

Operating Systems – 234123

Homework Exercise 1 – Dry

Winter 2021-22

Submitters:

Yan Tomsinsky, 207231267

Gabriel Kositsky, 318261948

Question 1 – Process management (60 points)

Part 1

```
int X = 1, p1 = 0, p2 = 0;

int ProcessA() {
    printf("process A\n");
    while(X);
    printf("process A finished\n");
    exit (1);
}

void killAll(){
    if(p2) kill(p2, 15);
    if(p1) kill(p1, 9);
}

int ProcessB() {
    X = 0;
    printf("process B\n");
    killAll();
    printf("process B finished\n");
    return 1;
}

int main(){
    int status;
    if((p1 = fork()) != 0)
        if((p2 = fork()) != 0){
            wait(&status);
            printf("status: %d\n", status);
            wait(&status);
            printf("status: %d\n", status);
        } else {
            ProcessB();
        } else {
            ProcessA();
        }
    printf("The end\n");
    return 3;
}
```

עבור כל אחת משורות הפלט הבאות, סמנו כמה פעמים הן מופיעות בפלט כלשהו, נמקו את תשובתכן.

1. process A

a. 0

b. 0 or 1

c. 1

d. 1 or 2

e. 2

נימוק: נבחין כי השורה Process A מודפסת רק בתוך הפונקציה ProcessA(). הקריאה היחידה לפונקציה זו נעשית ע"י הבן הראשון שהאבא יוצר בfork הראשון. נבחין בין 2 מקרים:

מודפס ProcessA: תהליך האב יוצר את p1 בfork. הבן p1 מתחיל לרוץ, קורא לפונקציה, מדפיס את השורה ProcessA ונתקע בלולאת while. לאחר מכן יוצר תהליך האב בן נוסף p2, אשר קורא לפונקציה ProcessB אשר שולחת SigKill לתהליך הבן הראשון (p1). לאחר מכן לא יהיו עוד הדפסות של השורה מכיוון שלא יהיו עוד קריאות לפונקציה ProcessA. לא מודפס Process A: גם במצב זה תהליך האב יוצר את p1 בfork אך הקריאה לפונקציה ProcessA או ההדפסה עצמה לאחר הקריאה מתעכבת, ובנתיים האב יוצר את תהליך הבן p2 אשר מספיק לשלוח את הסיגנל SigKill ל p1 לפני שההדפסה מתבצעת.

2. status: 1

a. 0

b. 0 or 1

c. 1

d. 1 or 2

e. 2

נימוק: הסטטוסים היחידים שיודפסו הם של תהליכי הבן שנוצרו על ידי האבא. התהליך הראשון שנוצר נכנס ללולאה אינסופית בתוך הפונקציה ProcessA (או שלא מגיע לנקודה זו כפי שהוסבר בסעיף הקודם), ולאחר מכן נעצר באמצעות Sigkill, לכן יחזיר את קוד הסיגנל (9) + 128. התהליך השני יריץ את הפונקציה ProcessB ולאחר החזרה מהפונקציה ימשיך להריץ את המשך הקוד, כלומר יחזיר את הערך 3 מתוך הmain ולכן גם במקרה זה לא יודפס סטטוס 1.

3. status: 137

a. 0

b. 0 or 1

c. 1

d. 1 or 2

e. 3

f. 2

נימוק: כפי שהוסבר בסעיף הקודם, כאשר הפונקציה ProcessB תשלח סיגנל Sigkill לתהליך הראשון, p1 סטטוס החזרה יהיה מספר הסיגנל – $128+9=137$ (כפי שצוין בהוראות התרגיל) וזהו המקום היחיד בו ייתכן סטטוס זה.

סטטוס זה בהכרח יודפס כי בכל מצב נקבל את הקריאה לפונקציה ProcessB אשר תקרא לKillAll והאב תמיד יחכה לתהליך הבן p1 אשר תקוע בלולאה אינסופית עד שיסתיים ויהיה במצב זומבי וזה יקרה לאחר שליחת ה SigKill לp1 שלאחריו בהכרח יודפס סטטוס החזרה 137.

4. status: 143

a. 0

b. 0 or 1

c. 1

d. 1 or 2

e. 2

נימוק: בקוד הנוכחי הדפסה זו תתכן רק בעת שליחת הסיגנל מספר 15 ל p2 בתוך הפונקציה KillAll. מכיוון שאנו ניגשים לפונקציה ProcessB רק מתהליך הבן השני (p2) אשר קוראת ל KillAll, נקבל שהתנאי לשליחת הסיגנל 15 ל p2 מעולם לא יתקיים שכן $p2=0$ במצב זה וזאת כי תהליך הבן p2 יקבל את ערך החזרה 0 מקריאת ה fork שיוצרה אותו.

