

Operating Systems – 234123

## **Homework Exercise 2 – Dry**

Teaching Assistant in charge:

**Alex Zhybirov**

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 –  
**123456789\_300200100.pdf**
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. Convolute 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

- Only the TA in charge of the assignment can authorize postponements. In case you need a postponement, please fill out the attached form: <https://forms.gle/C4aU2Gv3QDA7xohS8>

מvlaה זו מנוסחת בלשון זכר, אך מיועדת לשני המינים כאחד

## חלק 1 - שאלות בנושא התרגיל הרטו (30 נק')

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

1. מה עושה פקודה yes בLinux? מה הארגומנטים שהוא מקבלת? (3 נק')  
היעזרו בחマン, ולאחר מכן השתמשו בפקודה bashctl שלכם כדי לבדוק.

2. מדוע השתמשנו בפקודת yes עם מחרוזת ריקה במהלך הפקודה הבאה? (3 נק')

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

נסו להריץ את הפקודה make oldconfig לבדה וסבירו מה הבעיה בכאן.

3. מהמשמעות הפורטט GRUB\_TIMEOUT בקובץ ההגדרות של GRUB? (3 נק')

```
GRUB_TIMEOUT=5
```

סבירו מה היתרונות ומה החסרונות בהגדלת הפורטט GRUB\_TIMEOUT.

4. מדוע הפונקציה run\_init\_process() חייה אשר נמצאת בקובץ c/main/init.c בקוד הגרעין קוראת לפונקציה do\_execve() במקום לקרוא את המערכת? (3 נק')

```
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 בזה ובדקו האם הגרעין מתקין מפהל.

5. מה עושה קריאת המערכת syscall() sys? כמה ארגומנטים היא מקבלת ומה תפקידם? באיזו ספריה ממומשת קריאת המערכת syscall() sys? היעזרו בחמן בתשובתכם. (3 נק')

6. מה מדפיס הקוד הבא? כתבו קוד ברור יותר השקול לקוד שלפניכם. (3 נק')

```
int main() {
    syscall(62, syscall(39), 9);
    return 0;
}
```

רמז: התבוננו בקובץ arch/x86/entry/syscalls/syscall\_64.tbl בקוד הגרעין.

7. ביצעו את השורה הבאה. ציינו 2 ערכים אפשריים של s, וסבירו מה משמעות כל ערך. (3 נק')

```
int s = capable(CAP_SYS_ADMIN);
```

8. נרצה להרחיב את קריית המערכת `set_ban` כך שתוכל לחסום עוד קריאות מערכת. החתימה החדשיה:

```
long set_ban(int getpid_ban, int pipe_ban, int kill_ban,  
             int a, int b, int...)
```

כמה קרייאות מערכת שונות סך הכל יוכל לחסום בדרך זו? אם קיימת הגבלה על מספר זה, הסבירו בפירוט כמה היא נובעת. (3 נק')

9. התבוננו בתוכנית הבדיקה `test1.cxx` שסופקה לכם והסבירו במילים פשונות מה היא בודקת (6 נק'):

```
int main() {  
    long x = get_ban('g');  
    cout << "getpid() should not be banned initially (0) -> " <<  
x << endl;  
    assert(x == 0);  
  
    x = set_ban(1,1,0);  
    cout << "set_ban(110) returned: " << x;  
    if(x == -1) cout << " (did you use sudo?)";  
    cout << endl;  
  
    assert(x == 0);  
    x = get_ban('g');  
    cout << "getpid() should be banned (1) -> " << x << endl;  
    assert(x == 1);  
  
    x = flip_ban_branch(1, 'k');  
    cout << "set_flip_branch returned: " << x << endl;  
    assert(x >= 0);  
  
    x = check_ban(pid_t(getppid()), 'k');  
    cout << "Our parent should be banned from using kill() (1) ->  
" << x << endl;  
    assert(x == 1);  
  
    x = getpid();  
    cout << "getpid() returned: " << x << endl;  
  
