

ב"ה

### תרגיל מס' 3 – named pipes(FIFOS) and shared memory

#### הוראות הגשה

- שאלות בנוגע לתרגיל נא להפנות לצוות הקורס בכתובת [os.89231@gmail.com](mailto:os.89231@gmail.com) (לא יהיה מענה למיילים שיגיעו לכתובות מייל אחרות):
- מענה יינתן אך ורק בין התאריכים: 07-14/06/2017 (כולל)  
**אנא היערכו בהתאם !!**
- מועד אחרון להגשה: 23:59 30/06/17.
- יש לשלוח את הקבצים באמצעות האתר:  
<http://help.cs.biu.ac.il/submit.htm>  
לפני חלוף התאריך הנקוב לעיל.
- שם ההגשה של תרגיל 3 : ex3
- יש להקפיד מאוד על כל הוראות עיצוב הקלט והפלט כפי שלמדתם בקורס C (בדיוק אותו coding style)
- **עליכם לבדוק בנוגע לכל פונקציה האם היא הצליחה או לא**, אם היא לא הצליחה יש לתת הודעה מתאימה ל STDERR ולסיים את התכנית (בצורה נקייה כמובן).
- אין להשתמש בפונקציה ספריה אם ראינו בכיתה SYSTEMCALL מקבילה לה.
- **להזכירכם, העבודה היא אישית. "עבודה משותפת" דינה כהעתקה.**
- אין להדפיס למסך שום דבר מעבר למה שנתבקש בתרגיל.
- יש לוודא שהתרגיל מתקמפל ורץ על U2 ללא שגיאות/אזהרות.

### כללי:

בתרגיל זה תצטרכו להשתמש ב: fifo וב shared memory.  
התרגיל מורכב מ-2 קבצים היוצרים יחדיו את המשחק רברסי  
(<http://www.web-games-online.com/reversi/>).  
הקובץ הראשון הינו צד השרת והוא צריך לרוץ ראשון.  
הקובץ השני הינו צד הלקוח ותפקידו לשחק בשם השחקן ולהדפיס תזוזות של היריב למסך.

### רברסי – צד השרת

#### הנחיות עבור ex31

- שם התרגיל: ex3
- שם קובץ מקור (source file) שיש לשלוח: ex31.c

כתבו תכנית המייצרת זיכרון משותף בעל המפתח היחודי:

ftok("ex31.c", 'k')

בנוסף התכנית צריכה לייצר FIFO העונה לשם fifo\_clientTOServer  
על התכנית להאזין ל FIFO זה עד שיעבור פעמיים דרך ה FIFO מידע מסוג pid\_t.  
מידע זה הינו ה- pid (מזהה תהליך ייחודי) של 2 תהליכים שרוצים לתקשר בניהם.  
כאשר 2 התהליכים העבירו את המידע דרך ה FIFO, על התכנית למחוק את ה FIFO.  
כעת, על התכנית לאותת לתהליך הראשון בעזרת SIGUSR1 על מנת שיבחר את המהלך שברצונו לבצע.  
כאשר התהליך הראשון יסיים לכתוב את המהלך שלו לזכרון המשותף על התכנית לאותת לתהליך השני שעכשיו תורו.

מכאן והלאה התהליכים לא יקבלו יותר איתותים מהתכנית אלא יבדקו לבד מתי התור שלהם הגיע.  
התכנית תשמור לעצמה מידע על מצב הלוח וכאשר אחד הצדדים מנצח או שיש תיקו התכנית צריכה להדפיס GAME OVER, הודעת הניצחון או התיקו (מפורטות בהמשך) למחוק את הזיכרון המשותף ולהסתיים.

שימו לב:

1. בסה"כ ישלחו 2 איתותים מסוג SIGUSR1 (איתות לכל תהליך).
2. יעברו דרך ה FIFO 2 pids
3. המהלכים בין 2 השחקנים יירשמו בזיכרון המשותף.

אין לשלוח סיגנלים אחרים/נוספים או לשנות במקצת את הפרוטוקול שהוגדר.

## רברסי – צד הלקוח

### הנחיות עבור ex32

- שם התרגיל: ex3
- שם קובץ מקור (source file) שיש לשלוח: ex32.c

על התכנית שלכם לבצע את הפעולות הבאות:

