

## חלק א (55 נקודות)

ענו על שתי השאלות הבאות.

### שאלה 1 (15 נקודות)

אלגוריתם lottery scheduling הינו אלגוריתם תזמון הסתברותי. כל תהליך במערכת מקבל מספר כלשהו של "כרטיסי הגרלה" (לכל כרטיס מספר ייחודי); בכל שלב, האלגוריתם מגריל מספר של אחד מכרטיסי ההגרלה שחולקו, והתהליך שמחזיק בכרטיס הגרלה זה מקבל את המעבד לפרק זמן קצוב.

א. (5 נק') נניח כי לאלגוריתם יש 20 כרטיסים. כיצד עליו לחלק אותם בין ארבעה תהליכים (A, B, C, D) כך שהתהליכים יקבלו 10%(A), 5%(B), 60%(C), 25%(D) בהתאמה?

5-D, 12-C, 1-B, 2-A

ב. (5 נק') כיצד ניתן להגיע באמצעות אלגוריתם זה לביצועים הדומים לאלו המושגים על-ידי אלגוריתם round-robin?

ניתן להגיע לביצועים דומים על ידי  
התחלפות (פיזיתית) בין תהליכים

ג. (5 נק') איך ניתן להגיע באמצעות אלגוריתם זה לביצועים הדומים לאלו המושגים על-ידי אלגוריתם shortest remaining time first? (ניתן להניח כי יש preemption, התהליכים לא מגיעים יחד, אבל זמן הביצוע ידוע כאשר התהליך מגיע.)

ניתן להגיע לביצועים דומים על ידי  
שמירת זמן תחילת תהליך (time slice) וזמן תחילת תהליך (time slice)



THE UNIVERSITY OF CHICAGO



שאלה 2 (20 נקודות)

הוכיחו כי קיימת שקילות בין העברת הודעות לבין סמפורים על ידי כך שתראו שניתן לממש את העברת ההודעות באמצעות סמפורים ולהפך.

(10 נק') א. רמז לכיוון הראשון של ההוכחה: השתמשו בחוצץ משותף המשמש להחזקת ה-mailboxes כאשר כל mailbox הוא מערך המורכב מהודעות.

אגמול: סמפורים  $e, l, s$  כש  $e, l, s = 0$  בשני  $e, l, s = N$   
 כמו כן mailbox מכיל בחוצץ משותף  
 קוד קימול  $e, l, s$  הוגשו.

$Send(msg)$	$receive(msg)$
$down(e)$ // מתקין פטוי	$down(s)$ // מתקין הוצצה
$down(l)$ // נחילה	$down(l)$ // נחילה
$enqueue(mailbox, msg)$	$enqueue(mailbox, msg)$
$up(s)$ מספר הוצצה	$up(e)$ מקום פטוי
$up(l)$ // נחילה	$up(l)$ // שמור נחילה
}	}

האגמול  
 כל  $msg$  נכנס בתור  $msg$  וזו כפי שהכין  
 יהיה שמור כל  $msg$  שיש בו צורך.

(10 נק') ב. רמז לכיוון השני של ההוכחה: השתמשו בתהליך עזר לסנכרון.

לסמפור  $s$  נבנה גימול  
 $mb$  הנחיה על התבליים  
 הצכינים עזרה לסמפור - שמירת הוצצה "0".  
 כמו כן גימול המוחלל ~~הוצצה~~ ~~הוצצה~~  
 קריאת ערך מסמפור  $||$   $valueOf(s)$   
 $recv(l)$ ;  $n = recv(mb)$ ; ~~הוצצה~~ ~~הוצצה~~  
 $send(mb, n)$ ;  $send(l, 1)$ ; ~~הוצצה~~ ~~הוצצה~~  
 $return n$   
 המשך מקום לתשובה בעמוד הבא

המשך מקום לתשובה בעמוד הבא







up(s){	down(s){
recv(l);	recv(l)
n=recv(mbox)	n=recv(mbox)
if(n=0){send(e,1)}	if(n=0){
send(mbox,n+1)}	send(mbox,n)
send(l,1)	send(l,1)
}	recv(e)
בסבב נשלח את המסר	} else {
אם זהו אינו טוען שיש עוד	send(l,1)
הנחה למערכת ואם	}
יש טוען.	בסבב: נשלח את המסר, אם n=0 ממשיך להנחה אחרת.

