מערכות הפעלה

פתרון מבחן מועד ב', 28.03.03

מרצה: ד"ר אנה מוס
מתרגלת: גב' ז'קי איון

הנחיות:

1. משך הבחינה 3 שעות.
2. בטופס המבחן 9 דפים כולל דף זה. וודא כי כולם נמצאים בידך.
3. **יש לכתוב את התשובות בטופס המבחן.**
4. יש לענות על כל השאלות.
5. מותר להשתמש בחומר עזר לא אלקטרוני.

בהצלחה!

שאלה	ערך	הישג
1	20	
2	35	
3	45	
סכום	100	

שאלה 1 (20 נקודות)

בשאלה זו נרצה להוסיף ל-Xinu קריאת מערכת הפעלה חדשה למשלוח הודעות בשם send_ready( ).
הקריאה send_ready( ) תקבל כפרמטר מספר שלם msg. על הקריאה הזו לשלוח הודעה msg לכל התהליכים בתור ה-ready, אשר לא מחכה להם הודעה מקודם. לתהליכים בתור ה-ready שכבר מחכה להם הודעה, לא תשלח ההודעה החדשה. על הפונקציה להחזיר את מספר התהליכים אליהם נשלחה ההודעה החדשה msg.

```
SYSCALL send_ready(int msg)
{
    int  ps, pid, count = 0;
    struct pentry *pptr;

    disable(ps);
    pid = q[rdyhead].qnext;              /* the first process in the queue */
    while (pid < NPROC) {
        if ((pptr = &proctab[pid])->phasmsg == 0) {      /* no previous message */
            pptr->phasmsg++;
            pptr->pmsg = msg;
            count++;
            pid = q[pid].qnext;
        }
    }
    restore(ps);
    return count;
```

שאלה 2 (35 נקודות)

בשאלה זו נרצה לממש מנגנון חדש של סנכרון בין תהליכים ב-Xinu. המנגנון החדש נועד לאפשר למספר תהליכים לבצע קריאה/כתיבה לאותו קובץ. פעולות קריאה/כתיבה הינן קטעים קריטיים וביצען יהיה לפי הכללים הבאים:

- מותר למספר כלשהו של תהליכים קוראים להיות בקטע קריטי באותו זמן
- אם תהליך כותב כלשהו נמצא בקטע קריטי, אף תהליך אחר לא יוכל להיכנס לקטע קריטי
- לתהליכים המחכים לכתיבה תינתן עדיפות בכניסה לקטע קריטי על פני תהליכים המחכים לקריאה
- סדר הכתיבות לקובץ יהיה לפי סדר הגעת התהליכים הכותבים

לצורך מימוש המנגנון החדש נוסיף למבנה סמפור struct sentry שדות הבאים:

- ✓ **inside_count** – מסמן את מספר התהליכים הנמצאים בקטע קריטי
- ✓ **mode** – מקבל ערכים 0, 1 או 2, כאשר 0 מסמן שכעת אין אף תהליך בקטע קריטי, 1 מסמן שיש תהליכים קוראים בקטע קריטי, ו-2 מסמן שיש תהליך כותב בקטע קריטי
- ✓ **whead, wtail** – ראש וזנב של תור התהליכים המחכים לכתיבה
- ✓ **rhead, rtail** – ראש וזנב של תור התהליכים המחכים לקריאה

(במנגנון החדש לא יהיה שימוש בשדות הרגילים scount, sqhead, sqtail.)

המנגנון החדש ימומש ע"י קריאות מערכת הפעלה הבאות:

- rw_create( ) - יצירת סמפור חדש עבור המנגנון המתואר לעיל והחזרת המזהה שלו (בדומה ל-screate)
- read_wait(int sem) – המתנה לסמפור sem ע"י תהליך המבקש לקרוא מקובץ
- write_wait(int sem) – המתנה לסמפור sem ע"י תהליך המבקש לכתוב לקובץ
- rw_signal(int sem) – עזיבת קטע קריטי ע"י תהליך

הסברים נוספים לגבי אופן פעולת המנגנון:

- אם תהליך קורא מגיע בזמן שיש תהליכים קוראים אחרים בקטע הקריטי ואין אף תהליך בתור הכותבים, יוכל התהליך הקורא להיכנס לקטע הקריטי ללא המתנה.
- אם תהליך קורא מגיע בזמן שיש תהליך כותב בקטע קריטי או שתור הכותבים אינו ריק התהליך הקורא יצטרך להמתין בתור הקוראים.
- אם תהליך כותב מגיע בזמן שיש תהליכים קוראים או תהליך כותב אחר בקטע הקריטי, הוא יכנס לתור הכותבים.
- ברגע שתהליך אחרון עוזב קטע קריטי, קודם ייבדק תור הכותבים ואם הוא לא ריק, תהליך ראשון בתור הכותבים יוכל להיכנס. אם תור הכותבים ריק, אז כל התהליכים בתור הקוראים יוכלו להיכנס לקטע קריטי בבת אחד.

