

מערכות הפעלה - מועד א' סמסטר אביב 2023

המבחן של רחל קולודני. בסמסטר זה נלמדו הפרקים 1-10, ו-13 ו-14.
התשובות מופיעות בסוף הקובץ.

חלק א' – שאלות שצריך לתת להן הסבר קצר

כל שאלה 3 נקודות, 10 שאלות.

1 – איך נקרא תהליך בן שאבא שלו לא עשה לו `wait()`?

2 – האם שני חוטים של תהליכים נפרדים יכולים לשתף דאטה?

3 – מה יותר גדול, הזיכרון הפיזי או הזיכרון הווירטואלי? נמקו.

4 – כמה פעמים תודפס המילה "Hello" בקטע הקוד הבא:

```
int a = 0;

int main()
{
    fork();
    a++;
    fork();

    if (fork() == 0){
        printf("Hello");
    }
    else{
        printf("Goodbye");
    }

    return 0;
}
```

5 – כפי שלמדת בתרגול על מטמון הדפים (page cache), האם יתכן שקיימת מסגרת שאין אליה מצביע מאף מרחב זיכרון?

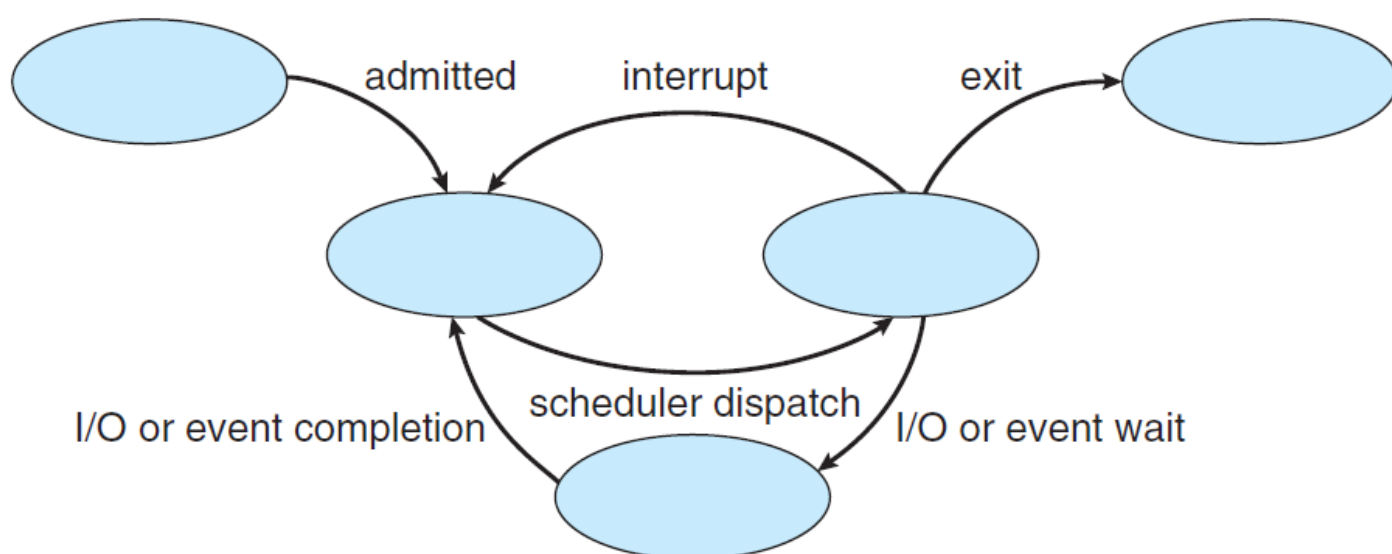
6 – האם הפקודה `exec()` בלינוקס מייצרת תהליך בן?

7 – מהו חוק המדאל?

8 – כאשר מגדילים את ה-cache, האם יתכן שהגדלה של ה-cache תביא לירידה ב-hit rate?