המשך הבחינה בעמוד הבא







שאלה 3 (20 נקודות)

5 נק' א. מנו שיקולים בעד החזקת דפים גדולים במערכת זיכרון וירטואלי.

דפים אצומו  $\in$  סחור דפס  $\in$   
 ניהול הדפים מהי יוג  $\in$  וברס פתור  
 זכרון.

5 נק' ב. מנו שיקולים בעד החזקת דפים קטנים במערכת זיכרון וירטואלי.

דפים קטנים  $\in$  תוכים  $\in$  אצומו  $\in$  לוי  
 גאומים  $\in$  אצומו  $\in$  ניומו  $\in$  אמה  $\in$   
 הזכרון Working Set קן לעצוק יוג.

10 נק' ג. יהי

•  $s$  – הגודל של תהליך ממוצע

•  $p$  – גודל הדף

•  $e$  – גודל של שורה בטבלת הדפים של תהליך

ציינו מה יהיה

• המספר המשוער של דפים שהתהליך זקוק להם.

• הגודל של טבלת דפים של תהליך.

• הבזבז עקב הריסוק הפנימי בדף האחרון של התהליך.

חשבו את התקורה הכוללת של השימוש בזיכרון הווירטואלי והסיקו ממנה מהו הגודל האופטימאלי של הדף.

~~הגודל האופטימאלי של הדף הוא  $\sqrt{2es}$  כאשר  $s$  הוא גודל תהליך ממוצע ו- $e$  הוא גודל שורה בטבלת הדפים של תהליך.~~

מספר דפים ששורס למהלך  $s/p$   
 אורס אמה דפים של תהליך  $e/p$

קסוס דקב כוסוק פנימי  $p/2$

מינימום בנקי קוצון  $s' = 0 - e$ ,  $f = \frac{p}{2} + \frac{es}{p}$

$f = \frac{p^2 + 2es}{2p} \Rightarrow s' = \frac{4p^2 - 2p^2 - 4es}{-4p^2} = 0$

$\Rightarrow p^2 = 2es, \Rightarrow p = \sqrt{2es}$   
 $p > 0$







## חלק ב (25 נקודות)

ענו על חמש השאלות הבאות. משקל כל שאלה 5 נקודות.

### שאלה 4

מהם השינויים שיש לבצע באלגוריתם LRU כדי שיהיה מותאם לניהול זיכרון מטמון של מערכת קבצים?