5. The end

a. 0

b. 0 or 1

c. 1

d. 1 or 2

e. 2

נימוק: שורה זו מודפסת בסוף פונקציית הmain. השורה תודפס לאחר שהאב יבצע את הקוד שלו ויגיע לסוף הפונקציה. בנוסף, בתוך התהליך הבן השני p2 לאחר שיסיים לבצע את הפונקציה ProcessB יחזור לmain, ימשיך להריץ את הקוד ויסיים אותו בדיוק באותה אופן של האב. תהליך הבן הראשון שנוצר יסתיים בעת ביצוע הפונקציה ProcessA שכן הוא נתקע שם בלולאת while אינסופית עד שהורגים אותו ולכן הוא לעולם לא יגיע להדפסה זו.

Part 2

עבור אותו קוד כמו בחלק הקודם - סמנו עבור כל שורה האם היא יכולה להתקיים בפלט כלשהו, ונמקו.

1. השורה process B מופיעה לפני השורה process A. **אפשרי** / **לא אפשרי**
נימוק: תהליך הבן השני מתבצע במקביל לתהליך הבן הראשון, וייתכן מצב בו אנו קוראים קודם לפונקציה ProcessB מתהליך הבן השני (p2) והשורה Process B מודפסת, ולפני שהפונקציה KillAll מתוך הפונקציה ProcessB אנחנו חוזרים להריץ את הקוד של תהליך הבן הראשון שנוצר ובו מודפסת השורה Process A.

2. השורה status: 137 מופיעה לפני השורה The end. **אפשרי** / **לא אפשרי**
נימוק: ברגע בו תהליך הבן השני שולח Sigkill לתהליך הבן הראשון, הוא עדיין במרחק מספר שורות קוד מהדפסת the end. מכיוון שהתהליכים רצים במקביל ייתכן שעד שתהליך הבן השני יגיע להדפסה זו, תהליך הבן הראשון יסתיים, יחזיר את הקוד שלו לאב ואז האב ידפיס את הסטטוס שלו (137) כפי שהוסבר בחלק הקודם של השאלה.

3. השורה status: 143 מופיעה לפני השורה status: 137. **אפשרי** / **לא אפשרי**
נימוק: כפי שנימקנו בחלק הראשון של השאלה, סטטוס 143 לעולם לא יודפס ולכן מצב זה כמובן לא אפשרי.

Question 2 – I/O (40 Points)

נתונה התכנית הבאה. הניחו כי קריאות המערכת, פונקציות הספרייה מצליחות והקריאות והכתיבות מסתיימות במלואן. כמו כן הניחו כי הקובץ myfile לא קיים לפני תחילת ריצת התכנית.

```
int main()
{
    // creates a new file having full read/write permissions
    int fd = open("myfile", O_RDWR|O_CREAT, 0666);
    write(fd, "haha\n", 5);
    close(fd);                                // line 6
    fd = open("myfile", O_RDWR);              // line 7
    close(0);
    close(1);
    dup(fd);
    dup(fd);
    if (fork() == 0)
    {
        char s[100];
        dup(fd);
        scanf("%s", s);
        printf("hello\n");
        write(2, s, strlen(s));               // line 18
        return 0;                             // line 19
    }
    wait(NULL);
    printf("Father finished\n");
    close(fd);
    return 0;
}
```

תיאור כללי:

בתחילת הפונקציה אנחנו יוצרים קובץ חדש, אליו אנחנו כותבים haha. לאחר מכן, סוגרים את הקובץ ופותחים אותו מחדש (כלומר ה- seek pointer מצביע שוב לתחילת הקובץ) ואז משנים את ערוצי הקלט והפלט הסטנדרטיים להצביע על הקובץ עצמו.

לאחר קריאת ה fork, הבן שנוצר מעתיק את טבלת ה FDT מאביו. כעת, לפי ההנחה גם במקום השלישי ב FDT אנחנו מצביעים על הקובץ. לאחר מכן, אנחנו סורקים מתוך ערוץ הקלט הסטנדרטי (אשר שונה להיות הקובץ) ולכן לתוך S נסרקת המחרוזת שבקובץ (haha). בנוסף, במהלך הסריקה ה seek pointer מתקדם ולכן שוב יצביע לסוף הקובץ וב print בשורה אחרי אשר תכתוב לקובץ, לא תדרוס את haha. לאחר מכן, ישנה כתיבה לערוץ השגיאות הסטנדרטי (אשר לא שונה בשום שלב) והוא מדפיס למסך. בהמשך, חוזרים לאב אשר מחכה לבנו שיסתיים ואז מדפיס לערוץ הפלט הסטנדרטי (עדיין הקובץ) את father finished.

1. מה יודפס על המסך?

יודפס על המסך haha, מכיוון שסגרנו את ערוץ הפלט הסטנדרטי של האב, והכנסנו את הקובץ לשם במקומו, לכן כל הדפסה שתבצע אצל האב תוכנס לקובץ. בנוסף, בעת יצירת תהליך הבן הוא מעתיק את ה FDT של האב, לכן גם ערוץ הפלט הסטנדרטי שלו יצביע לקובץ, וגם כל הדפסה שתבצע ממנו תהיה לקובץ.

