

## שיטה kill :

יתרונות:

- נותן את התוצאה הנכונה .

- קל לתפעול.

חסרונות:

- נוכל לאבד סיגנלים בדרך כי לא תמיד כל ה signals מגיעים בתור.

- במידה והגישה נדחתה לא אומר מדוע הגישה נדחתה. (יכול להיות כמה סיבות)

## שיטה getpgid :

שיטה נוספת לגלות אם process עם id מסויים קיים , השיטה מקבלת pid ומחזירה את ה group id של אותו process אז במידה וקיבלנו מספר שגדול שווה ל 0 זה אומר שה process אכן קיים , במידה וקיבלנו אותו process id כמו זה ששלחנו לשיטה , זה אומר שאנחנו ב process יחיד וה group process id שווה ל pid ששלחנו.

מקור : <https://stackoverflow.com/questions/35173897/how-does-getpgid-work>

יתרונות:

- במידה ויש שגיאה גם פה ה errno מתמלא בהתאם .

חסרונות:

- תמיד נקבל את ה group process id , במידה וקיים רק process אחד אז הוא יהיה זהה ל pid במידה וקיימים כמה הוא לא יהיה זהה , ואז לא נקבל במדויק את ה pid ששלחנו.

- במידה והגישה נדחתה גם פה ה errno מתמלא ב EPERM אבל לא מפרט יותר מדי (יכול להיות כמה סיבות)

## שיטה /proc/pid :

בשיטה זו אפשר לגשת לתוך הקבצים של process ח' , במידה ונריץ ב bash : /proc/500 וה process עם ה id יהיה קיים נכנס לתוך התיקייה של אותו process במידה והפקודה תחזיר "No such file or directory" נדע שאותו process אינו קיים.

כדי להריץ בקוד c את הפקודה הנ"ל נשתמש בפונקציה system שנמצאת בספריה התקנית של c

מקור - [https://www.tutorialspoint.com/c\\_standard\\_library/c\\_function\\_system.htm](https://www.tutorialspoint.com/c_standard_library/c_function_system.htm)

יתרונות:

- זיהוי מהיר האם קיים process כזה.

חסרונות:

- גישה לקבצים משמעותיים של אותו process , וע"י טעות כלשהיא ניתן יהיה לפגוע באותו process שרץ.

מקור: <https://www.geeksforgeeks.org/proc-file-system-linux/>

אלה הקבצים שנוכל לראות בתוך התיקייה של ה process :

DIRECTORY	DESCRIPTION
/proc/PID/cmdline	Command line arguments.
/proc/PID/cpu	Current and last cpu in which it was executed.
/proc/PID/cwd	Link to the current working directory.
/proc/PID/envIRON	Values of environment variables.
/proc/PID/exe	Link to the executable of this process.
/proc/PID/fd	Directory, which contains all file descriptors.
/proc/PID/maps	Memory maps to executables and library files.
/proc/PID/mem	Memory held by this process.
/proc/PID/root	Link to the root directory of this process.
/proc/PID/stat	Process status.
/proc/PID/statm	Process memory status information.
/proc/PID/status	Process status in human readable form.