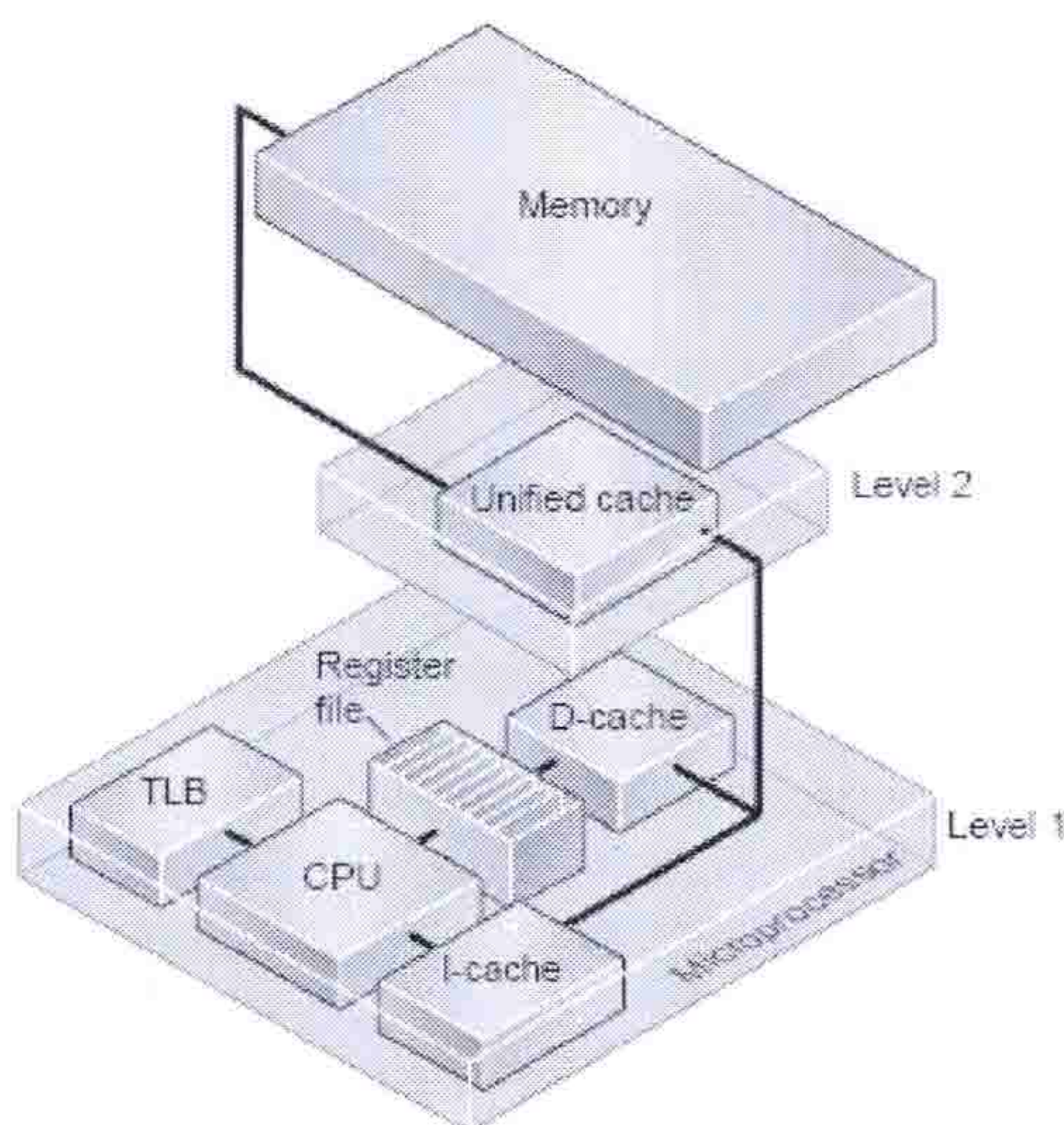
רמז – קחו בחשבון את השיקולים הבאים:

- מזעור של פניות הדיסק על ידי חיזוי של הבלוקים שיהיו בשימוש.
- ביצוע עדכונים בדיסק של הנתונים הדרושים לקונסיסטנטיות של מערכת הקבצים.

כדי להבין את הבעיה, נניח שיש לנו מערכת עם זיכרון מטמון LRU. המערכת עובדת עם נתונים שהם בלוקים. המערכת עובדת עם נתונים שהם בלוקים. המערכת עובדת עם נתונים שהם בלוקים.

### שאלה 5

באיור הבא מוצג תיאור סכמתי של מבנה המעבד עם 2 רמות של זיכרון מטמון:



איור 16

צינו יתרונות בשימוש ב- exclusive cache (L1 ו L2)

צינו יתרון אחד של השימוש ב- inclusive cache.

מקום לתשובה בעמוד הבא







$\frac{L}{L+1}$   
 exclusive - מה'ים יחד, כן מצב אחד נוצר  
 יונק, פשוט בן משנים פשוט מה'ים & thrashing  
 inclusive - מקשה מה'נה, עקבית, מולכות  
 בין מה'ים אשן כזים על משר מעבדים

## שאלה 6

הסבירו מהו עקרון הלוקליות בזמן ובמקום.

