תרגיל יצירת תהליכים

process creation exercise

להלן הגדרת תוכנית

- מקבלת מספר שלם X כפרמטר (בשורת הפקודה).
- X חיובי (מייצג הפעלה ראשונה): *התוכנית מריצה את עצמה עם (X-1) כפרמטר.*
- X שלילי (מייצג הפעלה עצמית): *תוכנית מפעילה את עצמה עם (X+1) כפרמטר.*
- אם X הוא 0: *מתחילים לחזור.*
- תהי AP המחרוזת "Arg: <arg> <pid>" כאשר <arg> הוא הפרמטר X (עם סימן) ו <pid> הוא מזהה התהליך של התהליך הנוכחי.

1. **כתוב את התוכנית עם system.**
   הדפס את AP ל stdout לפני הסיום.
2. **כתוב את התוכנית עם exec ו fork.**
   הדפס את AP ל stdout לפני הסיום.
3. **כתוב את התוכנית עם popen.**
   קרא מהצינור והדפס ל stderr (אם יש בן). שלח את AP לתהליך האב לפני הסיום.
4. **כתוב את התוכנית עם fork, exec, pipe.**
   קרא מהצינור והדפס ל stdout (אם יש בן). שלח את AP לתהליך האב לפני הסיום.

תרגיל יצירת תהליכים

process creation exercise

פלטים לדוגמה:

```
$ ./exec 3
Arg: +0 Pid: 27200
Arg: -1 Pid: 27199
Arg: -2 Pid: 27198
Arg: +3 Pid: 27197
```

```
$ ./system 3
Arg: +0 Pid: 27195
Arg: -1 Pid: 27193
Arg: -2 Pid: 27191
Arg: +3 Pid: 27189
```

```
$ ./pipe 3
Arg: +3 Pid: 27209
Arg: -2 Pid: 27210
Arg: -1 Pid: 27211
Arg: +0 Pid: 27212
```

```
$ ./popen 3
Arg: +0 Pid: 27208
Arg: -1 Pid: 27206
Arg: -2 Pid: 27204
Arg: +3 Pid: 27202
```

נסו להבין:

1. מדוע ההפרשים ב pid של התהליכים ב exec ו pipe הם 1 וב system ו popen הם 2?
2. מדוע ב popen היה צורך לבצע את ההדפסה ל stderr?
3. מדוע ב pipe היה צורך לבצע את ההדפסה בסדר הפוך?