מבוא למערכות הפעלה

שחזור

סמסטר ב' - 2022 / 2023

מועד ב'

מרצה: פרפ' רחל קולודני

** **<u>שימו לב</u>** שזה שחזור למבחן ולא המבחן עצמו (שלא פורסם על ידי צוות הקורס) השאלות שונות קצת מהשאולות שהיו במבחן, כמו ניסוח, ערכי נתונים ושמות משתנים.

זמן למבחן: שעתיים וחצי
מותר מחשבון

<u>חלק א'</u> – 10 שאלות נכון / לא נכון / נמקו (3 נקודות לכל שאלה)

1- תנו 2 חסרונות לדפים גדולים במערכת שמשתמשת ב paging
2- תנו 2 יתרונות לדפים גדולים במערכת שמשתמשת ב paging
3- במערכת מסוימת רוצים להימנע מ deadlocks, אם הבקשה למשאבים של תהליך יוצרת מעגל בגרף הקצאת משאבים, לא מאפשרים לתהליך לבקש אותם משאבים, האם זה יעבוד ? נמקו
4- המטרה העיקרית מ multi-level page table היא להאיץ תרגום כתובות וירטואליות לכתובות פיזית, נכון / לא נכון ? נמקו
5- המטרה העיקרית מ TLB היא להאיץ תרגום כתובות וירטואליות לכתובות פיזית נכון / לא נכון ? נמקו
6- במימוש shell מבצעים fork ואז exec, מותר לתהליך הילד או ההורה לבצע exec, האם זה משנה ?
7- מטמון הדפים הוא מקום רציף בזיכרון בגודל 4 KB, נכון / לא נכון
8- בשפת C, מערך נשמר במקום רציף בזיכרון הפיזי, נכון / לא נכון
9- מה ההבדל בין הנתיב האבסולוטי absolute path לנתיב יחסי relative path ?
10- האם אלגוריתם התזמון STRF יכול לגרום להרעבה ? נמקו

<u>חלק ב'</u> – 11 שאלות (לכל שאלה 6 או 7 נקודות)

1) **אילו מהטענות הבאות** נכונות לגבי invert page table:

א-

ב- לא אפקטיבי להשתמש ב invert page table במערכת בעלת זיכרון פיזי גדול

ג-

ד- תמיד לוקח פחות מקום מטבלת דפים רגילה

ה-

2) התוכנית הבאה רצה

```
int value = 0;
while (true) {
    printf ("%d", value);
    value++;
}
```

באותו זמן רצה התוכנית reset:

```
int value;
for (int i = 0; i < 5; i++){
   value = 0;
}
```

איזה רישות אפשריות לפלט?

א- 01234012134

ב-

ג- 012345678

ד-

ה-

3) במערכת 3 קבצים File1, File2, File3

ו 3 משתמשים user1, user2, user3

group1 מכיל user1, user2

group2 מכיל user2, user3

group3 מכיל user1, user3

File1:	rwx r-- r--	user2, group3
File1:	rw- rw- ---	user1, group2
File1:	rwx --- ---	user3, group1

א- איזה קבצים user1 יכול לקרוא ?

ב- איזה קבצים user2 יכול להריץ ?

4) התאים את הביטוי למשפט המתאים:

א) running	1) התהליך נוצר
ב) ready	2) התהליך מוכן לריצה / מחכה שיוקצה למעבד
ג) waiting	3) התהליך הסתיים
ד) terminated	4) התהליך מחכה לאירוע מסוים כמו O/I או סיגנל
ה) new	5) התהליך רץ

5) במערכת מסוימת משתמשים בטבלת דפים בעלת שתי רמות (2-level page table),
גודל דף 8 MB
הטבלה החיצונית ממלא בדיוק דף אחד
כל שורה בטבלה היא 4 בתים

א- השלם מספר bits בכתובת הווירטואלית (32 bits)

outer bits	inner bits	offset bits

ב- לכל טבלה פנימית משתמשים בדף אחד, כמה זיכרון מתבזבז (באחוזים) ?