בעמוד הבא נתון המימוש של קריאת מערכת הפעלה read_wait():

```
SYSCALL     read_wait(int sem)
{
        int     ps;
        register struct  sentry  *sptr;
        register struct  pentry  *pptr;

        disable(ps);
        if (isbadsem(sem) || (sptr = &semaph[sem])->sstate == SFREE) {
                restore(ps);
                return(SYSERR);
        }
        if ( isempty(sptr->whead) && (sptr->mode == 0 || sptr->mode == 1) {
                (sptr->inside_count) ++;
                sptr->mode = 1;
                restore(ps);
                return(OK);
        }
        (pptr = &proctab[currpid])->pstate = PRWAIT;
        pptr->psem = sem;
        enqueue(currpid,sptr->rtail);
        resched();
        restore(ps);
        return(OK);
}
```

א. (7 נקודות)

ממשו קריאת מערכת הפעלה ( )rw_create. ניתן להניח שאתחול של השדות rhead,rtail,whead,wtail מתבצע בזמן אתחול המערכת, אין צורך לאתחל אותם בקריאה הזו.

הדרכה: מומלץ להתבונן בקוד קריאת המערכת screate().

```
SYSCALL rw_create( )
{
    int ps, sem;

    disable(ps);
    if((sem = newsem()) == SYSERR)  {
        restore(ps);
        return(SYSERR);
    }
    semaph[sem].inside_count = 0;
    semaph[sem].mode = 0;
    restore(ps);
    return(sem);
```

ב. (14 נקודות)

ממשו קריאת מערכת המעלה ( )write_wait.

הדו"כה: מומלץ להתבונן בקוד קריאת המערכת ()read_wait שהוגדרה בשאלה.

```
SYSCALL write_wait(int sem )
{
        int     ps;
        register struct  sentry  *sptr;
        register struct  pentry  *pptr;

        disable(ps);
        if (isbadsem(sem) || (sptr = &semaph[sem])->sstate == SFREE) {
                restore(ps);
                return(SYSERR);
        }
        if (  sptr->mode == 0  ) {
                (sptr->inside_count) ++;
                sptr->mode = 2;
                restore(ps);
                return(OK);
        }
        (pptr = &proctab[currpid])->pstate = PRWAIT;
        pptr->psem = sem;
        enqueue(currpid,sptr->wtail);
        resched();
        restore(ps);
        return(OK);









}
```

ג. (14 נקודות)

השלימו את הקוד של קריאת מערכת ההפעלה ( )rw_signal.

```
SYSCALL  rw_signal(int sem )
{
    int ps, pid, slist;
    struct sentry *sptr;

    disable(ps);
    if (isbadsem(sem) || semaph[sem].sstate == SFREE
       || semaph[sem].inside_count <= 0)   {
        restore(ps);
        return SYSERR;
    }
    semaph[sem].inside_count --;
    if (semaph[sem].inside_count >= 0 )   {restore(ps); return OK;}
    if ( isempty( semaph[sem].whead ) ) {
          if (isempty ( semaph[sem].rhead ) ) {
                semaph[sem].mode = 0 ;
                restore(ps);
                return OK;
          }
          semaph[sem].mode = 1 ;
          slist = semaph[sem].rhead ;
          while((pid = getfirst(slist)) != EMPTY)   {
               ready(pid);
               semaph[sem].inside_count++;
           }
           resched();
    }
    else  {
          semaph[sem].mode = 2 ;
          ready( getfirst(semaph[sem].whead) );
          semaph[sem].inside_count = 1;
          resched();
    }
    restore(ps);
    return OK;


}
```

שאלה 3 (45 נקודות)

בשאלה זו 9 סעיפים, ניקוד כל סעיף 5 נקודות.

בכל אחד מהסעיפים הבאים יש להקיף בעיגול את התשובה הנכונה ביותר. אין לסמן יותר מתשובה אחת.
סימון של יותר מתשובה אחת או סימון לא ברור יביאו לניקוד 0 לסעיף!

1. נתונה מערכת המנהלת את הזיכרון הראשי בשיטת הזיכרון הווירטואלי הממומשת ע"י Demand Paging . הזיכרון הראשי מחולק ל- **3** מסגרות אשר במצב ההתחלתי פנויות. תהליך מייצר סידרת פניות הבאה לדפים (הסדר הוא משמאל לימין)

   0, 1, 0, 5, 2, 1, 0, 3, 2, 3, 0

   מהו מספר החטאות דף עבור הסידרה הנ"ל לפי האלגוריתמים FIFO ו-LRU ?