המקום היחיד בו תהיה הדפסה למסך הוא בעת קריאה לפונקציה write בתוך תהליך הבן לערוץ השגיאות הסטנדרטי. נזכיר כי ערוץ זה מדפיס גם הוא למסך. כפי שתיארנו, מכיוון שה seek pointer של הקובץ "אופס" כאשר פתחנו וסגרנו את הקובץ לאחר הדפסת haha לתוכו, s תסרוק בדיוק את haha ולבסוף יודפסו 2 הבתים הראשונים שנמצאים בתוך s.

2. מה יהיה תוכן הקובץ myfile בסיום התכנית?

תוכן הקובץ יהיה:

```
haha\nhello\nFather finished\n
```

בתחילה נכתב לקובץ haha. לאחר מכן, כפי שציינו מכיוון שסגרנו את הקובץ, ולאחר מכן פתחנו אותו שוב וסרקנו למחרוזת s מתוכו אנו נמשיך בכתיבה מאותה הנקודה. כלומר, תהליך הבן ידפיס לאחר haha את hello, ולאחר סיום תהליך הבן, האב גם יכתוב לקובץ Father finished.

3. האם יהיה שינוי בפלט ו/או בקובץ myfile ו/או בזמן ריצת התכנית אם נמחק את שורות 6,7? נמקו.

יהיה שינוי בפלט למסך אך לא בתוכן הקובץ. כעת, לא יודפס כלום למסך. לאחר הסרת השורות, מכיוון שלא נסגור ונפתח מחדש את הקובץ, ה seek pointer של הקובץ לא יתאפס וכאשר נסרוק לתוך s מתוך הקובץ, לא ייסרק כלום שכן ה seek pointer מצביע לסוף הקובץ. כלומר, s תהיה מחרוזת ריקה ולכן בכתיבה לערוץ השגיאות הסטנדרטי לא יודפס כלום למסך. הקובץ עצמו לא ישתנה שכן עקב אי סגירת הקובץ ה seek pointer כאמור לא "יתאפס" ולכן print בתהליך הבן תדפיס בדיוק באותו אופן לקובץ כפי שהדפיסה לאחר ש seek pointer התקדם בזכות הסריקה למחרוזת s.

זמן הריצה יתקצר שכן פתיחה וסגירה של קובץ דורשת גישה לדיסק אשר לוקחת זמן רב. בנוסף, הסריקה תיקח פחות זמן שכן אין מה לסרוק לתוך s מהקובץ.

4. האם יהיה שינוי בפלט ו/או בקובץ myfile ו/או בזמן ריצת התכנית (ביחס לסעיפים א' ו-ב') אם בשורה 18 במקום המספר 2 יופיע המספר 3? נמקו.

יהיה שינוי בפלט למסך ובקובץ. למסך לא יודפס כלום ותוכן הקובץ יהיה:

```
haha\nhello\nhaha\nFather finished\n
```

מכיוון שאנחנו כותבים לתוך fd מספר 3 אשר גם הוא מצביע לקובץ (שכן לא פתחנו קבצים נוספים), לא יודפס כלום על המסך (בשורה זו מתבצעת ההדפסה היחידה), אלא שתוכן המחרוזת s(haha\n) יודפס גם הוא לתוך הקובץ.

סביר להניח כי לא יהיה שינוי בזמן הריצה של התוכנית שכן אנחנו מבצעים אותה כמות של פקודות קריאה/כתיבה מלבד זה שאנו כותבים לקובץ במקום למסך.

5. האם יהיה שינוי בפלט ו/או בקובץ myfile ו/או בזמן ריצת התוכנית (ביחס לסעיפים א' ו-ב') אם נמחק את שורה 19?

לא יהיה שינוי בפלט למסך ויהיה שינוי בקובץ.

תוכן הקובץ יהיה:

```
haha\nhello\nFather finished\nFather finished\n
```

כעת, בסוף ה if של הבן לא יבצע return 0 (עד לנקודה זו הכל נשאר זהה) ולכן הוא ימשיך בריצת הקוד (בזמן שאביו מחכה לו שייגמר). בהגעת הבן לwait הוא מיד ימשיך הלאה שכן אין לו בנים לחכות להם. לאחר מכן הוא ידפיס לקובץ Father finished\n שכן לא בוצע שום שינוי ב fdt לעומת ההדפסות הקודמות ואז יסגור את הקובץ רק אצלו. לאחר מכן הבן יסיים את הריצה שלו והאבא יסיים לחכות לו וידפיס כמו מקודם Father finished\n לקובץ.

זמן הריצה יגדל, כי כעת גם הבן ידפיס לקובץ שורה נוספת, כלומר זוהי גישה לדיסק שלוקחת זמן רב יחסית.