

המחלקה להנדסת תוכנה

29/01/2015
9:00 – 12:00

מערכות הפעלה

ד"ר מיכאל אורלוב
מר דניאל חנקין

תשע"ה סמסטר א'
מועד א'

חומר עזר: 3 דפים A4 דו צדדיים, כתובים ע"י הסטודנט (לא מודפס, לא מצולם)
מחשבון רגיל שלא ניתן לתכנות

הנחיות כלליות:

- יש לענות על השאלות במחברת הבחינה בלבד
- יש להתחיל כל שאלה בדף חדש, ולסמן את מס' השאלה בצורה ברורה
- יש לענות בקצרה (ניתן לענות בעד 3 שורות על רוב הסעיפים)
- סכום הנקודות בבחינה הוא 100

השאלון מכיל 5 דפים (כולל דף זה).
השאלות מנוסחות במין זכר מטעמי נוחות בלבד.

בהצלחה !

=====

שאלה 1 (20 נק')

עיינו בתוכנית הבאה, המשתמשת בתהליכונים:

```
#include <stdio.h>
#include <pthread.h>
int sum=0;
pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;

void* thread(void *threadid) {
    int *ptr = (int *) threadid;
    pthread_mutex_lock(&mutex);
    sum += *ptr;
    pthread_mutex_unlock(&mutex);
}

int main() {
    int i;
    pthread_t tid;
    for (i=1; i<=10; i++) {
        pthread_create(&tid, NULL, thread, (void *)&i);
    }
    printf("%d\n", sum);
}
```

סעיף א'

מה התכנית הזו תדפיס בסוף?

- א. 0
- ב. 55
- ג. 0 או 55
- ד. הסכום $1+2+\dots+k$ עבור k כלשהו בין 1 ל-10
- ה. הסכום $1+2+\dots+k$ עבור k כלשהו בין 1 ל-10, או 0
- ו. מספר כלשהו בין 0 לבין 55 כולל
- ז. מספר כלשהו בין 10 לבין 55 כולל
- ח. היא תדפיס ערך שרירותי כלשהו ("זבל")
- י. התכנית תתקע לפני שתדפיס משהו

סעיף ב'

נשנה את התוכנית באופן הבא:

1. בסוף הפונקציה thread נוסף את השורה: `pthread_exit(NULL);`
2. לפני הקריאה ל-`printf()`, נוסף את השורה: `pthread_join(tid, (void**) &i);`

האם התשובה תשתנה, וכיצד?

יש להסביר ולנמק את התשובה לשני הסעיפים!

תשובה נכונה – 2 נק', הסבר – 8 נק'

שאלה 2 (20 נק', שאלה ממטלת בית מס' 2)

סעיף א

להלן תהליכים **P1, P2** שרצים במקביל.

1. קטעים **d b** ביחד הם קטעים קריטיים זה לזה ואסור שירוצו במקביל.
2. קטע **e** חייב להסתיים לפני התחלת קטע **c**.

השתמש בסמפורים על מנת לקיים את כל התנאים. הקפד לאתחל את הסמפורים בצורה נכונה. במידה ואתה משתמש בכמה סמפורים הקפד להבדיל ביניהם בשמות המשתנים. עליך להשתמש בפקודות של סמפורים: `sem_init(sem,n), up(sem), down(sem)` בלבד.

יש לרשום 3 קטעים: אתחולים, **P1, P2**.

P3	P4		P1	P2
<pre>while (true) { g; h; m; }</pre>	<pre>while (true) { j; k; s; }</pre>		<pre>a; b; c;</pre>	<pre>d; e; f;</pre>

סעיף ב

לעיל רשומים תהליכים **P4, P3** שרצים במקביל, שניהם בלולאה.

1. קטעים **h k** ביחד הם קטעים קריטיים זה לזה ואסור שירוצו במקביל.
2. קטעים **s i m** יעבדו לסירוגין, כאשר **m** מתחיל (**m** חייב להסתיים לפני התחלת קטע **s**, ו-**s** חייב להסתיים לפני ש **m** יפעל שוב וחוזר חלילה)

השתמש בסמפורים על מנת לקיים את כל התנאים. הקפד לאתחל את הסמפורים בצורה נכונה. עליך להשתמש בפקודות של סמפורים: `sem_init(sem,n), up(sem), down(sem)` בלבד.

יש לרשום 3 קטעים: אתחולים, **P4, P3**.

שאלה 3 (16 נק')

למרכז חישובים מגיעות 5 מטלות batch: A, B, C, D, E כמעט בו-זמנית (A הגיעה ראשונה, B שניה, וכך הלאה, אך כולן באותה פעימת שעון). זמני ריצה משוערים ועדיפויות אשר נקבעו בצורה חיצונית הן כדלקמן:

Job:	A	B	C	D	E
Running time (s):	10	6	2	4	8
Priority:	3	5	2	1	4

העדיפות הגבוהה הינה 5.

נתונים 4 אלגוריתמי תזמון:

1. preemptive round-robin עם quantum של 1s
2. non-preemptive priority scheduling
3. non-preemptive first-come first-served
4. non-preemptive shortest job first

חשבו את זמן ה-turnaround הממוצע (mean process turnaround time).

יש להראות את החישוב!

שאלה 4 (18 נק')

בכל אחד מן הסעיפים, רשמו נכון / לא נכון (1 נק') והסבירו (2 נק'):

- א. בזיכרון וירטואלי, דפים קטנים יותר גורמים לטבלאות דפים קטנות יותר.
- ב. דפים קטנים יותר, גורמים ליותר החטאות ב-TLB.
- ג. דפים קטנים יותר, גורמים לפחות page faults.
- ד. "זול" יותר לבנות תהליכונים בגרעין של מערכת ההפעלה, מאשר תהליכונים ב-user space.
- ה. חסימת תהליכון ברמת המשתמש (user space) גורמת לחסימת התהליך.
- ו. כל התהליכונים אשר מתוזמנים ע"י הגרעין הינם בעלי זיכרון וירטואלי משותף.

שאלה 5 (16 נק')

להלן מספר ייצוגים שונים למיפוי הבלוקים המוקצים לקובץ בדיסק בהם נתקלנו במהלך הקורס:

1. הקצאה רציפה של הדיסק
2. רשימה מקושרת (כל בלוק מכיל מצביע לבלוק הבא)
3. I-nodes
4. FAT

עבור כל ייצוג, כתבו ונמקו בקצרה האם הוא יעיל לגישות מן הסוג הבא:

- (a) קריאה סדרתית של הקובץ מתחילתו ועד סופו
(b) גישה אקראית (random access) לבלוק מסוים בקובץ

הניחו כי התהליך המבצע את הגישה לקובץ הוא היחיד שרץ במערכת.

שאלה 6 (10 נק')

נתון אלגוריתם מניעה הדדית עבור שני תהליכים — strict alternation:

<pre>while (TRUE) { while (turn != 0) /* loop */ ; critical_region(); turn = 1; noncritical_region(); }</pre>	<pre>while (TRUE) { while (turn != 1) /* loop */ ; critical_region(); turn = 0; noncritical_region(); }</pre>
--	--

(a)

(b)

האם האלגוריתם חופשי מהרעבה (starvation-free)? **נמק.**

בהצלחה!