

תרגיל בית #2

מטרות התרגיל:

1. להעמיק את הבנתכם בנושא עקרונות בסיסיים של מערכות הפעלה.
2. להעמיק את הבנתכם בנושא זימון תהליכים ב-Linux.



תרגיל יבש:

שאלה 1.

סעיף 1 - הגדירו את המושגים הבאים :

- א. **switch**
- ב. **Preemption**
- ג. **Interrupt**
- ד. **Context של תהליך**
- ה. **Scheduling**

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

ב. תפוקה (throughput)

ג. ניצולת מעבד (CPU utilization)

ד. ממוצע slow down

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

שאלה 2.

נתונה התוכנית הבאה שרצה ב UNIX ללא תקלות:

```
main(int argc, char ** argv)
{
    forkthem(5)
}

void forkthem(int n)
{
    if(n > 0)
    {
        fork();
        forkthem(n-1);
    }
}
```

כמה תהליכים נוצרים כאשר מורץ קטע הקוד להלן? הסבירו.

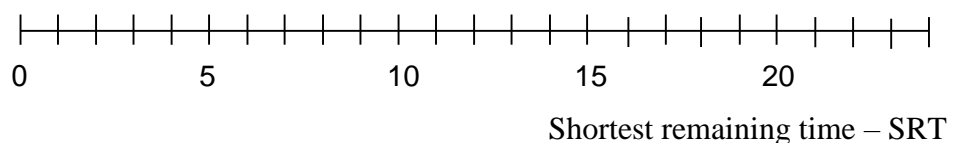
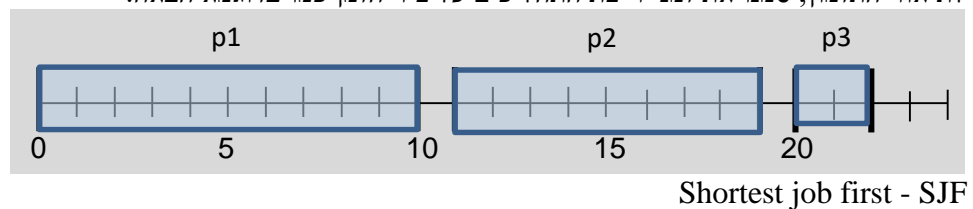
שאלה 3.

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.

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



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

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

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.)

שאלה 5.

- א. הסבירו את ההבדלים בין פסיקה סינכרונית לאסינכרונית. תנו דוגמה לכל אחד מהסוגים.
- ב. האם פסיקת שעון היא פסיקה סינכרונית או אסינכרונית? נמק
- 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)

שאלה 6.

נתונה התוכנית הבאה :

```
void main() { while(TRUE) if(fork()) exit(0); }
```

הסבירו בקצרה מדוע קשה להרוג תוכנית זו.

בהצלחה!!!