    ;cout << "===== SUCCESS =====" << endl  
    ;return 0  
}
```

## חלק 2 - זימון תהליכיים (50 נק')

במבחן עם מעבד בן 128 ליבוט רץ תהליך אנטו וירוס ייעודי (בשאלה נקרא לו AV) שתפקידו לניהל רישום של כל התהליכים שרצו אי פעם במחשב. AV הינו תהליך משתמש רגיל עם שתי תוכנות מיוחדות: (i) הנחמדות שלו היא 15-. (ii) מטעמי בטיחות הוא חייב להיות runnable לפחות כל הזמן. משום כך, גרעין הלינוקס של המחשב עודן להתריע מיד אם AV מפסיק להיות runnable לפחות זמן מסוים מאיזושהי סיבה, ע"י שליחת הודעה לאחראי הביטחון של החברה. בלתי אפשרי לחסום את ההתרעות של הגרעין ולז"פ או לשנות את הרישום של AV מנהל. מובטח לנו שכאשר AV רץ הוא מודע לכל התהליכים הקיימים במערכת באופןו הרגוע, בכך לניהל רישום מדויק.

שירה היא עובדת חברת בצוות ה-TA בעלת הרשות אדמיניסטרטור על המחשב הנ"ל. היא גם סוכנת עיינת שהוצאה להריצ' קובץ זדוני בשם sauron.exe בסוד, מבלי שהרצה זו תופיע ברישומיו של AV או שתשלוח הודעה לאחראי הביטחון. על המערכת להמשיך לפעול באופן תקין בתום ההרצה של הקובץ/zdoni. בשאלת זו תעזרו לשירה לבצע את זמנה ולהריץ את sauron.exe על מחשב החברה. העורות:

1. שימו לב לשירה לא יודעת את שם התהליך AV או את הPID שלו.
2. לא תוכלו להשתמש במודולים כדי לפטור את השאלה.
3. מותר להשתמש בקריאת המערכת הבאה, או בכל קריית מערכת אחרת שלמדתם עד כה בקורס:

```
// The process with ID 'pid' will run on only the CPU core with ID  
// 'core_id'.
```

```
int sched_setaffinity(int pid, int core_id /*between 0 to 127*/);
```

(הערה – זהה גרסה מופשטת של קריית המערכת האמיתית)

1. עם איזה ערך נחמדות נרץ את sauron.exe כדי שלא יתגלה ע"י AV? (5 נק')

- a. nice > -20
- b. כל ערך nice יעבד
- c. nice = -20
- d. שום ערך nice לא יעבד
- e. nice = 0

nymok:

---

---

---

2. באילו מהקרים הבאים תהליך שרע במדיניות זימון SCHED\_FIFO מועתר על המעבד? (5 נק')

- a. התהיליך יוצא לתור המתנה
- b. התהיליך קורא ל getpid()
- c. לתהיליך נגמר הקוונטום
- d. הגיע לתהיליך SCHED\_FIFO נוסף עם עדיפות זהה
- e. אחד מבניו מסיים לרוץ

רימוק:

---

---

---

3. חסימת הליבות (10 נק')

שירה התחליה לרשום את קוד הפריצה אבל היא צריכה את עזרתכם כדי להשלים אותו.

השלימו את הקוד כך שיחשב את כל הליבות במערכת – חוץ מליבה 0 (עליה שירה תרים את sauron.exe.).

```
sched_setaffinity(getpid(), 0);
struct sched_param param;
param.sched_priority = ____;
sched_setscheduler(getpid(), _____, &param);

pid_t p;
pid_t* children_pids = malloc(sizeof(pid_t) * ____);
for (int i=1;i<____;i++) {
    p = fork();
    if (p==0) {
        _____;
    }
    children_pids[i-1] = p; // we save the pids of all children for later
}
```

#### 4. הרכבת sauron.exe ומחיקת עקבות (10 נק')

בזמן שהשלמתם את הקוד, שירה עברה על קוד האסמבלי של sauron.exe וגילתה שהוא לא יוצא להמתנות, ולא קורא ל(`yield`). השלימו את הקוד הבא כך שsauron.exe יירוץ מבלי להתגלות ע"י AV.

זכרו – עליוכם להשאיר את המערכת במצב תקין, ולהשאיר מינימום עקבות במערכת. לרשותכם כל המשתנים ששמरתם עד כה בתרגיל (לדוגמא `children_pids`).

מכאן והלאה יהיו יותר שורות ריקות מהנדרש.