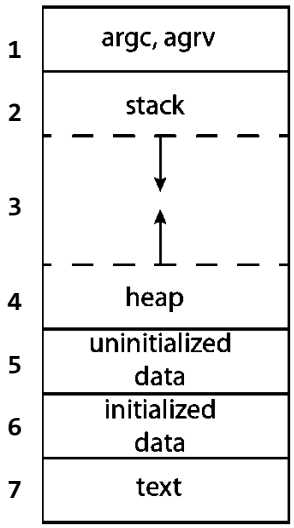
6) נתונה טבלת דפים בעשרוני (decimal), גודל דף 500 בית

Page number	Frame number	Valid bit	Dirty bit
0	3	1	0
3	14	1	1
1	19	0	0
2	7	1	1
200	11	1	0

א- האם נעשה שינוי בכתובת הווירטואלית 1200 לאחרונה?
ב- מה הכתובת הפיזית של הכתובת הווירטואלית 499 ?
ג- מה הכתובת הווירטואלית של הכתובת הפיזית 7321 ?

7) התאם את הפרמטרים לאיפה הם נשמרים

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

int x;                                  א
int y = 15;                             ב

int main(int argc, char *argv[])
{
  int *values;                          ג
  int i;                                ד

  values = (int*) malloc(sizeof(int)*5);  ה

  for(i = 0; i < 5; i++)
       values[i] = i;

  return 0;
}
```



8) נתונים 3 משתנים: lock, full, empty ושני מימושים לבעיית היצרן-צרכן:

implementation 1:

```
void produce(){
        empty.wait();
        lock.wait();
                /*produce item*/
        lock.signal();
        full.signal();
}
```

```
void comsume(){
        full.wait();
        lock.wait();
                /*consume item*/
        lock.signal();
        empty.signal();
}
```

implementation 2:

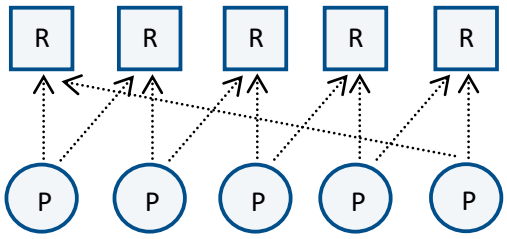
```
void produce(){
        lock.wait();
        empty.wait();
                /*produce item*/
        full.signal();
        lock.signal();
}
```

```
void comsume(){
        lock.wait();
        full.wait();
                /*consume item*/
        empty.signal();
        lock.signal();
}
```

מה נכון לגבי המימושים?

א- מימוש 1 נכון ומימוש 2 לא נכון
ב- מימוש 1 לא נכון ומימוש 2 נכון
ג- שני המימושים נכונים
ד- מימוש 2 תלוי במצב המערך (מלא, מלא חלקית, ריק)
ה- מימוש 1 תלוי במצב המערך (מלא, מלא חלקית, ריק)

9) מה מייצג הגרף ?



א-
ב- גרף הקצאת משאבים למניעת deadlock לבעיית הפילוסופים
ג-
ד-
ה-

10) מריצים קטע הקוד הבא בשתי מערכות, אחת משתמשת במתזמן FCFS, והשנייה ב RR עם t קצר:

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

```
pthread_mutix_t lock1, lock2, lock3;

int main ()
{
  pthread_t thread_1, thread_2, thread_3;

  pthread_create (&thread_1, NULL, func1, NULL);
  pthread_create (&thread_2, NULL, func2, NULL);
  pthread_create (&thread_3, NULL, func3, NULL);

  pthread_join(thread_1, NULL);
  pthread_join(thread_2, NULL);
  pthread_join(thread_3, NULL);

  return 0;
}

void *func1 ()
{
  pthread_mutix_lock (&lock1);
  pthread_mutix_lock (&lock2);
  pthread_mutix_lock (&lock3);
  /* do something */
  pthread_mutix_unlock (&lock3);
  pthread_mutix_unlock (&lock2);
  pthread_mutix_unlock (&lock1);
}

void *func2 ()
{
  pthread_mutix_lock (&lock2);
  pthread_mutix_lock (&lock3);
  pthread_mutix_lock (&lock1);
  /* do something */
  pthread_mutix_unlock (&lock1);
  pthread_mutix_unlock (&lock3);
  pthread_mutix_unlock (&lock2);
}

void *func3 ()
{
  pthread_mutix_lock (&lock3);
  pthread_mutix_lock (&lock1);
  pthread_mutix_lock (&lock2);
  /* do something */
  pthread_mutix_unlock (&lock2);
  pthread_mutix_unlock (&lock1);
  pthread_mutix_unlock (&lock3);
}
```

א- התוכנית עם RR רצה עד הסוף, ו FCFS לא

ב- התוכנית עם FCFS רצה עד הסוף, ו RR לא

ג-

ד-

ה-

(11