**שאלה 1**

1. **מה יודפס על המסך עבור הקוד הבא:**

**int main() {**

**int res = fork();**

**if (res != 0) {**

**close(STDOUT);**

**}**

**int fd = open("myFile", O\_RDWR);**

**if (res != 0) {**

**printf("Hello from father\n");**

**}**

**else {**

**printf("Hello from son\n");**

**}**

**}**

* תהליך האב סוגר את שנמצא בכניסה ה-1 ב- ואז פותח את במקומו, לכן ייכתב אל תוך הקובץ . תהליך הבן פותח את בכניסה כלשהי ב- (שאינה במקום ה-1), לכן יודפס לטרמינל. לכן על המסך יודפס .

1. **כיצד תשתנה התשובה לסעיף הקודם אם נשנה את הקוד כך:**

**int main() {**

**close (STDOUT);**

**int res = fork();**

**int fd = open("myFile", O\_RDWR);**

**if (res != 0) {**

**printf("Hello from father\n");**

**}**

**else {**

**printf("Hello from son\n");**

**}**

**}**

* תחילה תהליך האב סוגר את שנמצא בכניסה ה-1 ב-. לאחר מכן פעולת ה- יוצרת תהליך בן אשר גם עבורו הכניסה ה-1 ב- פנויה. גם תהליך האב וגם תהליך הבן יפתחו את בכניסה ה-1 בטבלת ה- במקום . לכן שני התהליכים יכתבו אל הקובץ במקום אל המסך, ועל המסך לא יודפס דבר.

1. **נתון קטע הקוד הבא (הניחו ש- מאותחל במקום כלשהו)**

**#define BUF\_SIZE 100000**

**int my\_pipe[2];**

**pipe (my\_pipe);**

**char buf[BUF\_SIZE];**

**int status = fork();**

**// Filled buf with message…**

**if (status == 0) { /\* son process \*/**

**close(my\_pipe[0]);**

**write(my\_pipe[1],buf,BUF\_SIZE\*sizeof(char));**

**exit(0);**

**}**

**else { /\* father process \*/**

**close(my\_pipe[1]);**

**wait(&status); /\* wait until son process finishes \*/**

**read(my\_pipe[0], buf, BUF\_SIZE \* sizeof(char));**

**printf("Got from pipe: %s\n" buf);**

**exit(0);**

**}**

1. **מה הבעיה בקטע הקוד הנתון?**
   * + - ו- לא מתחייבות לקרוא/לכתוב את כל הנתונים שב- (בפועל מובטח כי לפחות תו אחד ייכתב או ייקרא) לכן לא מבוטח כי תהליך הבן יצליח לכתוב את כל מה שב- אל ה- וכן לא מובטח שתהליך האב יצליח לקרוא מה- את כל מה שתהליך הבן כתב אליו.
2. **הראו כיצד ניתן לפתור את הבעיה ע"י תיקון הקוד שמבצע תהליך הבן.**
   * + - על מנת לפתור את הבעיה נציע לתקן את הקוד שמבצע תהליך הבן כך

**close(my\_pipe[0]);**

**int n = BUF\_SIZE;**

**int res;**

**while (n > 0){**

**res = write(my\_pipe[1], buf, n\*sizeof(char);**

**n -= res;**

**buf = ((char\*)buf) + res;**

**}**

**exit(0);**

כלומר, נוסיף לולאה בתהליך הבן שתדאג שכל התווים ב- יקראו אל תוך ה- לפני סיום קטע הקוד של תהליך הבן.

1. **נתון הקוד הבא:**

**int x = 0;**

**int i = 3;**

**void catcher3(int signum){**

**i = 1;**

**}**

**void catcher2(int signum){**

**if(i != 0){**

**x = 5;**

**}**

**}**

**void catcher1(int signum){**

**printf("%d\n", i);**

**i--;**

**if (i == 0){**

**signal(SIGFPE, catcher2);**

**signal(SIGTERM, catcher3);**

**}**

**}**

**int main(){**

**signal(SIGFPE, catcher1);**

**x = 10/x;**

**printf("Goodbye");**

**}**

1. **מה יודפס למסך כאשר מריצים את הקוד כמו שהוא? נמקו.**
   * + - שגרת הטיפול ב- מוחלפת בשגרה .

עם ביצוע השורה "" ב- נעשית חלוקה ב-0 ולכן נשלח ה- ונקראת בשגרה . לכן על המסך מודפס וערכו מוקטן ב-1, לאחר מכן התוכנית תחזור על השורה שגרמה לשליחת ה-, כלומר השורה "" תתבצע בשנית. על המסך יודפס כך:

3

2

1

לאחר שערכו של מגיע להיות 0, מוחלפת שגרת הטיפול ב- לשגרה , כאשר זו אינה תבצע דבר משום ש- לעולם לא יהיה שונה מ-0 יותר. כלומר, התוכנית תכנס ללולאה אינסופית ו לא לעולם לא יודפס.

