

תרגיל בית 2

מבנה מערכות הפעלה

046209

מגישים:

דור משעלי 311510630

כריסטיאן שקור 208157826

שאלה 1.

סעיף 1 - הגדירו את המושגים הבאים :

- א. **Context Switch** — החלפת הקשר של התהליך שרץ כרגע על המעבד. מערכת ההפעלה שומרת את הקשר של התהליך ומחליפה אותו בהקשר של תהליך חדש.
- ב. **Preemption** — הפקעה - מערכת ההפעלה לוקחת משאב מתהליך ומעבירה את התהליך למצב ready ובזמן הזה תהליך אחר ירוץ בעזרת המשאב.
- ג. **Interrupt** — פסיקה - פסיקה היא אות המגיע למערכת הפעלה (יש או פסיקות או חריגות). כאשר נשלח פסיקה למערכת ההפעלה, מופעל הטיפול בפסיקות ובאותו זמן התהליך שרץ נפסק. לאחר הטיפול, התהליך חוזר לרוץ.
- ד. **Context של תהליך** — ההקשר של התהליך מכיל מידע על התהליך: PC, mode, ps, PID. ההקשר הינו אבסטרקציה של מערכת ההפעלה.
- ה. **Scheduling** — מנגנון זימון, מנגנון הזימון הוא אלגוריתם שמחליט את סדר הרצת התהליכים בCPU.

סעיף 2 - אילו מהמדדים הבאים ניתן לשפר בעזרת שימוש ב-preemption במערכת עם משאב יחיד? נמק.

- א. זמן ביצוע כולל ממוצע (average turnaround time) או זמן שעון קיר ממוצע — כפי שראינו בהרצאות preemption משפר את זמן הביצוע הכולל הממוצע בכך שהוא משפר את זמן ההמתנה הממוצע וזאת משום שבעזרת preemption ניתן לגרום לתהליכים קצרים לרוץ לפני התהליכים הארוכים מה שגורם לזמן ההמתנה לקטון.
- ב. תפוקה (throughput) — בזכות preemption ניתן לשפר תפוקה שכן תפוקה מוגדרת כמספר התהליכים שהמעבד סיים לבצע ליחידת זמן ולכן אם יש preemption בעזרת אלגוריתם ניתן לבצע את התהליכים הקצרים בתחילת תור הביצוע וכך לגרום לתפוקה לעלות.
- ג. ניצולת מעבד (CPU utilization) — במחשב עם מעבד יחיד, הניצול של המעבד הוא 100 % וזאת משום שהCPU עובד כל עוד התוכניות לא סיימו לרוץ, לכן preemption לא משנה את הניצולת.
- ד. ממוצע slow down — כפי שציינו בא' אזי בעזרת preemption זמן ההמתנה הממוצע קטן ולכן ממוצע הslow down יירד.

אילו מהתשובות ישתנו אם נדון במערכת בעלת שני משאבים (למשל, מעבד ודיסק)?

ניתן יהיה לשפר את תכונות א,ב, ד שכן ב-א' אם יהיו שני מעבדים אז זמן הביצוע הכולל יקטן, עבור ב' לפי אותה סיבה אם יהיו שני מעבדים אזי כמות התהליכים יסתיימו מהר יותר לכן התפוקה תגדל, ועבור ד' אם יהיו שני מעבדים אז זמן ההמתנה יקטן ולכן הslow down יקטן גם.

פירוט:

עבור $n=3$ זה נוצר 31 תהליכים

היא: $2^5 - 1 = 31$.

הסבר: בניסוי `forkthem(5)`, התנאי `if(n > 0)` אכן מתקיים וכן

למעשה `fork()` יוצר תהליך `fork()`, ואז התהליך והבן שלו יוצרים

לפניהם הבאה ונכנסים לניקוז `forkthem(4)` שיהיה זמן בלבד זה יע 2 תהליכים

בהמשך נוצרים 32 תהליכים, כאשר בכל שלב מספר התהליכים גדל פי 2, $2^5 = 32$ במחץ גרף הריבוי נקרא 31. והן:

$n=1 \Rightarrow 32$ תהליכים $\Rightarrow n=2 \Rightarrow 16$ תהליכים $\Rightarrow n=3 \Rightarrow 8$ תהליכים $\Rightarrow n=4 \Rightarrow 4$ תהליכים $\Rightarrow n=5 \Rightarrow 2$ תהליכים

שאלה 3.

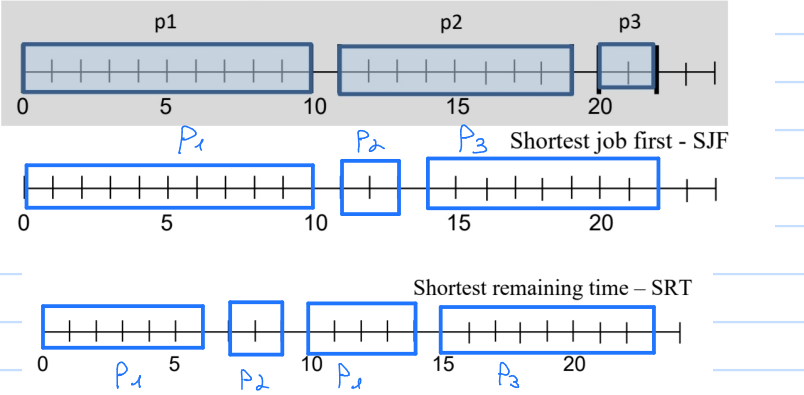
נתונים זמני ההגעה וזמני הריצה של התהליכים במערכת שבה החלפת תהליכים לוקחת יחידת זמן אחת.