9 – במערכת עם multithreading, 1,000 חוטים קוראים ממערך נתונים משותף. האם צריך מנגנון סנכרון כדי למנוע race condition, נמקו באיזה מנגנון.

10 – מלאו את המצבים הבאים של תהליך במקומות המתאימים:
new, ready, waiting, running, terminated



חלק ב' – נכון/לא נכון, או תשובה סופית

כל שאלה היא 6 או 7 נקודות, 11 שאלות.

- 1 –** כפי שלמדתם בתרגול, מאיפה מוקצה הזיכרון כאשר משתמשים ב-malloc()?
(א) מוקצה בערימה בזיכרון המשתמש.
(ב) מוקצה באזור חדש הזיכרון.
(ג) מוקצה בערימה בזיכרון הקרנל.
(ד) 1 ו-2 נכונים.
(ה) 1, 2 ו-3 נכונים.

- 2 –** מה שמור בתוך ה-TLB?
(א) ה-TLB שומר בתוכו את התוכן הפיזי של דפים שתורגמו לאחרונה, ומי ששם את המידע בתוך ה-TLB הוא ה-CPU.
(ב) ה-TLB שומר בתוכו את התרגום של כתובות לוגיות לכתובות פיזיות, ומי ששם את המידע בתוך ה-TLB הוא ה-CPU.
(ג) ה-TLB שומר בתוכו את התוכן הפיזי של דפים שתורגמו לאחרונה .. ?
(ד) ??
(ה) ה-TLB שומר בתוכו את המידע של ה-scheduler, ומי ששם את המידע בתוך ה-TLB הוא ה-CPU.

- 3 –** נתון מרחב כתובות ווירטואליות עם 22 ביט. משתמשים ב-paging עם page table ברמת אחת. נתון מרחב כתובות פיזי עם 18 ביט. גודל כל דף הוא 4KB. כמו כן, כל כניסה בטבלת הדפים מכילה 2 ביטים לבקרה.

- (א) כמה מסגרות יש בזיכרון הפיזי?
(ב) כמה כניסות יש בטבלת הדפים?
(ג) מהו גודל טבלת הדפים?

- 4 –** נתון מרחב כתובות ווירטואליות עם 32 ביט. משתמשים ב-paging עם multilevel page table עם שתי רמות. נתון ש-16 הביטים הימניים הם ה-offset. 6 ביטים אחריהם מתארים את הרמה העליונה. 10 הביטים האחרונים מתארים את הרמה התחתונה.

- (א) כמה כניסות יש ברמה העליונה?
(ב) כמה כניסות יש ברמה התחתונה?
(ג) כמה דפים סה"כ יש בזיכרון הווירטואלי?

5 – לפניכם 4 תהליכים כאשר מפורטים זמני ההגעה שלהם למעבד ואורך ה-CPU burst של כל אחד מהם. מלאו את הטבלה לפי הזמן בו התהליכים היו במעבד עבור שלושת אלגוריתמי התזמון FIFO, SRTF, ו-RR. שימו לב שעבור RR, תהליך חדש שנכנס ל-ready queue נכנס לראש התור.

תהליך	זמן הגעתם למעבד	אורך ה-CPU burst
1	0	5
2	1	3
3	3	2
4	9	2

זמן	RR	FIFO	SRTF
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			

הנתונים הבאים הם עבור שאלות 6 ו-7.
לפניכם מימוש של compare and swap, שימו לב שמימוש זה אינו זהה לזה שראיתם בביתה.

```
typedef struct __lock_t {int flag;} lock_t;

int compare_and_swap(int* value, int expected, int new_value){
    int temp = *value;
    if(*value == expected)
        *value = new_value;
    return temp;
}

void init(lock_t* lock){
    lock->flag = abc;
}

void acquire(lock_t* lock){
    while (compare_and_swap(&lock->flag, 1, xyz) != xyz);
}

void release(lock_t* lock){
    lock->flag = 1;
}
```

6 – באיזה ערך יאותחל abc?