1. **מה יודפס למסך אם נתון שלאחר 10 שניות מבצעים את הפקודה בטרמינל (כאשר ה- הנתון הוא של התהליך שמריץ את הקוד הנ"ל).**
   * + - בהנחה כי 10 שניות הם זמן מספיק עבור ערכו של להשתנות ל-0, ביצוע הפקודה תשלח אשר יפעיל את שגרת הטיפול . ערכו של ישתנה ל-1 וכאשר תתבצע השורה "" שוב, שגרת הטיפול תשנה את ערכו של המשתנה ל-5. לכן, בנוסף להדפסות שמבצעת השגרה , כעת תבצע התוכנית את השורה מבלי לגרום ל- , ותודפס גם ההודעה . על המסך יודפס כך:

3

2

1

1. **מה יודפס למסך אם נוסיף פקודת , אחרי השורה "" שנמצאת בפונקציה ? (כאשר עדיין שולחים סיגנל בעזרת לאחר 10 שניות מתחילת ריצת התוכנית).**
   * + - הוספת הפקודה תגרום לתוכנית להמתין במשך 100 שניות במצב בו לא יכולים להתקבל מסוג אך יכולים להתקבל מסוגים אחרים. במהלך המתנה זו, ולפני החלפת השגרה לטיפול ב- לשגרה , מתקבל ה- . הטיפול ב- זה נעשה לפי ברירת המחדל ולכן הטיפול בתהליך יהיה בעזרת , כלומר סיום התהליך. התהליך יסתיים כאשר על המסך מודפס כך:

3

2

1

**שאלה 2**

1. **סערת פסיקות () הוא כינוי לתופעה כשאר קצב הגעת הפסיקות גבוה מאוד. מה היא הבעיה במצב זה וכיצד להתמודד איתה?**

* כאשר פסיקות מגיעות בקצב הגעה גבוה מאוד ייתכן מצב בו פסיקות מגיעות כאשר הפסיקות במעבד חסומות (דגל כבוי). כתוצאה מכך ביצוע פסיקות קריטיות עלול להתעכב ואף פסיקות עלולות להיאבד (וכך עלול להיאבד מידע). על מנת להתמודד עם בעיה זו, מחולק הטיפול בפסיקה לשלבי טיפול, כאשר הקטע הקריטי () בכל טיפול מצומצם למינימום ומבוצע מיד עם קבלתה. לאחר השלב הקריטי, מודלק דגל חזרה והמעבד מאפשר קבלה של פסיקות חדשות. שאר חלקי הטיפול בפסיקה מטופלים מיד לאחר סיום השלב הקריטי של הפסיקה החדשה (), עבור חלקים הדורשים התייחסות דחופה (אך לא מידית), וחלקי טיפול אשר אינם דורשים התייחסות מידית מועברים למאגר משימות ממתינות אשר טיפולן נעשה בעדיפות נמוכה הרבה יותר.

1. **נתון כרטיס רשת שעובד בקצב של ושולח ומקבל חבילות () בגודל קבוע של . נניח כי מערכת ההפעלה רוצה לעבוד עם הכרטיס בשיטת .**
2. **מדוע לא כדאי לקבוע את תדר ביצוע ה- ל- .**
   * + - מאחר קצב עבודת כרטיס הרשת הוא , הוא מסוגל לשלוח (או לקבל) חבילה בגודל אחת לכ-. לא כדאי לקבוע את תדר ביצוע ה- ל- מאחר ומערכת ההפעלה תדגום את מוכנות כרטיס הרשת אחת ל-, כך שרק אחת לכמאה דגימות יהיה המידע בכרטיס הרשת מוכן. מתוך כך נובע כי מערכת ההפעלה תבזבז הרבה משאבים לחינם בעקבות בתדירות כזו.
3. **מדוע לא כדאי לקבוע את תדר ביצוע ה- ל-.**
   * + - מאחר קצב עבודת כרטיס הרשת הוא , הוא מסוגל לשלוח (או לקבל) חבילה בגודל אחת לכ-. לא כדאי לקבוע את תדר ביצוע ה- ל- מאחר ומערכת ההפעלה תדגום את מוכנות כרטיס הרשת אחת ל-, כך שקצב עבודת כרטיס הרשת יהיה פי מאה מהיר יותר מקצב דגימת מערכת ההפעלה. מתוך כך נובע כי מערכת ההפעלה לא תאפשר שימוש יעיל בכרטיס הרשת בעקבות בתדירות כזו.
4. **בתרגול ראינו כי בפונקציה , לפני שניגשים לבצע פעולות על ה- משתמשים בפקודה אשר תופסת את ה- של ה- וחוסמת את הפסיקות במעבד הנוכחי.**
5. **מדוע יש צורך לחסום פסיקות ולא ניתן להסתפק בתפיסת מנעול?**

**?**

1. **באיזו בעיית סנכרון היינו נתקלים אם לא היינו חוסמים את הפסיקות? יש לתאר מצב בו מתרחשת בעיה זו.**

**?**

1. **נתונה מערכת בעלת שני מעבדים ו- אשר מריצים תהליכים ו- בהתאמה. ברגע תהליך שולח אל תהליך . הניחו כי מערכת ההפעלה אינה מבצעת החלפת הקשר ושהפסיקות והסיגנלים לא חסומים.**
2. **מה פרק הזמן המקסימלי בין הרגע בו נשלח הסיגנל לרגע בו יתחיל לטפל בו?**

פרק הזמן המקסימלי בין הרגע בו נשלח הסיגנל לרגע בו יתחיל לטפל בו הוא הזמן שיעבור עד אשר תתקבל פסיקת השעון הבאה במעבד והטיפול בפסיקה זו יסתיים. שבמהלך החזרה לרמת ה- מרמת ה- העובדה שתהליך קיבל סיגנל תתגלה והתהליך יתחיל בטיפול בסיגנל.

1. **תן דוגמא למצב בו יתחיל לטפל בסיגנל לאחר פרק זמן קצר יותר.**

**?**

1. **תארו את ההבדל בהתנהגות של הפקודה בעת חזרה מפסיקה מקוננת לעומת חזרה מפסיקה לא מקוננת.** 
   * + - בפסיקה מקוננת משחזרת את הערכים על המחסנית