

Operating Systems – 234123

**Homework Exercise 2 – Dry**

Teaching Assistant in charge:

**Akiva Block**

Assignment Subjects & Relevant Course material

**Modules, Scheduling (Lectures 4--5, Tutorials 4--5)**

**Submission Format**

1. Only typed submissions in PDF format will be accepted. Scanned handwritten submissions will not be graded.
2. The dry part submission must contain a single PDF file named with your student IDs –

3. The submission should contain the following:
  - a. The first page should contain the details about the submitters - Name, ID number and email address.
  - b. Your answers to the dry part questions.
4. Submission is done electronically via the course website, in the **HW2 Dry** submission box.

## Grading

1. All question answers must be supplied with a full explanation. Most of the weight of your grade sits on your explanation and evident effort, and not on the absolute correctness of your answer.
2. Remember – your goal is to communicate. Full credit will be given only to correct solutions which are clearly described. Convolved and obtuse descriptions will receive low marks.

## Questions & Answers

- The Q&A for the exercise will take place at a public forum Piazza **only**. Please **DO NOT** send questions to the private email addresses of the TAs.
- Critical updates about the HW will be published in **pinned** notes in the piazza forum. These notes are mandatory and it is your responsibility to be updated.

A number of guidelines to use the forum:

- Read previous Q&A carefully before asking the question; repeated questions will probably go without answers
- Be polite, remember that course staff does this as a service for the students
- You're not allowed to post any kind of solution and/or source code in the forum as a hint for other students; In case you feel that you have to discuss such a matter, please come to the reception hour
- When posting questions regarding **hw2-dry**, put them in the **hw2-dry** folder.

## Late Days

- Please **DO NOT** send postponement requests to the TA responsible for this assignment. Only the **TA in charge** can authorize postponements. In case you need a postponement, please fill out the attached form:

<https://forms.gle/575XhDwcwmfBWzEN7>

## חלק 1 - שאלות בנושא התרגיל הרטוב (50 נק')

מומלץ לקרוא את הסעיפים בחלק זה לפני העבודה על התרגיל הרטוב, ולענות עליהם בהדרגה תוך כדי פתרון התרגיל הרטוב.

1. (6 נק') מה עושה פקודת `yes` בלינוקס? מה הארגומנטים שהיא מקבלת? היעזרו ב-`man page`, ולאחר מכן השתמשו בפקודה ב-`shell` שלכן כדי לבדוק.

2. (6 נק') מדוע השתמשנו בפקודת `yes` עם מחרוזת ריקה במהלך הפקודה הבאה?

```
>> yes '' | make oldconfig
```

נסו להריץ את הפקודה `make oldconfig` לבדה והסבירו מה הבעיה בכך.

3. (6 נק') מה משמעות הפרמטר `GRUB_TIMEOUT` בקובץ ההגדרות של `GRUB`?

```
GRUB_TIMEOUT=5
```

הסבירו מה היתרונות ומה החסרונות בהגדלת הפרמטר `GRUB_TIMEOUT`.

4. (6 נק') מדוע הפונקציה `run_init_process()` אשר נמצאת בקובץ `init/main.c` בקוד הגרעין קוראת לפונקציה `do_execve()` במקום לקריאת המערכת `execve()`?

```
944 static int run_init_process(const char *init_filename)
945 {
946     argv_init[0] = init_filename;
947     return do_execve(getname_kernel(init_filename),
948                     (const char __user *const __user *)argv_init,
949                     (const char __user *const __user *)envp_init);
950 }
```

נסו להחליף את הפונקציות זו בזו ובדקו האם הגרעין מתקמפל.

5. (6 נק') מה עושה קריאת המערכת `syscall()`? כמה ארגומנטים היא מקבלת ומה תפקידם? באיזו ספריה ממומשת קריאת המערכת `syscall()`? היעזרו ב-`man page` בתשובתכן.

6. (10 נק') מה מדפיס הקוד הבא? האם תוכלו לכתוב קוד ברור יותר השקול לקוד הבא?

```
int main() {
    long r = syscall(39);
    printf("sys_hello returned %ld\n", r);
    return 0;
}
```

רמז: התבוננו בקובץ `arch/x86/entry/syscalls/syscall_64.tbl` בקוד הגרעין.

7. (10 נק') התבוננו בתוכנית הבדיקה `test1.c` שסופקה לכן והסבירו במילים פשוטות מה היא בודקת:

```
int main() {
    int x = get_weight();
}
```

```
cout << "weight: " << x << endl;
assert(x == 0);
x = set_weight(5);
cout << "set_weight returns: " << x << endl;
assert(x == 0);
x = get_weight();
cout << "new weight: " << x << endl;
assert(x == 5);
cout << "==== SUCCESS =====" << endl;
return 0;
}
```

## חלק 2 - זימון תהליכים (50 נק')

1.

- a. (4 נק') אילו משיטות התזמון הבאות עלולות לגרום לאפקט השיירה (convoy effect)? (שימו לב שעשויה להיות יותר מתשובה אחת נכונה)
- i. FCFS
  - ii. RR
  - iii. SRTF
  - iv. כל מדיניות זימון עם הפקעה (preemption)
  - v. כל מדיניות זימון בלי הפקעה (preemption)
  - vi. אף תשובה אינה נכונה

- b. (3 נק') כפי שלמדנו, תחת תנאים מסוימים אלגוריתם SJF הינו אופטימלי עבור מדד זמן תגובה ממוצע. מהם שלושת התנאים?

