

## מערכות הפעלה – המכללה האקדמית תל אביב – יפו

### תרגיל בית מספר #3

סמסטר א', תשפ"א 2021

מרצה: כרמי מרימוביץ. מתרגל: צבי מלמד

תאריך הגשה: מוצ"ש 26/12/20 בחצות

### עדכונים:

⊕ 12.12 – תיקון ניסוח [כאן](#) – הודעות עומדות לרשות הצרכנים רק לאחר שהתווספו לרשימה.

### הוראות כלליות

התרגיל הזה כולל משימה אחת מרובת פתילים. את הסנכרון בין הפתילים עליכם לבצע באמצעות משתני תנאי ומנעולי MUTEX של threads. התרגיל כולל קבצים שאותם אתם מקבלים, עליכם להשתמש בהם (למשל שימוש בקבועים או קריאות לפונקציות שמוגדרות שם), ואסור לכם לשנות אותם.

### שאלה #1 – מימוש מערכת יצרן-צרכן ועבודה עם threads.

#### תיאור כללי:

בתכנית הזאת הפתיל הראשי יוצר מספר פתילים יצרניים – להלן producers, ומספר פתילים צרכניים, להלן consumers. היצרנים מייצרים פריט מטיפוס item ומוסיפים אותו לרשימה מקושרת. הצרכנים מאתרים ברשימה איבר שלא טופל עדיין, מבצעים עליו עיבוד כלשהו, מעדכנים אותו, וכותבים על כך לפלט.

#### פעולת היצרנים:

לכל פתיל יצרן יש מספר מזהה (1, 2, ...). כל אחד מהיצרנים מגריל מספרים ע"י קריאה לפונקציה `get_random_in_range` עד שהוא מגריל שני מספרים ראשוניים. את הבדיקה הוא מבצע ע"י קריאה לפונקציה `is_prime`. הייצרן מייצר מבנה `struct item`, ובשדה המתאים הוא כותב את המכפלה. כמו כן הוא כותב בשדה הסטטוס את הערך `NOT_DONE`. לאחר מכן, הוא מוסיף את הפריט לרשימה ע"י קריאה לפונקציה `add_to_list`.

#### פעולת הצרכנים:

לכל פתיל צרכן יש מספר מזהה (1, 2, ...). כל אחד מהצרכנים שולף פריט `item` לא מעובד מהרשימה, ע"י קריאה לפונקציה `get_undone_from_list`. הוא מוצא את המרכיבים הראשוניים של המכפלה וכותב אותם לשדות `n1`, `n2` של המבנה, וכמו כן משנה את הסטטוס ל `DONE`. זה מתבצע ע"י קריאה לפונקציה `set_two_factors`.

#### פעולת הפתיל הראשי:

הפתיל הראשי מייצר `N_PROD` פתילים יצרניים ולאחר מכן הוא מייצר `N_CONS` פתילים צרכניים. לאחר מכן הוא ממתיין שכל הפתילים יסיימו את ריצתם. לאחר שהם סיימו את ריצתם הוא מדפיס את הרשימה ע"י קריאה לפונקציה `print_list`.

## הדפסות לפלט הסטנדרטי:

עליכם לדאוג להדפסות הבאות לפלט הסטנדרטי:

- (יצרן) הודעה על הוספת פריט לרשימה – מתבצע ע"י קריאה לפונקציה `write_adding_item`.
- (צרכן) הודעה על שליפת פריט מהרשימה – מתבצע ע"י קריאה לפונקציה `write_getting_item`.
- (יצרן) הודעה של כל פתיל יצרן בסיום עבודתו – ע"י קריאה לפונקציה `write_producer_is_done`.
- (צרכן) הודעה של כל פתיל צרכן בסיום עבודתו – ע"י קריאה לפונקציה `write_consumer_is_done`.
- (הפתיל הראשי) הדפסת הודעות שהוא יצר את הפתילים היצרניים/צרכניים או שכולם הסתיימו. זה מתבצע ע"י שימוש בהגדרות `ALL_PROD_CREATED`, `ALL_CONS_CREATED`, `PROD_TERMINATED`, `CONS_TERMINATED`. אין להוסיף תוספות להגדרות/הדפסות. לדוגמא, קריאה יכולה להיות: `printf(ALL_PROD_CREATED);`.
- (הפתיל הראשי) הדפסת הרשימה ע"י קריאה לפונקציה `print_list`.

## דרישות סנכרון

- היצרנים ממתינים עד שכל הפתילים נוצרו, ורק אז הם מתחילים לרוץ.
- הצרכניים ממתינים **עד שנוצרו והתווספו** `ITEM_START_CNT` הודעות לרשימה, ורק אז הם מתחילים לרוץ. (שימו לב שהפונקציה `add_to_list` מבצעת יצירה של פריט והוספתו לרשימה).
- היצרנים מייצרים בסה"כ `TOTAL_ITEMS` פריטים, ומסתיימים לאחר שנוצרה הכמות הזאת של פריטים. הצרכנים צורכים (שולפים מהרשימה ומעבדים) `TOTAL_ITEMS` פריטים, ומסתיימים לאחר שעבדו את הכמות הזאת של פריטים.
- אם נוצר מצב שבו כמות הפריטים שנוצרה שווה לכמות הפריטים שעובדו, אזי הצרכניים ממתינים. כלומר, הם לא ינסו לשלוף פריט מהרשימה, אם אינם יודעים בוודאות שיש שם לפחות פריט אחד שהסטטוס שלו הוא `NOT_DONE`.
- אסור שההדפסות יתערבבו אחת עם השנייה. כל כתיבה לפלט צריכה להיות "אטומית" ע"י שימוש במנעול.
- הכנסת איבר לרשימה וההדפסה על כך, צריכות להיות ביחד (גם כאן, לגרום לכך שזאת תהיה סוג של "פעולה אטומית". באותה אופן גם השליפה של איבר מהרשימה וכתיבה ההודעה על כך.
- בגלל מגבלות של הפונקציה `rand()` – אין לקרוא בו זמנית לפונקציה הזאת משני פתילים. ולכן יש להתייחס אליה כאל קטע קריטי. מבחינתכם, הקריאה לפונקציה `get_random_in_range` היא קטע קריטי.

