

מערכות הפעלה – תרגיל 2

(1)

ראשית אנו באבא כאשר $a=0$

נגיע לשורה השנייה של `main` נבצע `fork` (מס' התהליכים החדשים כעת הוא 1).

כעת נתפצל לשני מקרים:

(1) אבא - `pid` יהיה `pid` של הבן ולכן הוא לא יהיה 0 ולכן נקבל $a+=2$ ונקבל $a=2$ כעת, נגיע לזו אכן $a=2$ ולכן נעשה שוב `for` (מס' התהליכים החדשים כעת הוא 2).

כעת שוב נתפצל ל-2 מקרים:

1.1) אבא - כעת מתבצע $a++$ ונקבל $a=3$ נגיע לזו ואכן $a=3$ ולכן נבצע שוב `fork` (מס' התהליכים החדשים כעת הוא 3) – כעת גם האבא וגם הילד הנוצר יהיו בסוף `main` ולכן יסיימו את פעולתם.

1.2) בן – יתבצע בדומה לאבא (1.1) ולכן ייווצר בסך הכל תהליך חדש אחד נוסף (מס' התהליכים החדשים כעת הוא 4).

(2) בן – `pid` של הבן יהיה 0 ולכן יתבצע $a+=3$ ונקבל $a=3$, כעת עבור `if` הראשון לא ניכנס מכיוון ש a לא שווה 2.

כעת יתבצע $a++$ ונקבל $a=4$ וכעת שוב `if` לא נכנס מכיוון ש a לא שווה 3. ונסיים את התוכנית.

לסיכום יצרנו 4 תהליכים חדשים.

(2)

```
PARENT: x is 150
PARENT: forking
PARENT: forked...
PARENT: forked... (can be either here or after CHILD: 300)
CHILD: happy birthday
CHILD: 300
PARENT: child completed
PARENT: 450
```

המספרים 300 ו-450 יודפסו מכיוון שעבור כל `job` המשתנה `x` יועתק לכל אחד מה `job` ולכן כאשר נשנה אחד, העתק אחר שלו לא ישתנה.

3

quitting

2

יתבצע fork והאבא יחכה יכנס למקרה default בו הוא יחכה לבן שיסיים ויקבל את exit code שלו לתוך סטטוס.
כעת הבן ידפיס "quitting" ולאחר מכן יסתיים באגרסיביות (בעקבות _) עם קוד יציאה 2. אותו יקבל האבא ובאמצעות המקור WEXITSTATUS נקבל את הערך של קוד היציאה וידפיס.

4

ראשית הvalue יהיה שווה ל3 כעת נבצע fork.
בתוצאת if נקבל עבור האבא 0 והוא יחכה לקבלת קוד יציאה.
הבן לא יעבור את תנאי if ולכן יכנס לelse אשר יבצע יציאה עם הערך של value שהוא 3.
האבא יקבל את את הקוד יציאה לסטטוס ובאמצעות המקור WEXITSTATUS הוא יקבל את הערך 3 לאחר מכן נעלה ב1 את ערכו של value (כעת הוא 4) ונחזיר אותו.