סרגיי בוכמנס

שם

321188898

מספר זהות

serbuh@t2

דואר אלקטרוני

דני אנשלס

שם

311727200

מספר זהות

sdanshel @t2

דואר אלקטרוני

תרגיל בית יבש מספר:

3

מערכות הפעלה

234123

ציון

אלכסנדר בוגצ'נקו

שם

312867187

מספר זהות

shadow89@t2

דואר אלקטרוני

להחזיר לתא מס'

הוגש ע"י:

87

HW3

# שאלה 1

## סעיף 1

מספר חוטים יכולים לרוץ ב"מקביל" על מעבד אחד בעזרת החלפת הקשר ביניהם כלומרpthread\_create() יוצרת חוט חדש המתבצע ב"מקביל" לחוט הקורא (main).

התוצאות האפשריות הן:

מקסימלי 200 - יצירת 2 חוטים ב-, הרצת הפונקציה של חוט 1 ללא החלפת הקשר ובסיומו . הרצת חוט 2 ללא החלפת הקשר כך שבסיומו .

מינימלי 100 - יצירת 2 חוטים ב- הלולאה הראשונה. חוט 1 מקבל  ואז מתבצעת החלפת הקשר לחוט 2 שרץ ללא החלפת הקשר ובסיומו נגיע ל-. החלפת הקשר לחוט 1, חוט זה מחבר לערך  את 1. כעת  ובסיומו ריצת חוט זה נקבל .

## סעיף 2

int result;

int main() {

pthread\_t threads[2];

int i;

result = 0;

for(i = 0; i < 2; i++) {

pthread\_create(&threads[i], NULL, do\_calc, NULL);

**pthread\_join(threads[i], NULL);**

}

printf("%d\n", result);

return 0;

}

במקרה זה התוצאה היא 200 והינה תוצאה קבועה. זאת מהסיבה שלאחר הפעלת כל חוט תתבצע המתנה עד לסיומו כלומר ה- יגדל ב-1 בכל אחד מהחוטים 100 פעמים (סה"כ 200).

# שאלה 2

## סעיף 1

המשאבים המשותפים לחוטים הם: מרחב הזיכרון, גישה לקבצים והתקני חומרה, מנגנונים שונים של מערכת הפעלה.  
המשאבים בלעדיים הם: מחסנית ורגיסטרים.

## סעיף 2

הבדלים בין תהליך לחוט:

* תהליך, הוא בלתי תלוי בדר"כ. חוט לעומתו מתקיים כתת קבוצה של תהליך.
* תהליך נושא יותר מידע מאשר חוט בעוד שחוטים בתהליך משתפים משאבים (סעיף 1) .
* בחישוב על ידי תהליכים צריך איזשהו ערוץ תקשורת בין התהליכים ואילו בחישוב על ידי חוטים פשוט משתמשים באזור הזיכרון המשותף.
* לכל תהליך מרחב כתובות משלו לעומת חוטים שמשתפים את מרחב הכתובות שלהם.

## סעיף 3

היינו משתמשים בריבוי חוטים בתוכנית שלנו על מנת שנוכל לבצע חלקים באופן בלתי תלוי במשימות של אותו תהליך, לצורך שיפור ביצוע של תהליך על מעבד יחיד וגם מהסיבה שהוספת חוט לביצוע תכנית זולה מהוספת תהליך לאותה מטרה (אינה כרוכה בהקצאת משאבים נוספים).

שימוש בריבוי חוטים לא מתאים במימוש תכניות קצרות ופשוטות – יצירת חוט לוקחת זמן, החלפת הקשר בין החוטים לוקחת זמן במילים אחרות תקורה מיותרת מבלי לשפר ביצועים ולכן עדיף להריץ תהליך אחד ולסיימו.

אולם שימוש בריבוי חוטים עוזר בתוכניות שניתן לחלק ליחידות לוגיות כך שחלק מהזמן או רוב הזמן יש לנו יחידות שממתינות. למשל יחידות שמבצעת פלט כגון מסך, יחידות שמבצעת קלט כגון מקלדת ויחידות נוספת שמבצעת חישובים. במקרה זה היחידות של פלט וקלט נמצאות במצב המתנה ולכן בחלוקה לחוטים יהיה חוט שממתין לפלט, חוט שממתין לקלט וחוט שמבצע חישובים ורץ רוב הזמן במערכת. אחרת ללא חלוקה זו במקרה ויש לנו תהליך אחד שמבצע חישובים וכל פעם שהוא מקבל קלט הוא נכנס לתור המתנה וכתוצאה מכך לא מתקדם (לא מבצע את החישובים שלו).

