בס"ד

עבודה מסכמת – מערכות הפעלה

שאלה 2 סעיף 1:

החסרון של kill(<pid>,0) (**ובאופן כללי חסרון זה שייך גם לשיטות הנוספות שאציין בהמשך**) הוא שזה לא בטוח לשימוש (race conditions), יכול להיות שהתהליך שאליו התכוונו הסתיים וישר התחיל תהליך אחר עם הpid של התהליך שחיפשנו ולכן זה מראה שהתהליך שחיפשנו קיים למרות שהוא בעצם לא קיים (אלא קיים תהליך חדש עם הpid של התהליך שחיפשנו), או שיכול להיות שהתהליך יסתיים מיד אחרי שבדקנו ומבחינתנו הוא עדיין קיים.

זאת אומרת שיכול להיות שיהיה לנו מידע לא אמין ונבצע על פיו פעולות מסויימות.

– kill(<pid>,0)

יתרונות –

- פשוט יחסית.
- עובד גם במקרים שהרשאות הגישה לספרייה proc הוגדרו להיות ברמת אבטחה 2 על ידי שימוש ב hidepid (stat לא עובד במקרה הזה).
- במידה והתהליך קיים, שיטה זו מספקת מידע האם ניתן לשלוח סיגנלים לתהליך זה (אם התקבלה הודעה מסוג EPERM זה אומר שהתהליך קיים אבל אין הרשאה לשלוח לו סיגנלים כאלה).

חסרונות –

- race conditions.
- לא מספק הרבה מידע על התהליך (בעיקר ביחס לשיטה של stat שמופיעה בהמשך).

שיטות נוספות לבדיקה האם תהליך קיים:

1) בעזרת הפונקציה getpgid(pid) – אם ערך החזרה גדול או שווה לאפס התהליך קיים (ערך החזרה זה הGID של אותו תהליך שעליו אנחנו בודקים), אחרת, התהליך לא קיים.

יתרונות –

- יותר אלגנטי ופשוט, שורה אחת של קוד.

חסרונות –

- אין פירוט, לא מספק הרבה מידע על התהליך.

2) בעזרת הפונקציה stat("proc/<pid>", &sts) –

```
if (stat("/proc/<pid>", &sts) == -1 && errno == ENOENT)
```

אם ה if הזה מתקיים אז התהליך לא קיים.

יתרונות –

- מספק הרבה מידע על התהליך.

חסרונות –

- לא ניתן לביצוע עבור ספריות proc שהוגדרו להיות ברמת אבטחה 2 על ידי שימוש ב hidepid (עדיין אפשר לבדוק בעזרת kill).
- לוקח הרבה זמן (צריך לעבור על הספרייה proc).