

מעבדה 6 - תזמון תהליכים (אלגוריתם התזמון של מ"ה)

תרגיל 1:

עליכם לכתוב תכנית אשר בוחנת את תזמון centos. כלומר, מוצאת מהו בערך ה time slice (time quanta) והאם הוא תלוי במשך הריצה של התוכנית ובכמות החוטים לתהליך.

לשם כך עליכם לכתוב תכנית המפעילה N חוטים ומגדירה משתנה גלובלי שמשקף מי החוט שכרגע מחזיק CPU.

כל החוטים יקראו לאותה הפונקציה f, עם פרמטר המציין את המספר הסידורי של החוט (חוט ראשון - מספר 1 וכן הלאה...).

בפונקציה f (אותה יריץ כל חוט) נדרש בלולאה אינסופית לקחת זמן התחלה כשהחוט התחיל לרוץ ולבצע לולאה עד החלפת חוט. בלולאה זאת, זמן הסיום יתעדכן כל הזמן. ניתן לזהות בעזרת המשתנה הגלובלי מתי הוחלף החוט. בהחלפה חזרה, ידפיס את זמן הסיום פחות זמן ההתחלה (על פי elapsedTime. ראו הסבר מטה).

המספר N יועבר לתוכנית כפרמטר משורת הפקודה.

לדוגמא, הפלט בהרצת התוכנית:

עבור 2 Threads:

```
braude@Cent ~]$ ./time.out 2
Time slice for thread 1 = 55.308000 ms.
Time slice for thread 2 = 98.495000 ms.
Time slice for thread 1 = 84.468000 ms.
Time slice for thread 2 = 49.385000 ms.
Time slice for thread 1 = 56.530000 ms.
Time slice for thread 2 = 50.367000 ms.
Time slice for thread 1 = 47.043000 ms.
Time slice for thread 2 = 49.856000 ms.
...
```

עבור 5 Threads:

```
Time slice for thread 2 = 54.827000 ms.
Time slice for thread 4 = 87.678000 ms.
Time slice for thread 1 = 87.798000 ms.
Time slice for thread 5 = 66.035000 ms.
Time slice for thread 3 = 88.086000 ms.
Time slice for thread 2 = 86.809000 ms.
Time slice for thread 4 = 86.967000 ms.
Time slice for thread 1 = 86.169000 ms.
Time slice for thread 5 = 86.152000 ms.
Time slice for thread 3 = 86.130000 ms.
...
```

דוגמא לשימוש ב `gettimeofday`:

```
#include <stdio.h>
#include <time.h>           // for gettimeofday()

int main()
{
    struct timeval t1, t2;
    double elapsedTime;

    // start timer
    gettimeofday(&t1, NULL);

    // do something
    // ...

    // stop timer
    gettimeofday(&t2, NULL);

    // compute and print the elapsed time in millisec
    elapsedTime = (t2.tv_sec - t1.tv_sec) * 1000.0;    // sec to ms
    elapsedTime += (t2.tv_usec - t1.tv_usec) / 1000.0; // us to ms
    printf("elapsedTime = %lf ms.\n", elapsedTime);

    return 0;
}
```

תרגיל 2:

עליכם לכתוב תכנית אשר בוחנת האם כאשר מופעלים N חוטים תזמן `centos` נותן זמן CPU בצורה שוויונית לחוטים. כלומר, עליכם לכתוב תכנית היוצרת $N+1$ חוטים ו N מונים (long int counters) מוג `ms`).

בתכנית:

N חוטים לאורך כל חייהם מקדמים מונה מסוים. כלומר לכל i ($1 \leq i \leq N$), חוט i מקדם מונה i . החוט ה $N+1$ מדפיס את כל המונים כל 2 שניות.

על התכנית שכתבת לרוץ 20 שניות בלבד.

המספר N יועבר לתוכנית כפרמטר משורת הפקודה.

כך, לאחר הרצת התכנית, תוכלו לבדוק אם באופן כללי המונים מתקדמים באותו קצב...

הוראות הגשה:

יש להגיש קובץ WORD ובו:

1. שמות ות.ז. של שני המגישים.

2. תכנית פתרון לתרגיל 1 וצילום מסך הרצה.

3. תכנית פתרון לתרגיל 2 וצילום מסך הרצה.

ההגשה במודל או באופן פרונטלי למתרגל המעבדה. בהגשה במודל, יש להגיש את הקובץ

על ידי שני בני הזוג. בכל מקרה, ההגשה עד שבוע מזמן המעבדה לה אתם רשומים.