

Q2:

(1) מצורפת תמונה שמראה כי לא כל הסיגנלים שנשלחו – התקבלו.

הסיבה לכך הינה בשל עומס בתור , וכאשר ישנו סיגנל מסוג מסויים שמתקבל בתור- עד שלא יטופל , שאר הסיגנלים מאותו סוג "יזרקו" ולא יכנסו לתור. (יתעלם- ignore) .

במקרה שלנו- נשלחו סיגנלים מסוג SIGINT, ועד שלא טופל הסיגנל הראשון שבתור- כל הסיגנלים מסוג SIGINT "יזרקו לפח" ולא ייכנסו לתור. רק כאשר יסתיים הטיפול בסיגנל שבתור- יוכל להיכנס סיגנל נוסף מסוג SIGINT לתור. ולכן- ישנם סיגנלים ש"נאבדים בדרך" מ client ל server.

כאשר אנחנו שולחים סיגנל יחיד מסוג SIGUSR1 הוא מדפיס לנו את מספר הסיגנלים מסוג SIGINT שהתקבלו בעזרת ה handler הייעודי לסיגנל זה.

להלן התמונה שממחישה כי לא כל הסיגנלים התקבלו-

מצד ימין- טרמינל שמריץ server

מצד שמאל- טרמינל שמריץ client (וכמובן בודק לפני כן pid של ה server)

```
omr@omni-VirtualBox: ~/Desktop/fwork_206265233/q_2/q_22
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$ pidof server
3709
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$ ./cilent 3709 2 1000
omr@omni-VirtualBox:~/Desktop/fwork_206265233/q_2/q_22$ ./cilent 3709 10 1
```

ניתן לראות כי התקבלו 77 סיגנלים בלבד לאחר שנשלחו 1000. שאר הסיגנלים לא נכנסו לתור.

(2) יתרונות וחסרונות של 2 השיטות : Real time signals vs Standard signals

יתרונות של Real time signals על גבי Standard signals :

Real time signals יכולים להיציבר לתור גם אם הם מאותו הסוג ובכך לא נאבדים סיגנלים ומידע. מה שלא קורה ב Standard signals – שם נצבר רק סיגנל יחיד מכל סוג, ושאר הסיגנלים נאבדים. Real time signals בעלי יכולת לשלוח מידע/מילה עם הסיגנל, מה שלא קיים ב Standard signals .

ב- Real time signals סדר המידע מובטח, ויגיע באותו הסדר שיצא מהמקור. לעומת Standard signals – שם כלל לא מובטח שיגיעו הסיגנלים..

ב Real time signals מתבצעת שמירה של Pid,Uid של התהליך השולח- בשדות שנועדו לכך (si_pid,si_uid שבתוך struct siginfo) מה שלא קיים ב Standard signals דבר שיכול להועיל למתכנתים בביצוע פעולות כאלו ואחרות.

יתרונות של Standard signals על גבי Real time signals :

Standard signals נתמכים בכל kernel של Linux\Unix , לא משנה באיזו גרסה ועד כמה היא ישנה- מה שלא קורה ב Real time signals בהם צריך לפעמים לבצע התאמות מיוחדות ועטיפות API כאלו ואחרות .

Standard signals אינם מצריכים שום מימוש . הם בנויים built in בכל גרסת kernel . מה שב- Real time signals מצריך לפעמים מימוש של תור הסיגנלים.

בLinux , כמו ביישומים רבים אחרים- Standard signals מתועדפים באופן חזק יותר מ Real time signals במקרה שבו שניהם ממתינים לתהליך מסוים, לא משנה "מי הגיע קודם".

עובדות נוספות:

Real time signals הרחיבו את מגוון הסיגנלים למטרות יישום שונות (מספרים 32-64) לעומת המגוון שב Standard signals אשר מכיל את 1-31 בלבד.