

## שאלה 2 סעיף 1-

קיימות שיטות רבות לבדוק האם תהליך מסוים חי, אפרט מתוכן שתיים:

### (1) `ps -p <PID>`

אם התהליך חי, הערך שיוחזר לאחר הרצת הפקודה יהיה ה-Pid של התהליך, מאיזה טרמינל התבצע התהליך, זמן המעבד שהתוכנית תפסה ואת הפקודה ששימשה להפעלת התהליך. למשל- ניתן לראות בתמונה המצורפת מטה, כי תהליך מס 4860 חי ולכן קיבלנו עבורו בשורה שמתחת את ה-pid שלו. לעומת תהליך מס 2003 שעבורו לא קיבלנו זאת.

```

yarden@osboxes: ~/fwork_207205972/q_2/q20
1439 yarden 20 0 392016 76856 42524 S 2.0 1.5 0:21.62 Xorg
3924 yarden 20 0 958220 47752 34536 S 1.7 0.9 0:02.92 gnome-ter+
1703 yarden 20 0 2843736 361236 115696 S 1.0 6.9 1:54.61 gnome-she+
782 mongod 20 0 998436 81668 33156 S 0.7 1.6 0:35.66 mongod
1430 yarden 9 -11 1409256 18796 14928 S 0.7 0.4 0:01.02 pulseaudio
4860 yarden 20 0 11896 3916 3156 R 0.7 0.1 0:00.03 top
1550 yarden 20 0 91596 2136 1676 S 0.3 0.0 0:17.23 VBoxClient
1 root 20 0 167228 11008 7972 S 0.0 0.2 0:02.68 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd
3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_gp
4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_par_gp
6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0+
9 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu+
10 root 20 0 0 0 0 S 0.0 0.0 0:00.25 ksoftirqd+
11 root 20 0 0 0 0 I 0.0 0.0 0:02.08 rcu_sched
12 root rt 0 0 0 S 0.0 0.0 0:00.07 migration+
13 root -51 0 0 0 S 0.0 0.0 0:00.00 idle_inje+

[21]+ Stopped top
yarden@osboxes:~/fwork_207205972/q_2/q20$ ps -p 4860
PID TTY TIME CMD
4860 pts/1 00:00:00 top
yarden@osboxes:~/fwork_207205972/q_2/q20$ ps -p 2003
PID TTY TIME CMD
yarden@osboxes:~/fwork_207205972/q_2/q20$
  
```

### יתרונות:

- דרך מהירה וקלה.
- לא נדרשת תוכנית מעטפת כדי לבדוק זאת.
- כדי לבדוק אם התהליך חי לא נדרש מידע נוסף על התהליך אלא רק את מס' pid שלו.
- מקבלים מידע נוסף על אותו תהליך ולא רק האם הוא חי/מת.

חסרונות:

- אם התהליך קיים לא נוכל לדעת האם יש לנו הרשאה לשלוח לו signal.
- בשיטה זו חייב שיהיה לנו את ה-pid של התהליך כדי לבדוק את הסטטוס שלו ולא ניתן לבדוק זאת בדרכים אחרות.

**Top (2**

פקודה זו מחזירה את מספרי כל התהליכים שיש להם שימוש רב במעבד/זיכרון/דיסק. בגלל שכבר קיבלנו את מס' התהליך, נוכל להריץ את הפקודה top בטרמינל ולבדוק האם תהליך זה נמצא ברשימה.

יתרונות:

- מתבצעת ע"י פקודה אחת בלבד.
- נניח ונרצה לבדוק יותר מתהליך אחד- על אותה פקודה נוכל לבדוק כמה תהליכים והאם הם חיים בלי לכתוב פקודה נוספת.
- לא נדרשת תוכנית מעטפת כדי לבדוק זאת.
- הפקודה מעדכנת כל כמה שניות את התהליכים שרצים במערכת ההפעלה וזהו יתרון גדול.

חסרונות:

- אם התהליך קיים לא נוכל לדעת האם יש לנו הרשאה לשלוח לו signal.
  - בשיטה זו חייב שיהיה לנו את ה-pid של התהליך כדי לבדוק את הסטטוס שלו ולא ניתן לבדוק זאת בדרכים אחרות.
  - שיטה זו מחזירה לנו את כל ה-pid של התהליכים ולכן זה יכול לקחת זמן לחיפוש pid של התהליך הרצוי.
-

בתוכנית check\_pid.c השתמשתי בשיטה `kill(pid,0)` כדי לבדוק האם תהליך חי. אפרט על היתרונות והחסרונות עבור שיטה זו –

#### יתרונות:

- נותן את סוג השגיאה במידה ונכשל.
- באמצעות שיטה זו ניתן לדעת האם התהליך קיים ובנוסף - האם קיימת \לא קיימת הרשאה לשלוח לו סיגנל .
- במידה וה-kill נכשל נוכל לקבל אינדיקציה מדוע זה קרה.
- גם בלי הרשאה לשליחת סיגנל לתהליך מסוים, נוכל לדעת אם תהליך קיים.

#### חסרונות:

- שיטה זו מחייבת שיהיה לנו את pid של התהליך. כלומר, לא ניתן לגלות את הסטטוס שלו לפי משהו אחר שהוא לא pid.
- שיטה זו מחייבת שתהיה לנו תוכנית מעטפת שאותה נקמפל בטרמינל. כלומר, `kill(pid,0)` זו לא פקודה אלא פונקציה שנשתמש בה בתוכנית שכתבנו .
- שליחת סיגנל ללא הרשאה לתהליך מסוים, עלולה לגרום לשגיאה במידה ולא נתפוס אותה .