העקרון של שום נטרל time frame  
 מספיק מתקן נקרא working set קבוצת למה'ק  
 - פיתוח לעקרון זה כוונת משרכת ממון  
 /חמלה הצביע

## שאלה 7

נתון כי:

- כתובת וירטואלית הינה בעלת 64 bits
- היחידות בעלות מען הקטנות ביותר הן בגודל של 1 byte
- גודל דף הינו 8 kb
- גודל שורה של טבלת דפים הינו 4 bytes

מהו גודל טבלת הדפים?

$2^{13}$  pages מ  $2^{13}$  זכר  
 $2^{13} \cdot 2^{10} = 2^{23}$  bytes הוא הזכר  
 משרכת משרכת משרכת

## שאלה 8

הסבירו מהו RAID5 ומה יתרונו על שיטת RAID4.

RAID5 סיביות parity "כזה" בין הזכר סקרים  
 בניאז - RAID4 - דימיון במעמלה עקרון  
 עקרון נכונה לקרוא את הפאק הקונק ולעצבן parity  
 שיון צויר בקבוק והעומם ממלה בין הצופקים





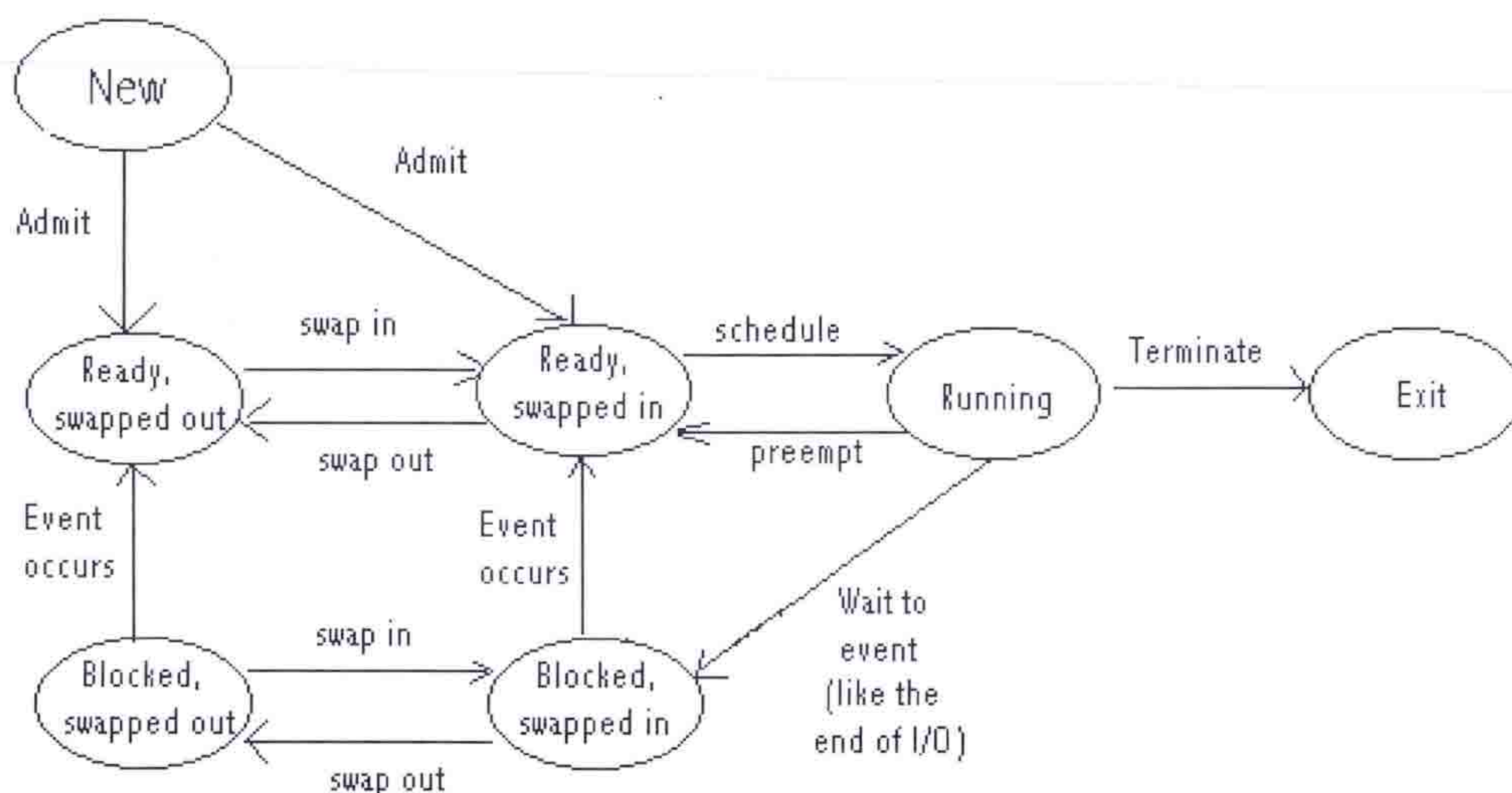


## חלק ג (20 נקודות)

ענו על ארבע שאלות רב-ברירה (אמריקאיות). משקל כל שאלה 5 נקודות.  
בכל שאלה יש לבחור את התשובה הנכונה ולהקיף בעיגול את אות התשובה שבחרתם.

### שאלה 9

לגרף עם שלושת המצבים הבסיסיים של התהליך במערכת ההפעלה הוספנו עוד כמה מצבים :



עדיפותו של תהליך P שנמצא במצב "Ready, swapped out" גבוהה מעדיפותם של כל אחד מהתהליכים שנמצאים במצב "Ready, swapped in". האם ייתכן מצב שמנגנון התזמון ישאיר את התהליך P במצב "Ready, swapped out" וימשיך לתזמן תהליכים שנמצאים בזיכרון מבלי לבצע swap in ל P?

- כן. המצב ייתכן, למשל משיקולי גודל התהליך P אשר צריך להביאו לזיכרון.