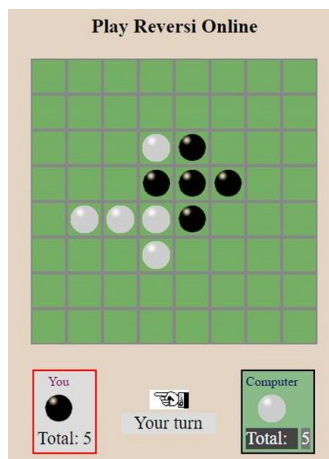
- לפתוח את ה FIFO clientTOserver fifo\_ ולכתוב לתוכו את ה PID של התהליך
  - לחכות לסיגנל מסוג SIGUSR1, לאחר הגעת הסיגנל יש להתחבר לזכרון המשותף שהמפתח שלו הוגדר לעיל (בחלק הראשון).
  - לבקש מהמשתמש להזין את המשבצת שאליה הוא מעוניין להוסיף אסימון פורמט משבצת [x,y] עמודה x שורה y. קלט לדוגמא: [4,3] (שימו לב, לאחר הזנת הקלט על המשתמש ללחוץ enter)
    - לבדוק שהמשבצת המבוקשת פנויה ותקינה (גם מבחינת גודל הלוח וגם מבחינת **הערה כללית 3**). אם לא, יש להדפיס הודעת שגיאה למשתמש ולבקש ממנו להזין פרמטרים חדשים ולחזור על הפעולות בסעיף 3b.
    - להדפיס למסך את הלוח לאחר הצעד של המשתמש
    - להדפיס למשתמש הודעה שעכשיו ממתינים לצד השני שישחק
    - לכתוב את המהלך לזיכרון המשותף.
    - לזכרון המשותף צריך לכתוב ארבעה תווים: **התו הראשון**: אם אני התהליך הראשון שקיבל SIGUSR1 התו צריך להיות "b", אחרת הוא צריך להיות "w".
  - התו השני**: מספר עמודה (שבה המשתמש רוצה למקם את האסימון שלו)  
**התו השלישי**: מספר שורה (שבה המשתמש רוצה למקם את האסימון שלו)  
**התו הרביעי**: '0'
  - לבדוק בזיכרון המשותף אם הוכנס מהלך חדש (בלולאה עם sleep של שניה והדפסת הודעת המתנה למשתמש השני [מפורטת בהמשך] בכל איטרציה)
  - ברגע שהוכנס מהלך חדש, יש לעדכן את לוח המשחק שאצלי (על פי הצעד של היריב) ולהדפיס למסך. **שימו לב שהצעד של היריב משפיע לא רק על האסימון שהניח ויכול להשפיע גם על אסימונים נוספים.**
  - חזרה לשלב 3
- אם אחד השחקנים ביצע מהלך שמסיים את המשחק (הלוח מלא או שכל האסימונים באותו צבע - כלומר שלשחקן הבא אין צעד חוקי לבצע) יש להדפיס למסך את ההודעה המתאימה (מפורט בהמשך) ולסיים את התכנית בצורה מסודרת. המנצח הוא השחקן שמספר האסימונים בצבע שלו על המסך גדול יותר.

### הערות כלליות:

- גודל הלוח הינו 8 עמודות על 8 שורות.
- בדומה לחוקי המשחק הלוח הראשוני מכיל 4 אסימונים. 2 אסימונים שחורים באינדקסים [3,3] ו [4,4] ושני אסימונים לבנים באינדקסים [4,3] ו [3,4].
- כל שחקן חופשי למקם אסימון מהצבע שלו רק במקומות שבהם יש באפשרותו להפוך אסימונים של היריב (בהתאם להוראות המשחק המקורי).
- כדי לבדוק את התרגיל שלכם הבודק יפתח 3 חלונות טרמינל, בראשון הוא יריץ את קובץ הריצה המתקבל מ ex31, בשני הוא יריץ את קובץ הריצה המתקבל מ ex32 ובשלישי שוב הוא יריץ את קובץ הריצה המתקבל מ ex32.
- הניחו שהמשתמש תמיד מזין מספר שלם עבור העמודה והשורה בה ברצונו להוסיף אסימון.

6. יש לטפל בשגיאות שמתקבלות מקריאה לפונקציה מערכת ע"י הדפסת הודעת שגיאה (סטנדרטית) ויציאה מהתכנית.
7. השתמשו בקריאות המערכת `fork,dup.dup2,exec,open,write,read`. אסור להשתמש בפונקציות ספרייה אלטרנטיביות לפונקציות הנ"ל.
8. יש לצייר את הלוח עם התווים: 2 לתאים עם אסימון שחור, 1 לתאים עם אסימון לבן ו0 לסימון תא ריק.  
המחשת הדפסה:

```
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 1 2 0 0 0
0 0 0 2 2 2 0 0
0 1 1 1 2 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```



9. התא הראשון בלוח (השמאלי העליון) יסומן כ  $[0,0]$
10. ההודעות שיש לכתוב למסך:

בקשת קלט:

Please choose a square

קלט שגוי:

הזון מספר משבצת לא חוקי

No such square

Please choose another square

המשבצת לא חוקית על פי הערה כללית 3 (כולל את המקרה של משבצת מלאה)

This square is invalid

Please choose another square

הדפסת לוח:

The board is:

המתנה לשחקן השני:

Waiting for the other player to make a move

ניצחון:

Winning player: Black

Winning player: White

תיקו:

No winning player

בהצלחה!