תהליך	זמן הגעה	זמן ריצה
p1	0	10
p2	4	8
p3	6	2

א. תארו את התזמון של התהליכים בכל אחד מאלגוריתמי התזמון להלן.

א. תארו את התזמון של התהליכים בכל אחד מאלגוריתמי התזמון להלן.

לתיאור התזמון, סמנו את זמני ריצת התהליכים על ציר הזמן כמו בדוגמה הבאה:



ב. באיזה מהאלגוריתמים היית משתמש במערכת שמשלבת תהליכים עתירי חישוב ותהליכים אינטראקטיביים? הסבירו.

פתרון:

במערכת המשלבת תהליכים עתירי חישוב ותהליכים אינטראקטיביים היינו משתמשים ב-SRT שכן במצב online לא ניתן להחליט מראש איזה תהליכים יגיעו ויביאו את התוצאות. ההחלטות ביותר אלצם האופטימלי. דוגמה, צלן התיאור מה מתקצר מה שהולך לתהליכים אינטראקטיביים.

ג. באיזה מהאלגוריתמים היית משתמש במערכת שבה תהליכים עתירי חישוב בלבד? הסבירו.

פתרון:

באילויות SJF, כפי שהוכחנו בהוצאות, במצב offline (האקדמי) צלן גרמנג הוא SJF כאשר א/ תהליכים אינטראקטיביים.

שאלה 4.

עבור כל אחת מהפונקציות הבאות, רשמו אם לדעתכם היא משתמשת במימוש שלה בקריאת מערכת, והסבירו בקצרה את החלטתכם :

`printf()`, `sqrt()`, `rand()`, `gettimeofday()`, `sleep()`

(השתמשו ב `man pages` ע"מ ללמוד מה מבצעת כל פונקציה.)

פתרון:

- `sleep ()` — פונקציה במערכת ההפעלה `unix` שמשהה תהליך עבור פרק זמן מוגדר. הפונקציה משתמשת במימוש שלה בקריאת מערכת זו כדי לבקש ממערכת ההפעלה לא להריץ את התהליך.
- `gettimeofday ()` — פונקציה במערכת שמציגה את הזמן הנוכחי בשניות או מיקרו שניות. הפונקציה משתמשת במימוש שלה בקריאת מערכת כי אנחנו מעוניינים בזמן המערכת.
- `rand ()` — הפונקציה מגרילה מספר, מבצעת את ההגרלה על ידי חישוב באמצעות נוסחה ולכן לא משתמשת במערכת. הפונקציה לא משתמשת במימוש שלה בקריאת מערכת.
- `sqrt ()` — הפונקציה מחשבת שורש ריבועי של מספר, לא משתמשת במערכת כי היא עושה חישוב מספרי. הפונקציה לא משתמשת במימוש שלה בקריאת מערכת.
- `printf ()` — פונקציה שבעזרתה ניתן להדפיס למסך. הפונקציה משתמשת במסך ולכן מכיוון שזהו משאב של המערכת, במימוש שלה תהיה קריאת מערכת.

שאלה 5.

א. הסבירו את ההבדלים בין פסיקה סינכרונית לאסינכרונית. תנו דוגמה לכל אחד מהסוגים.

פס'ק ס'כונ'ט - פס'ק סכונ'ט ה'א פס'ק שנוצרת או למיננה או מחומה אב באופן ידוע ו'צפ. אדאמא, `syscall` ה'א פס'ק שנוצרת ז' י'צ' קוז שחצה אדמת אדמיז דכחאן באופן י'צפ.

פס'ק אסיכונ'ט - פס'ק שנוצרת מחומה באופן שא תאי קוז ש'ף כצת במזבז. אדאמא, פס'ק שזון.

ב. האם פסיקת שעון היא פסיקה סינכרונית או אסינכרונית? נמק

פתרון:
פס'ק שזון ה'נה פס'ק א סכונ'ט. כפ שאמת בס'ף קוזצ פס'ק זו אינה נדומת גלמן י'צפ ו'צח לראש ואכן אינה פס'ק סכונ'ט בס'ף ה'א פס'ק שנוצרת לרוב מחומה באופן שאינ תאי גת'ק ש'ף כצת במזבז.

ג. להלן קוד פשוט אשר מחלק באפס ותוצאת הריצה שלו. הסבירו בקצרה מהי שרשרת הפעולות שמובילה ליציאה מהתוכנית.

```
#include<stdio.h>
```

```
int main(void)
{
    int a = 0;
    int b = 1;

    b = b/a;

    return b;
}
```

Floating point exception (core dumped)

פתרון:
תחילה המזבז יר'ף את השורה "א'ש=ב" להיקוז הכוח'. המזבז גבין יש פס'ק וזכן ממפס בג'חא IDT את השורה שמג'חא בפס'ק למס' זה. אחו גכן נעשה ש' כ'י אהא'ף אמצב kernel וא'י השורה IDT מתחיל אה'ף קוז במזבז אדאמא. בתח'ק ג'סו בפס'ק נ'חא ס'דג א'תח'ק. אחו ג'סו בפס'ק וא'ני שתח'ק חוזר אה'ף נבדק וקאר הס'ט'ץ א' גתח'ק אבזק א'ש יש ס'ט' שחא צריב אס' בו.
מב'ון שבמאן ג'סו פס'ק ש'ט' א'ט' א'ט' ס'ט' בתח'ק שמחה א'ט'ו, כונ' ג'ט'ו בס'ט' רצה וה'א ג'דמת י'צ'אג א' גתח'ק אבזק ג'ט'אג.