```
pid_t sauron_pid = fork();  
if (sauron_pid==0) {  
  
}  
else {  
  
}  
return 0; // sauron executed on the system undetected
```

#### 5. טיפול בהמתנות (20 נק')

זמן קצר לאחר שסויימתם להריץ טיסטים על הקוד שלכם, המפעלים של שירה מסרו לכם גרסא חדשה של sauron.exe. שירה בדקה את הגרסא ולצערכם הרוב היא מכילה יציאות להמתנה. כתע בכל הסימולציות של שshireה מריצה AV מצליח לרשום את sauron.exe. למה הקוד שתכתבם הפסיק לעבוד?

---

---

---

שעת השין מתקרבת, ועליכם לסיים את המלאכה. ממשו מחדש את הרכבת sauron.exe, הפעם בצווארה שמנועת מאן לרוץ על מעבד במקרה בו sauron.exe י יצא להמתנה.

```
pid_t sauron_pid = fork();  
if (sauron_pid==0) {  
  
}  
else { // should hold the core while sauron.exe is not actively running  
  
}  
return 0; // sauron executed on the system undetected
```

שירה מורידה את הקוד ל-USB רגע אחריו שסימתם לכתוב אותו. היא נועעת להריץ אותו על מחשב החברה מרובה הליבות בתקווה שהוא יבצע את המשימה בסודיות הנדרשת. לכמ' נשאר לחכות ולקוות שהפעם הבאה שתראו אותה לא תהיה בחדשות, אלא בקריבים.

## העשרה

- לפתרון שלכם חסרים כמה פרטיים קטנים (מחוץ לסקופ הקורס), שם תשלימו אותם תוכלו לנסוט את הקוד הנ"ל על המחשב/מכונה וירטואלית.
1. בפועל, לנוקס לא מאפשרת לתהיליכי RT להשתלט לגמרי על ליבות (כדי למנוע מצב בו המערכת נתקעת). לנוקס מאפשרת לתהיליכי RT מיקסimum של 95% זמן המעבד עליו הם רצים. ניתן לשנות את הפרמטר זהה, ואז הקוד שתכתבם יעבד כמצופה. הפרמטר נמצא ב-"/proc/sys/kernel/sched\_rt\_runtime" (钜, מטרו 1,000,000.000 ל-1). כדי לאפשר לתהיליכי RT שליטה מוחלטת על הליבה שלהם. לא לשוכח לשנות בחרזה!
  2. השתמשו בפונקציה (N)N\_SC\_NPROCESSORS\_ONLN sysconf כדי לקבל את כמות המעבדים הפעילים על המחשב שלכם.
  3. קראו באינטראנט/man על הפונקציה sched\_setaffinity, ההגדירה שלה במציאות שונה מהגרסת המופשטת שהשתמשתם בה בשאלת.
  4. שימו לב שהרצה לא מושכלת של הקוד עלולה לגרום למחשב שלכם להתקע (למשל, אם exe sauron.exe הוא לולאה איןסופית).

### חלק 3 - מודולים (20 נק' + 10 בונוס)

1. ממשו דרייבר להתקן פיקטבי, בעל המפרט הבא:

- I) בפתיחת (open) ה

# התקן

 יודeo "READY".
  - II) בעת קרייה מההתקן, יוחזר 'f'.
  - III) בעת כתיבה להתקן, נדועו שלא יותר מקום בהתקן.
  - IV) במידה ולהתקן soho 8 יוחזר בקרייה '8' במקום 'f'.

## הערות:

- ממשו רק את `sfs_ops` + הפונקציות הנלוות לו (... `open`, `read`, `write`, ...), ניתן להשמיט חלקים כמו `module_init`, `module_exit` וכו'.
  - במהלך המימוש תדרשו לכתוב מידע לחוצצים במרחב המשתמש, מרחיב הגauważן. דבר זה אינו טריויאלי כי שנלמד בהמשך הקורס, ונitin לעשות זאת ע"י שימוש בפונקציה הבאה:

```
void put_user(char data, char* usr_ptr);
```

**בונוס (10 נק')**

2. נניח שבחלך הקודם יכולתם לשימוש במודולים בפתרון שלכם.  
הסבירו כיצד הייתם משתמשים במודולים כדי להריץ את exe.sauron מבלי להתPOSE (לא חייב לכתוב קוד).  
ניתן להניח שאין הרבה תהליכי עלי נחמדות 15 - על מחשב היעד, ולשנות את דרך פעולה זמנית לא יציק  
למערכת.

---

---

---

---

---

בשםחה תמיד,  
סגל מערכות הפעלה