# שאלה 1

## סעיף 1.

יוסי לא צודק. מפני שלאחר סיום ה-epoch יכולים להיות תהליכים שנצאים במצב המתנה וה-counter שלהם שונה מ-0. ולכן חישוב של counter/2 אינו מיותר.

## סעיף 2.

משה טועה. התשובה תלוי בכמות המעבדים. כי במקרה ש יש מספר מעבדים- יכול להיות שהעדיפות של תהליך המוחלף גדולה יותר מאשר של אחד מהתהליכים הרצים באחת מהליבות האחרות. במידה ויש ליבה אחת- השורה מיותרת.

# שאלה 2

## סעיף 1.

התשובה ג' הינה נכונה. כי כאשר מגיעה פסיקת חומרה אנחנו צריכים לעבור מ-User mode ל-Kernel mode וכתוצאה מכך להרחיב אנחנו צריכים לקרוא בין היתר את המצביע לבסיס המחסנית הגרעין של התהליך הנוכחי.

## סעיף 2.

תשובה ג נכונה.

א. אין צורך לתת שם עם sys\_ בהתחלה כי קיימת טבלה של קריאות מערכת. ובזמן של פסיקת תוכנה מועבר גם מספר של קריאת מערכת, ולפיו המערכת נגשת לפונקציה שרשומה תחת מספר השירות המתאים.

ב. מספר הארגומנטים יכול להיות גדול מ6 כי אין שום הגבלה על מספר הפרמטרים. כי wraper זאת פונקציה רגילה.

ג. **נכון,** מספר הארגומנטים לא יכול להיות גדול מ6 כי אחרת נקבל שגיאת ריצה כי המעבר נעשה ע''י הרגיסטרים שיש לנו מקסימום 6 כאלה.

ד. פקודת int 0x80 היא פקודת assembly inline. הקומפיילר לא מסתכל לשם של המשתנה. כי הוא יודע שזאת פקודה לא ב-С, והוא לא ינסה להגדיר משתנה חדש. הוא יקבל את הפקודה בתור פקודת assembly.

## סעיף 3.

**ד' נכון**: אם נעשה את ההחלפה של שורות 3,4,5 לפקודה call, אז הפקודה call אוטומטית תשמור את הכתובת של פקודה הבאה שזה הכתובת של label 1. במקום זה נדחוף למחסנית את הכתובת של פקודה הבאה דרך שדה thread.eip. כי במקרה שבו התהליך רץ בפעם הראשונה- נצטרך לחזור לתחילת הפונקציה ret\_from\_fork() (שיהיה שמור בnext->thred.eip) ולא label 1. הרי מובטח לנו שאם התהליך רץ לא בפעם הראשונה, אז ב שדה next->thread.eip כבר מופיע כתובת של label 1. אז כדי לאפשר את האופציה הזאת נצטרך לעשות את הקפיצה לפונקציה \_\_swich\_to באופן "ידני" בעזרת movl, pushl, jmp. ולא עם call.

# שאלה 3.

נתון ש , ולכן .   
נחשב את הtimeslice בעזרת מאקרו TASK\_TIMESLICE:

בהנחה שאין הרעבה התהליך יקבל את הtime slice של 151 msec ואז בגלל שהתהליך הוא אינטרקטיבי והיחידי, אז הוא יקבל פעמיים time slice בepoch הנוכחי. כלומר סה''כ הוא ירוץ **302 msec**.