו האם אתם יכולים לעשות כמו שאלה 4 אבל ניתן להשתמש בפונקציה `pthread_yield` וב- `sigwait`.
אם כן ממשו התכנית.