## קבצים שאתם מקבלים:

בתרגיל הזה עליכם לכתוב קובץ מקור שנקרא `ex3_q1.c`, ולשיקולכם אם אתם רוצים לכתוב גם `ex3_q1.h`. קיימים שני קבצים שבהם עליכם להשתמש. הקבצים האלו מכילים הגדרות של קבועים, פרוטוטיפים של פונקציות, ומימוש של פונקציות. שמות הקבצים האלו הם: `ex3_q1_given.c` and `ex3_q1_given.h`. עליכם להגיש את התרגיל שלכם כך שהוא עובד עם הקבצים המקוריים שנתנו לכם. **זאת דרישה בסיסית, ומי שלא יקפיד על כך, התרגיל שלו עלול שלא יתקמפל אצל הבודק, והציון יהיה אפס**, מכיוון שהבודק ישתמש בעותק הפרטי שלו של הקבצים האלו, ולא במה שתגישו. לדוגמא, **אין להוסיף INCLUDE** לקבצים האלו.

יחד עם זאת, תוך כדי תהליך הפיתוח, אתם יכולים לשנות את הקבצים האלו, לשנות את גודל הקבועים, אולי להכניס הודעות DEBUG בחלק מהפונקציות. אפילו פונקציה כמו `get_random` אתם יכולים לשנות כראות עיניכם, תוך כדי תהליך הפיתוח, אם זה מקל עליכם. זכרו – התכנית שאתם מגישים חייבת להתמקפל ולרוץ עם הקבצים המקוריים.

עיינו בעיקר בקובץ `ex3_q1_given.h` לפני תחילת העבודה. מומלץ לעיין גם בקובץ `ex3_q1_given.c` למקרה שתמצאו להבין יותר טוב "מה קורה" (או "מה צריך לקרות") או לנגוע בקוד (למשל כתיבת הודעות דיבאג) תוך כדי הפיתוח. אל תשכחו לבדוק את התרגיל שלכם עם הקבצים המקוריים שקיבלתם לפני ההגשה במאמא. חבל (באמת) שירדו נק' על שטויות כאלו (ובבקשה – אם לא הקפדתם על כך – אז לא להתחנן לאחר מכן).

### הפלט המצופה:

קובץ נוסף שאתם מקבלים, הוא קובץ פלט שיצירתי בהרצת התרגיל שלי. שם הקובץ: `out.log_master`. שימו לב שבגלל אקראיות ותזמונים שונים, הפלט יכול להיות שונה מריצה לריצה – בעיקר הבדלים מסוימים בסדר ההודעות. ככל שההגרלה של המספרים לא תהייה באמת אקראית (ע"י שתילת SEED קבוע) הפלטים יהיו דומים זה לזה, אבל לא זהים.

### א. Makefile והקבצים הנתונים

עליכם למסור `MAKEFILE` תקין ועובד, שקיימת בו כניסת `test`. הכניסה הזאת תכיל את הכניסה הבאה:

```
test:
    ./ex3_q1 > out.log 2> err.log
```

### הערות והנחיות

א. יש הפרש זמן בין יצירת פריט לבין הכנסתו לרשימה. כלומר, לא מספיק לעקוב (למנות) את מספר הפריטים שהוכנסו לרשימה, אלא יש לספור גם את מספר הפריטים שנוצרו. **אין ליצור פריט חדש, אם נוצרו כבר TOTAL\_ITEMS של פריטים**. יחד עם זאת, כאשר ניגשים לשלוח פריט מהרשימה – יש לבדוק את מספר הפריטים שהוכנסו כבר, לעומת מספר הפריטים שנושלפו כבר, ואם הם שווים זה לזה, יש להמתין עד שיוכנס פריט חדש לרשימה.

ב. יוצא מהסעיף הנ"ל, שיש לנו לפחות את המונים הבאים: מספר הפריטים שנוצרו, מספר הפריטים שהוכנסו לרשימה, מספר הפריטים שעובדו (לאחר שנושלפו מהרשימה). מומלץ להתשמש במנעול אחד עבור כל המונים האלו, ולא מונה נפרד לכל מונה. זה יפשט לכם את הקוד, ויחסוך לכם מצבים של `dead-lock`. מדוע? כי ריבוי מנעולים זה מתכון למצבים כאלו. בכלל, נדרשת זהירות מרבית בעבודה עם המנעולים, בכדי לא לגרום להיתקעויות כאלו.

### ג. בתרגיל זה אין להשתמש בסמפורים!

ד. קחו בחשבון מבחינת כתיבת התכנה - התרגיל הבא עשוי להכיל משימה שבה תתבקשו לכתוב תרגיל דומה לתרגיל הנוכחי, אלא שהסינכרונים השונים יתבצעו באמצעות סמפורים במקום מנעולים.

ה. >יתכן שיתווספו כאן הנחיות נוספות בהמשך, כתגובה לשאלות של תלמידים<.

**בהצלחה!!**