- כן. המצב ייתכן כאשר P שוהה זמן רב במצב "Ready, swapped out".
- כן. המצב ייתכן כאשר יש חשיבות מכרעת לתזמון תהליכים עם עדיפות גבוהה על פני שיקולים אחרים.
- כל התשובות הקודמות הן נכונות.

### המשך הבחינה בעמוד הבא







# שאלה 10

נתונים שני תהליכים שרצים במקביל. להלן הפסאודו-קוד שלהם:

Process 0	Process 1
<pre>while (1){     for (<u>i</u>=0; i&lt;N; i++)         down(Si);     /* Critical section */     for (i=N; <u>i</u>&gt;0; i--)         up(Si); }</pre>	<pre>while (1){     for (<u>i</u>=0; i&lt;N; i++)         down(Si);     /* Critical section */     for (i=N; <u>i</u>&gt;0; i--)         up(Si); }</pre>

כאשר  $S_i$  – ים הם  $N$  סמפורים בינאריים שאותחלו ל 1.  $N$  הוא מספר טבעי גדול מ 2.

בחרו את הטענה הנכונה:

- שני תהליכים יכולים לשהות בו זמנית בקטע קריטי
- שני תהליכים עלולים להיכנס למצב קיפאון
- הפרוטוקול מבטיח קדימות של Process 0 על פני התהליך המתחרה
- הפרוטוקול מבטיח קדימות של Process 1 על פני התהליך המתחרה
- הפרוטוקול פותר את בעיית הקטע הקריטי

# שאלה 11

בחרו סיגנל (signal) אשר אי-אפשר להתעלם ממנו (באמצעות SIG\_IGN):

- SIGINT
- SIGKILL
- SIGSEGV
- SIGALRM

# שאלה 12

איזו מבין התופעות הבאות איננה קשורה בהכרח למקביליות בין תהליכים/תהליכונים?

- קיפאון (deadlock)
- בעיית הקטע הקריטי
- סחרור בזיכרון (thrashing)
- אף תשובה קודמת איננה נכונה

בהצלחה!