- א) 0.
- ב) 1.
- ג) 2.
- ד) 0 או 1.
- ה) 0 או 2.
- ו) 0 או 1 או 2.

7 – איזה ערכים xyz יכול לקבל?

- א) 0.
- ב) 1.
- ג) 2.
- ד) 0 או 1.
- ה) 0 או 2.
- ו) 0 או 1 או 2.

*שימו לב בשאלה זו יש טעות – בתוך לולאת ה-while, במקום != צריך להיות ==.

```

#define N 100
int count = 0;

int producer()
{
    while (True)
    {
        if (count==N)
            sleep();
        produce_item();
        count++;
        if (count==1)
            wakeup();
    }
}

int consumer()
{
    while (True)
    {
        if (count==0)
            sleep();
        consume_item();
        count--;
        if (count==N-1)
            wakeup();
    }
}

```

8 – בקטע הקוד הבא מתואר בעיית היצרן-צרכן.

אילו טענות מהבאות נכונה?

(א) הפתרון נכון, אך אינו יעיל מאחר וסובל מ-busy waiting.

(ב) ??

(ג) הפתרון לא נכון מאחר ויש race conditions.

(ד) ??

(ה) הפתרון לא נכון מאחר ולא מתקיים progress.

(ו) הפתרון לא נכון מאחר ולא מתקיים bounded waiting.

9 – במערכת בה יש priority scheduling ומשתמשים גם ב-priority inheritance, יש שני מנעולים ו-3

תהליכים עם העדיפויות הבאות:

תהליך	עדיפות
A	3
B	6
C	10

קורים 5 האירועים הבאים:

1 – תהליך A עושה acquire() ל-lock1 ול-lock2.

2 – תהליך B עושה acquire() ל-lock2.

3 – תהליך C עושה acquire() ל-lock1.

4 – תהליך A עושה release() ל-lock2.

5 – תהליך A עושה release() ל-lock1.

מה היא העדיפות בפועל של התהליך A בכל השלבים (משמאל לימין):

(א) 3,6,10,6,3

(ב) 3,6,10,10,3

(ג) 3,6,6,10,3

(ד) 3,6,10,3,3

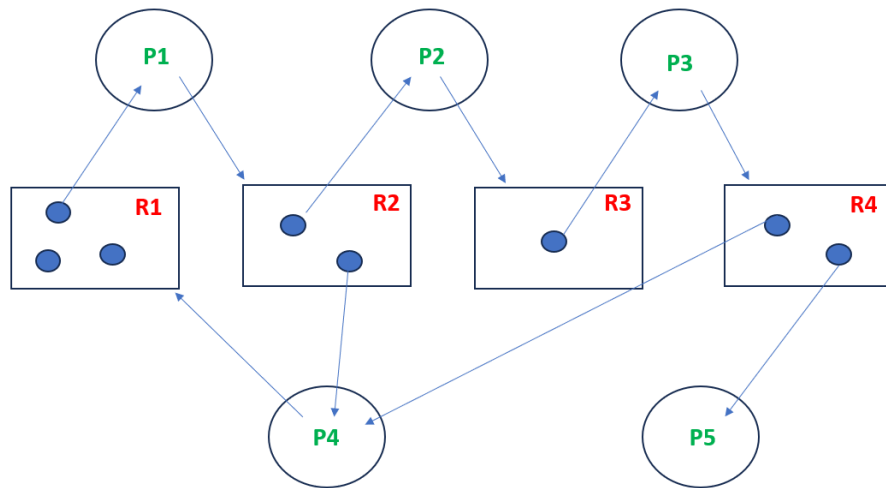
(ה) 3,10,10,6,3

10 – מובא לפניכם גרף הקצאת משאבים – Resource Allocation Graph.

האם יש deadlock?

אם כן, כתבו את סדר התהליכים שנמצאים ב-deadlock.

אם לא, כתבו את סדר סיום התהליכים.