## סעיף 4

**1-1**: יוצר או  לכל חוט של תכנית בנפרד. יתרון מודל זה הוא המימוש הקל ברמת הספרייה, אבל החיסרון הוא שכל חוט שנוצר נהיה יקר יותר עבור משאבי הגרעין. סוג זה הוא התלוי ביותר במודל  של הגרעין הבסיסי.

**M-1:** מיישם את כל החוטים ב- ואילו הגרעין רואה אותם כתהליך עם חוט בודד. יתרון סוג זה מתבטא בכך שהינו הנייד ביותר מהסיבה שאינו דורש תכונות מיוחדות מהגרעין הבסיסי. לעומת זאת חסרונו מתבטא בכך שהוא דורש שכל החסימות של קריאות מערכת  ידמו בספרייה כ- בגרעין. הדמיה זו מוסיפה תקורה משמעותית לקריאות המערכת.

**M-N:** סוג זה הוא כמו **M-1** בכך שאינו יוצר  לכל חוט של תכנית בנפרד. הספרייה יוצרת מספר רב ככל שצריך  עבור השירות של חוטי התכנית ולאחר מכן מתזמנת חוטים של תכנית עליהם.  
חיסרון סוג זה מתבטא בזאת שבמקרים מסוימים הוא בעל משקל כבד יותר בכך שהתזמון מתבצע גם בגרעין בין ה- עבור התהליך וגם בספריה עבור החוטים של התכנית. לעומת זאת יתרונותיו מתבטאים בכך שאינו צורך משאבי גרעין עבור חוטי התכנית אשר אינם רצים בפועל. בנוסף סוג זה מספק ביצועים טובים משמעותית כאשר החוטים בתכנית מסונכרנים אחד עם השני והתזמון ברמת הספרייה מתבצע הרבה יותר מהר ע"י החלפה בין החוטים מהסיבה שאינו צריך לעבור לגרעין.

## סעיף 5

1. הביצועים הטובים ביותר – הסנכרון מתבצע ברמת הספרייה. בהנחה שאין תכניות אחרות שרצות, לא יתבצעו החלפות הקשר וכל  ירוץ על מעבד שהוקצה לו והמעבדים הנותרים יהיו בהמתנה. במידה ויש תכניות נוספות יתבצעו פחות החלפות הקשר מהסיבה שיש מעבדים פנויים אחרים להרצת התכניות.
2. ביצועים בינוניים – כמו ב-1, בהנחה שאין תכניות נוספות שרצות, לא יתבצעו החלפות הקשר וכל  ירוץ על מעבד שהוקצה לו ולא יהיו מעבדים בהמתנה. במידה ויש עוד תכניות יתבצעו החלפות הקשר עם ה- במידת הצורך.
3. ביצועים לא טובים – יתבצעו החלפות הקשר בין  עצמם מהסיבה שקיימים יותר חוטים מהמעבדים.

# שאלה 3

## סעיף 1

1. אנחנו רואים בProcess status report 4 חוטים (חוץ מחוט של shell ):  
   16742 – חוט של main. בחוט הזה מתבצע הקוד של הmain.

16743 – manager thread שמיוצר ברגע שmain מנסה ליצור את החוטים עם הפונקציות f1 ו-f2. תפקידו לנהל את היצירה ושחרור של התרדים שmain יוצר. [Drepper, section 1]

16744 – החוט הראשון שmain יצר (עם f1). בו מתבצע קוד של הפונקציה f1.

16745 – החוט השני שmain יצר (עם f2). בו מתבצע קוד של הפונקציה f2.

* קיום של הthread manager מוסיף בעיות. למשל עם הורגים את התרד הזה, אז לא נהיה מסוגלים לערוג את התרדים שהוא יצר.
* מערכת של הסיגנלים לא תואמת באופן מלא לPOSIX סטנדרט.
* בעיה הקודמת גורמת גם לבעיות נוספות בפרימיטיבים של הסינכרוניזציה שמסתמכים על המערכת של הסיגנלים.
* בעיות כתוצאה מPID שונה לכל תרד שנוצר.
* הגבלה על מספר החוטים האפשריים (8191) גורמת בעיות.
* בגרעין יש בעיה בשימוש של /proc במקרה של מספר רב של החוטים לתהליך יחיד.
* בעיות במימוש של הסיגנלים גורמת לבעיות באי תמיכה בגרעין.

1. חסר!!!