

סמסטר א' תשס"ה
מועד: א' 14/02/2005
משך הבחינה: שלוש שעות
חומר עזר: דף אחד בכתב
יד, ומחשבון.

בחינה בקורס מבוא לתכנות מערכות

מרצה: ד"ר אבישי וול

הנחיות כלליות

- במבחן 7 שאלות.
- משקל כל שאלה רשום ליד השאלה.
- מספר הנקודות המכסימלי שניתן לצבור הוא 100.

תזכורת:

- $1\text{ns} = 10^{-9}\text{ sec}$, $1\mu\text{s} = 10^{-6}\text{ sec}$, $1\text{ms} = 10^{-3}\text{ sec}$
- $1\text{MB} = 2^{20}\text{ bytes} \approx 10^6\text{ bytes}$
- $1\text{GB} = 2^{30}\text{ bytes} \approx 10^9\text{ bytes}$

שאלה 1 (20 נקודות, 2 נקודות לכל מושג)

הסבר את המושגים הבאים בקצרה (משפט אחד או שניים לכל היותר):

- 1. deadlock
- 2. TLB
- 3. polling
- 4. RAID
- 5. i-node
- 6. thrashing
- 7. CPU Burst
- 8. Safe state (in banker's algorithm)
- 9. Trojan horse
- 10. working set

שאלה 2 (15 נקודות)

באלגוריתם תזמון תהליכים בשם selfish round robin (SRR) קיימים שני תורים: תור המתנה W ותור ריצה R . כל תהליך i שמוכן לריצה נכנס לתור ההמתנה W , עם עדיפות בסיסית $p(i)=0$. תהליך i עובר לתור הריצה R רק כאשר העדיפות שלו $p(i)$ גדולה או שווה לעדיפות של תהליך כלשהוא שכבר נמצא ב- R , או כאשר R ריק. רק התהליכים שנמצאים בתור הריצה R מקבלים זמן CPU, בשיטת תזמון Round Robin. בכל יחידת זמן, העדיפות של כל תהליך בתור ההמתנה W גדלה ב- a , והעדיפות של כל תהליך בתור הריצה R גדלה ב- b , כאשר $a \geq b$:

$$p(i) += a \text{ for all } i \text{ in } W$$

$$p(j) += b \text{ for all } j \text{ in } R$$

- 1. איך מתנהג SRR כאשר $a = b$?
- 2. איך מתנהג SRR כאשר $b = 0$?

שאלה 3 (15 נקודות)

במחשב המשתמש רק בזיכרון פיזי (real memory) בשיטת מחיצות משתנות (variable partitions) מתוחזק הזיכרון הפנוי ע"י רשימה מקושרת של בלוקים פנויים. כרגע הרשימה מכילה את הבלוקים הבאים, מסודרים משמאל לימין:

150KB, 360KB, 400KB, 625KB, 200KB

מערכת ההפעלה מקבלת בקשות לשטחי זיכרון בגדלים הבאים (משמאל לימין):

215KB, 171KB, 86KB, 481KB

מה מכילה רשימת הבלוקים הפנויים כאשר כל הבקשות טופלו, אם המערכת משתמשת באלגוריתם הקצאה של:

- 1. best-fit
- 2. first-fit
- 3. worst-fit

הניחו כי האלגוריתם חוזר לתחילת הרשימה (המעודכנת) לאחר טיפול בכל אחת מהבקשות.

שאלה 4 (15 נקודות)

להלן תאור אלגוריתם תזמון של ראש דיסק בשם VSCAN. נסמן את מספר הצילינדרים בדיסק ע"י C. בכל רגע נתון האלגוריתם יודע את מספר הצילינדר שמעליו הראש, ואת כיוון התנועה הנוכחי של הראש (פנימה או החוצה). האלגוריתם משתמש בפונקציה מחיר F כדי לבחור את התנועה הזולה ביותר של הראש ממקומו הנוכחי אל צילינדר מבוקש. פונקציה המחיר F תלויה במרחק בין המיקום הנוכחי והמיקום המבוקש (שיסומן ב-D), ובערך של פרמטר בשם R, ומוגדרת כך:

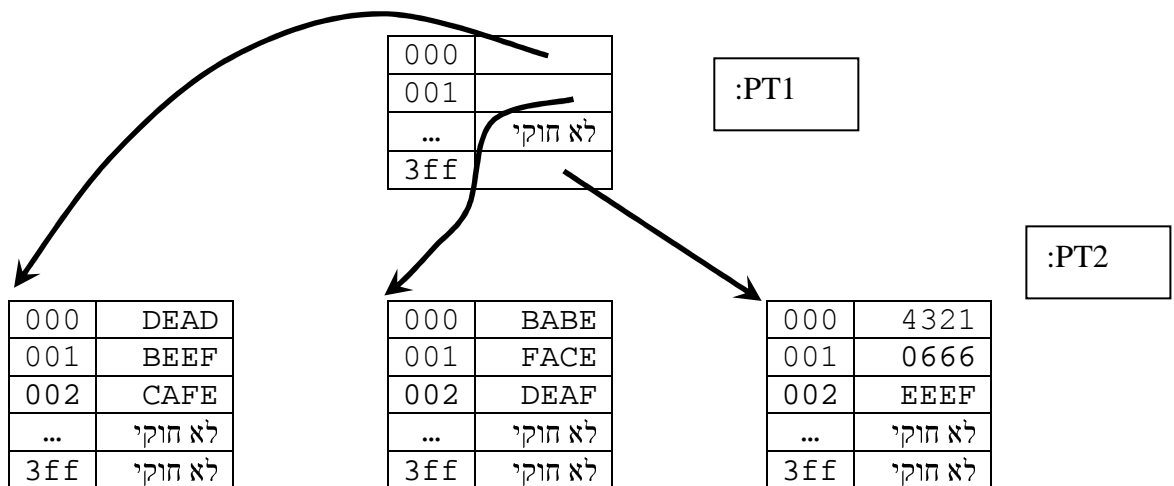
- אם הצילינדר המבוקש נמצא "עם כיוון התנועה הנוכחי" אז $F := D$
 - אם הצילינדר המבוקש נמצא "נגד כיוון התנועה הנוכחי" אז $F := D + R * C$
- מבין כל הבקשות לצילינדרים הממתינות לשרות, האלגוריתם בוחר את זו בעלת המחיר המינימלי.
1. איך מתנהג האלגוריתם כאשר $R=0$?
 2. איך מתנהג האלגוריתם כאשר $R=1$?
 3. מהם היתרונות והחסרונות של כל אחת משתי האפשרויות הנ"ל?