2. (7 נק') כזכור, בתרגול ראיתן נוסחה לקצב התקדמות הזמן הווירטואלי ונוסחה לחישוב הקוונטום של כל תהליך. בסעיף זה ננסה לבחון את הקשר בין 2 הנוסחאות. הניחו כי:

- a. בסוף כל epoch לכל התהליכים יש את אותו זמן וירטואלי.  
b. הזמן הווירטואלי מתקדם ביחס הפוך לעדיפות התהליך:  $VR_i += \frac{C}{w_i} dT$  עבור  $C > 0$  כלשהו.

- c. אורך ה-epoch הוא קבוע  $Q_i = sched\_latency$ .

הראו כי הנחות אלו גוררות את הנוסחה שראינו בתרגול לחישוב הקוונטום של כל תהליך:

$$Q_i = \left( \frac{w_i}{\sum_j w_j} \right) \cdot sched\_latency$$

הערה: שימו לב כי הנחות אלו הן הנחות מקלות שנועדו להקל עליכן לפתור את התרגיל. הנוסחה שהוצגה בתרגול נכונה כמובן גם ללא הנחות אלו, שאינן נכונות במקרה הכללי.

3. (24 נק') הילה, מומחית עולמית לאלגוריתמי זימון, החליטה לנסות לשפר את אלגוריתם CFS המוכר. הילה הציעה שיפורים שונים לאלגוריתם. בכל אחד מהסעיפים הבאים, מוצע שינוי/שיפור לאלגוריתם CFS. כתבו חיסרון שנקבל כתוצאה מאותו שינוי.

- a. (4 נק') תהליך שחזר מהמתנה יישאר עם אותו זמן וירטואלי שהיה לו כשהוא יצא להמתנה.  
b. (4 נק') שימוש במבנה נתונים של רשימה במקום עץ אדום-שחור.  
c. (4 נק') הסרת המינימום על גודל הקוונטום של תהליך (min\_granularity)  
d. (4 נק') הגדרת הקוונטום של כל תהליך, להיות בלתי תלוי במספר התהליכים, ובפרט  $Q_i = a \cdot W_i$  (עבור קבוע חיובי  $a$  כלשהו)  
e. (4 נק') שינוי קצב התקדמות הזמן הווירטואלי להיות מהצורה  $VR_i \propto \frac{1}{\sqrt{w_i}}$   
f. (4 נק') תהליכים חדשים נכנסים לתור הריצה עם זמן וירטואלי השווה לזמן הווירטואלי הממוצע של שאר התהליכים הקיימים במערכת.

4. (6 נק') שירי, קולגה של הילה, רצתה להפחית את התקורה של חימום מטמונים בעת החלפת הקשר. לכן היא הציעה שתהליכים שמוכנים לריצה וחיים באותו מרחב זיכרון כמו התהליך שרץ אחרון על המעבד, יקבלו עדיפות על המעבד וירוצו לפני כל התהליכים האחרים. עומר, סטודנט למערכות הפעלה, טוען שהפתרון בעייתי כי בצורה זו תהליך יכול לגנוב זמן מעבד ולהרעib תהליכים אחרים. הסבירו כיצד.

5. (6 נק') נזכיר, שבאלגוריתם CFS, לכל תהליך יש משקל שנקבע לו בהתאם לערך ה-nice שלו (6 נק') נזכיר, שבאלגוריתם CFS, לכל תהליך יש משקל שנקבע לו בהתאם לערך ה-nice שלו (19 ≤ nice ≤ -20):

```
static const int prio_to_weight[40] = {
/* -20 */ 88761, 71755, 56483, 46273, 36291,
/* -15 */ 29154, 23254, 18705, 14949, 11916,
/* -10 */ 9548, 7620, 6100, 4904, 3906,
/* -5 */ 3121, 2501, 1991, 1586, 1277,
/* 0 */ 1024, 820, 655, 526, 423,
/* 5 */ 335, 272, 215, 172, 137,
/* 10 */ 110, 87, 70, 56, 45,
/* 15 */ 36, 29, 23, 18, 15,
};
```

כמו כן, ראינו כי גודל הקוונטום של תהליך, נמצא ביחס ישר למשקל התהליך:

$$Q_i = \left( \frac{w_i}{\sum_j w_j} \right) \cdot \text{sched\_latency}$$

לכן, לתהליך בעל ערך nice נמוך, יש משקל גבוה, וכתוצאה מכך קוונטום ארוך יותר. בסעיף זה, נרצה לבחון את הרציונל מאחורי בחירת ערכי המשקלים.

a. (3 נק') ציירו (בעזרת אקסל או כל תוכנה אהובה לבחירתכן) גרף של לוג משקל התהליך (בבסיס 1.23), כפונקציה של ערך ה-nice המתאים לה. צרפו תמונה של הגרף לפתרון.

b. (3 נק') על סמך הגרף, מהו הקשר המתמטי בין ערך ה-nice למשקל התהליך?