11 – מה לא נכון לגבי B שעשה hard link ל-A?

א) B זה שם נרדף ל-A.

ב) כתיבה ל-A תכתוב גם ב-B.

ג) A יכול להיות תיקייה.

ד) A יכול להיות במערכת קבצים אחרת.

ה) מחיקה של B תוביל למחיקה של A.

תשובות

חלק א' – שימו לב **שצריך נימוק**.

1 – כאשר תהליך מסיים ותהליך האב לא עשה לו `wait()`, אז התהליך נקרא זומבי (zombie). אם תהליך האב מסיים מבלי לעשות `wait()`, הבן נקרא יתום (orphan).

2 – כן.

3 – אף אחד מהם – תלוי.

4 – המילה Hello תודפס 4 פעמים.

5 – יתכן.

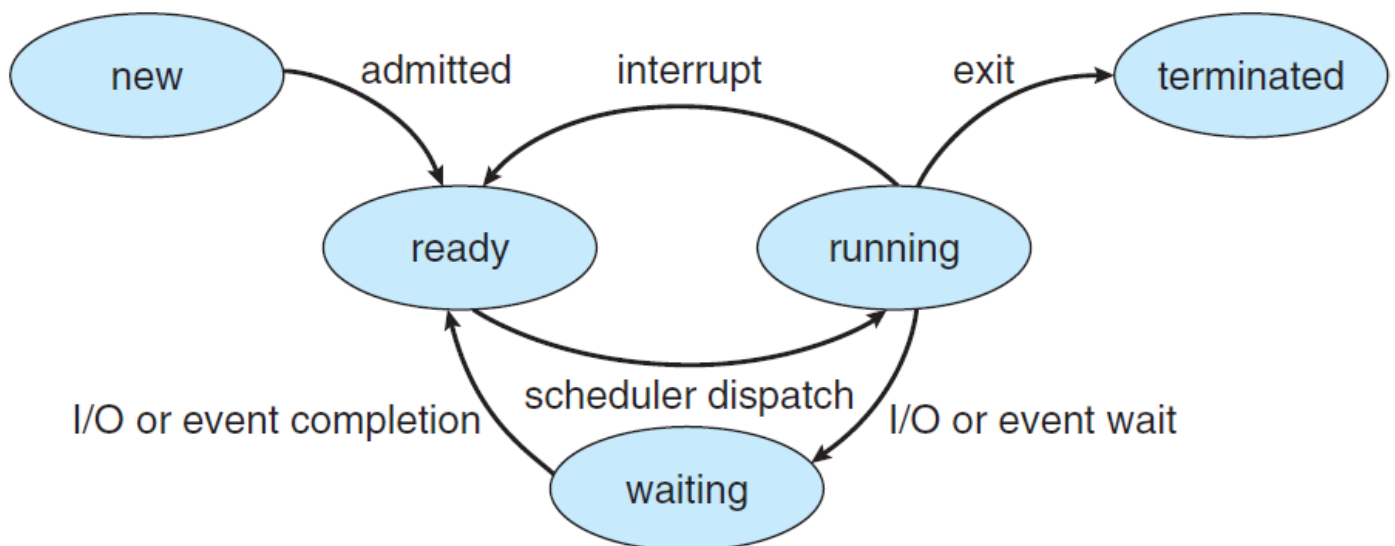
6 – לא.

7 – תקראו בספר.

8 – כן – קלאסי האנומליה של בילאדי.

9 – לא – מאחר והם רק קוראים.

10 –



חלק ב'

1 – ד'.

2 – ב'.

– 3

א) 2^6 .

ב) 2^{10} .

ג) $2^{10} = 1\text{KB}$.

– 4

א) 2^6 .

ב) 2^{10} .

ג) 2^{16} .

5 – פשוט מאוד..

6 – ב'.

7 – ה'.

8 – ג'.

9 – ב'

10 – אין deadlock. יש מספר אפשרויות לסדר תהליכים.

11 – ג', ד'.