

# מבחן במערכות הפעלה

ד"ר סיון טולדו ופרופ' חזי ישורון

סמסטר ב' תשנ"ט, מועד א', 21 יולי 1999

## הוראות

משך הבחינה שלוש שעות. מותר השימוש בכל חומר עזר שהוכן לפני הבחינה. אסור להעביר חומר עזר מתלמיד לתלמיד. יש לענות על כל השאלות. *בהצלחה!*

## שאלה 1 (36 נקודות + 10 נקודות בonus)

קטע הקוד הבא מיישם מבנה נתונים של תור fifo תוך שימוש ברשימה מקושרת דו-צדדית. התור מכיל מספרים שלמים אי-שליליים. הקוד מיועד לשימוש בתכניות עם מספר חוטים ולכן הוא משתמש במנעולים (mutexes). (מנעול שקול לסמפור בינרי, כאשר הפעולה lock שקולה ל-down והפעולה unlock שקולה ל-up).

```
typedef struct {
    queue_element* next;
    queue_element* prev;
    int data;
    mutex m;
} queue_element;

queue_element* head = null;
queue_element* tail = null;

mutex head_m;
mutex tail_m

fifo_put(int d) {
    lock(head_m);
    queue_element* new = malloc(sizeof(queue_element));
    new->data = d;
    new->prev = null;
    new->next = head;
    if (new->next != null) {
        lock(new->next);
        new->next->prev = new;
        unlock(new->next);
    } else {
        lock(tail_m);
        tail = new;
    }
}
```

```

    unlock(tail_m);
}
unlock(head_m);
}

fifo_get() {
    int d; /* return value */
    lock(tail_m);
    if (tail==null) {
        d = -1; /* signifies that the queue is empty */
    } else {
        queue_element* temp = tail;
        lock(tail->m);
        d = temp->data;
        if (tail->prev != null) {
            lock(tail->prev->m);
            tail->prev->next = null;
            tail = tail->prev;
            unlock(tail->m);
        } else {
            lock(head_m);
            head = null;
            tail = null;
            unlock(head_m);
        }
        unlock(temp->m);
        free(temp);
    }
    unlock(tail_m);
    return d;
}

```

1. מדוע איננו נועלים את איבר הרשימה החדש new ברוטינה `fifo_put`?
2. שנה את הקוד הקיים כך שהשגרה `fifo_get` לא תחזיר 1- כאשר התור ריק אלא תחכה עד שאיבר יוכנס לתור. השתמש באירועים מסוג `manual reset` של חלונות. הוסף לא יותר מ-7 שורות לקוד ומחק לא יותר מ-3. כתוב את הפתרון על גבי המבחן המודפס.
3. שימוש בשגרות אלה עלול להביא לאחיזות מוות (`deadlock`). הראה תסריט שבו שני חוטים נכנסים לאחיזות מוות.
4. הצע פתרון פשוט שימנע אחיזות מוות.
5. (בונוס; אנו מייעצים לך לענות רק לאחר שענית על שאר השאלות במבחן!) הצע (במילים) פתרון שמשיג את המטרות הבאות: (1) מונע אחיזות מוות, (2) כאשר התור ארוך, פעולות `put` ו-`get` מתבצעות במקביל, (3) זמן ריצה ממוצע קבוע אינו תלוי באורך התור לשתי הפעולות. כלומר, זמן הריצה הכולל עבור  $n$  פעולות של הכנסת איבר ו- $k$  פעולות של הוצאת איבר הוא  $O(n + k)$ .

## שאלה 2 (24 נקודות)

נכון או לא נכון? נמק!

1. כדאי להתחשב ב-rotational latency באלגוריתמים לתזמון דיסקים.
2. במחשב עם מעבד מהיר מאוד כדאי להשתמש בתזמון תהליכים בשיטת round robin, שבה כל התהליכים המוכנים לריצה נמצאים בתור יחיד והתהליך שבראש התור מקבל את קוונטת זמן, קבועה שלאחריה הוא חוזר לסוף התור.

3. במערכת עם demand paging נמדדו:

- CPU Utilization = 25%
- Paging disk Utilization = 90%
- Other I/O Utilization = 6%

איזה מהשינויים הבאים ישפר את ה-CPU Utilization (ניצולת המעבד)? (א) מעבד (CPU) מהיר יותר; (ב) רשת תקשורת מהירה יותר; (ג) גודל דף (page size) גדול יותר.

### שאלה 3 (20 נקודות)

מערכת מחשב משתמשת בדפים בגודל 4096 בתים ודיסק לדרפוף שקצב העברת הנתונים המירבית שלו 10MB/sec וזמן seek ממוצע שלו הוא 10ms (10 אלפיות שנייה). כאשר תהליך חוזר לזכרון לאחר swap, ניתן להביא לזכרון את כל ה-working set שלו (כל הזכרון הוירטואלי שהוקצה לו בפועל). נסמן את גודל ה-working set ב-s, ונניח ש  $0 < \alpha < 1$  הוא חלק הדפים שישתמשו בהם בהמשך מתוך ה-s שהוקצו.

1. מהם הערכים של  $\alpha$  שעבורם כדאי להביא את כל s הדפים לזכרון מייד לאחר ה-swap in?
2. האם ניתן לחשב באופן מדויק לחלוטין את התשובה לסעיף הראשון תוך שימוש בנתונים שצויינו?

### שאלה 4 (20 נקודות)

יש מודולים מסויימים במערכת מחשב שחייבים להיות חלק ממערכת ההפעלה ויש מודולים אחרים שניתן לממש בספריות תכנה שיופעלו כחלק מתהליך רגיל הפועל ב-user mode. למשל, השגרה open שפותחת קבצים היא חלק ממערכת ההפעלה, ואילו השגרה fopen שמספקת פונציונליות נוספת היא חלק מספרייה. היא קוראת ל-open, אך הפונציונליות הנוספת ממומשת מחוץ למערכת ההפעלה. עבור כל אחת מהדוגמאות הבאות, ציין האם מודול התכנה חייב להיות חלק ממערכת ההפעלה, ונמק בקצרה. במקרים של מודולים שאינם חייבים להיות חלק ממערכת ההפעלה אבל הם כן חלק ממנה, הסבר למה הם חלק ממערכת ההפעלה.

1. מודול UDP המאפשר לשלוח הודעות מתיבת דואר (port) במחשב אחד לתיבת דואר אחרת במחשב אחר.
2. מודול TCP המאפשר ליצור ערוץ תקשורת סדור ואמין בין port במחשב אחד ל-port במחשב אחר.
3. מודול חלונות המאפשר לתכניות לפתוח חלונות על המסך ולצייר או לכתוב לתוכם.
4. מודול המאפשר לתכניות להצפין קבצים ולגשת לקבצים מוצפנים.