   א. 8 עבור FIFO ו- 6 עבור LRU
   ב. 8 עבור FIFO ו- 8 עבור LRU
   (ג) 6 עבור FIFO ו- 8 עבור LRU
   ד. 6 עבור FIFO ו- 6 עבור LRU

2. שאלה זו עוסקת באלגוריתם הבנקאי לחמיקה מקיפאון. נתון מצב הבא של המערכת ברגע מסוים:

	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P_0	0	2	1	0	2	2	3	2	3	1	0	1
P_1	2	0	3	1	3	4	3	2				
P_2	1	1	2	2	1	2	2	2				
P_3	2	0	0	1	3	0	0	3				

   איזו סידרה היא סידרה בטוחה ?

   א. $< P_2, P_3, P_1, P_0 >$
   (ב) $< P_2, P_0, P_3, P_1 >$
   ג. $< P_3, P_0, P_2, P_1 >$
   ד. א' וגם ב'
   ה. לא קיימת סידרה בטוחה

3. תהליך ב- Xinu **לא יכול** לעבור ישירות (ללא מצבי ביניים)

   א. ממצב PRREADY למצב PRSUSP
   ב. ממצב PRSUSP למצב PRCURR
   ג. ממצב PRSUSP למצב PRRECV
   ד. א' וגם ב'
   (ה) ב' וגם ג'
   ו. א', ב' ו-ג'

4. במנגנון הפסיקות ב-Xinu, איפה רשומה הכתובת של שגרת הטיפול של Xinu עבור פסיקה מספר k ?

א. בכניסה k בווקטור הפסיקות
ב. בכניסה k במערך sys_imp
ג. בכניסה של sys_imp המוצבעת ע"י כניסה k בווקטור הפסיקות
ד. בכניסה של ווקטור הפסיקות המוצבעת ע"י כניסה k במערך sys_imp

5. לבקר הדיסק מגיעות בו זמנית הפניות הבאות לצילינדרים:
12, 37, 21, 48, 15, 52, 30
בהתחלה זרועת הדיסק נמצאת מעל הצילינדר 25.
בהנחה שכיוון תנועת הזרועה בעת שרות הבקשות הוא מצילינדרים בעלי מספרים נמוכים לצילינדרים בעלי מספרים גבוהים, ושזהו כיוון התזוזה בעת הגעת הבקשות הנתונות, מהו המרחק הכולל (בצילינדרים) של תזוזות הזרועה עד לסיום שרות הבקשות הנ"ל לפי האלגוריתם C-LOOK ?

א. 76
ב. 67
ג. 63
ד. 77

6. כמה בתים יש להוסיף למבנים של Xinu אם רוצים להוסיף ל- Xinu סמפור אחד נוסף (בהנחה שמשתנים מסוג char תופסים בית אחד ומשתנים מסוג short ו- int תופסים 2 בתים) ?

א. 19
ב. 7
ג. 12
ד. 60

7. שתי הפקודות האחרונות בפונקציה resched() הן:

```
ctxsw(&optr->pregs, &nptr->pregs);
return;
```

מתי התהליך המבצע את הפונקציה resched() יבצע את פקודת ה- return הנ"ל ?

א. מייד עם סיום הביצוע של הפונקציה ctxsw
ב. כאשר התהליך יבחר שוב לריצה ע"י resched() הנקרא מתהליך אחר או משגרת פסיקה
ג. פקודת ה- return הנ"ל לא תחרצע אף פעם
ד. אף תשובה אינה נכונה

א. הוא מספר שלם בין 0 ל- 29
ב. מופיעה בכניסת התהליך בטבלת התהליכים proctab
ג. מופיעה בכניסת התהליך במערך התורים q
(ד) תשובות ב' ו- ג' נכונות
ה. תשובות א' ו-ב' נכונות
ו. תשובות א', ב' ו- ג' נכונות

9. נתונה תכנית משתמש הבאה ב-Xinu:

```
#include<conf.h>
#include<kernel.h>
int sem1,sem2;
void xmain( )
{    int pid1,pid2;
     void func1(), func2();
     sem1 = screate(1);
     sem2 = screate(0);
     pid1 = create(func1, INITSTK, INITPRIO – 2, “proc1”, 0);
     pid2 = create(func2, INITSTK, INITPRIO – 1, “proc2”, 0);
     resume(pid1);
     resume(pid2);
}

void func1( )
{
     printf(“A”);
     wait(sem1);
     signal(sem2);
     printf(“B”);
}

void func2( )
{
     printf(“C”);
     wait(sem2);
     signal(sem1);
     printf(“D”);
}
```

מה יהיה הפלט של התכנית ?

א. ABCD
(ב) CADB
ג. AC
ד. CA
ה. CABD

ABC