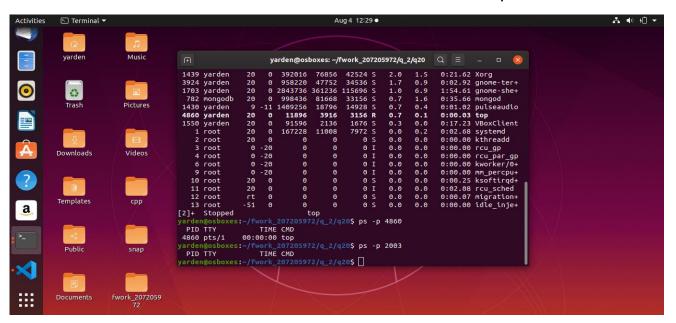
-1 שאלה 2 סעיף

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

ps -p <PID> (1

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.



יתרונות:

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

<u>חסרונות:</u>

- אם התהליך קיים לא נוכל לדעת האם יש לנו הרשאה לשלוח לו signal.
- של התהליך כדי לבדוק את pid- בשיטה זו חייב שיהיה לנו את ה-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** הpid של התהליכים ולכן זה יכול לקחת pid של התהליך הרצוי.

בתוכנית check_pid.c השתמשתי בשיטה check_pid.c 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 מסויים, נוכל לדעת אם תהליך קיים.

חסרונות:

- שיטה זו מחייבת שיהיה לנו את הpid של התהליך. כלומר, לא ניתן pid לגלות את הסטטוס שלו לפי משהו אחר שהוא לא
- 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 .