שאלה 5 (15 נקודות)

במחשב מסוים כתובות וירטואליות הן בנות 32 ביט, וכתובות פיזיות הן בנות 28 ביט. מערכת הדפדוף בנויה משתי רמות, וכל כתובת וירטואלית מחולקת בצורה הבאה:

10 bits: PT1	10 bits: PT2	12 bits: offset
--------------	--------------	-----------------

תוכן טבלאות הדפדוף נראה כך (כל הכתובות הקסדצימליות):

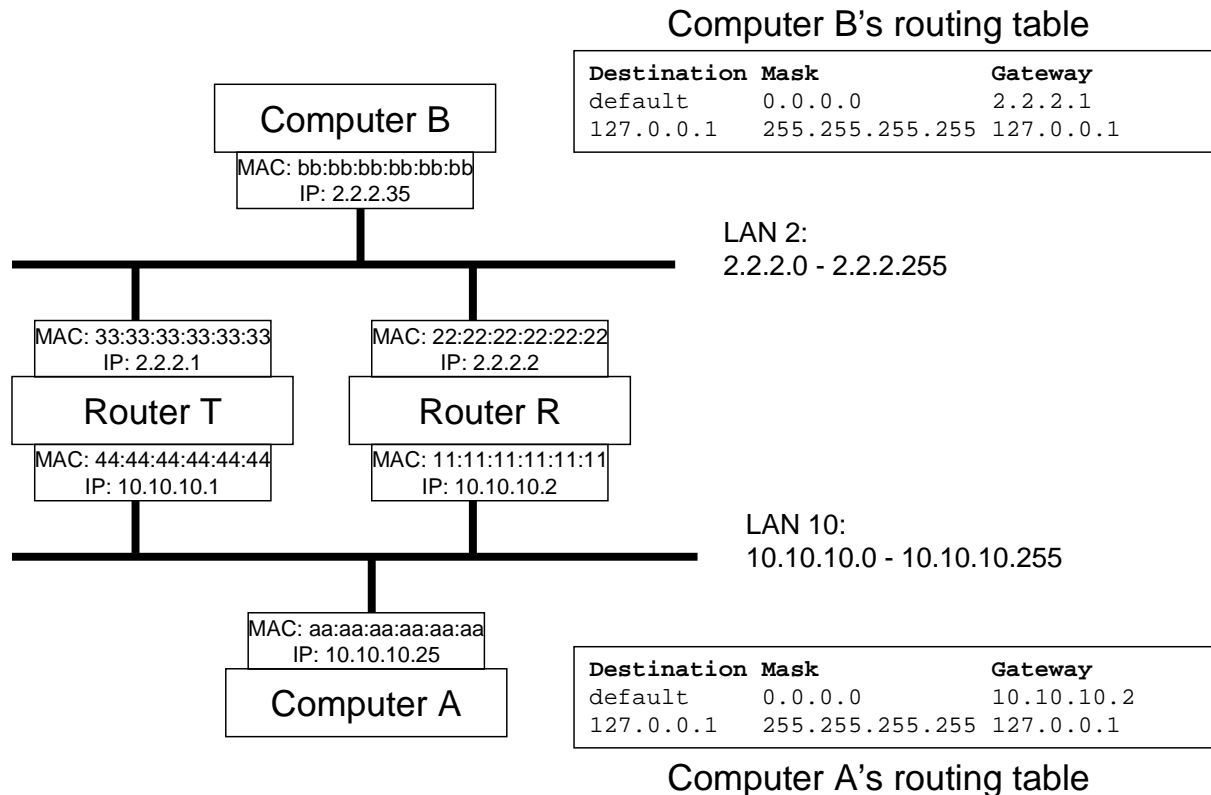


מצאו את הכתובות הפיזיות המתאימות לכתובות הוירטואליות הבאות, או ציינו כי אינן חוקיות:

1. 00000000
2. 00401ABA
3. 00100FFF
4. FFC02EEF

שאלה 6 (10 נקודות)

להלן סכמה של רשת תקשורת המורכבת משתי רשתות מקומיות (LAN10 ו-Ethernet: LAN2) וביניהם שני נתבים (R ו-T). על רשת LAN10 מחובר מחשב A, ועל רשת LAN2 מחובר מחשב B. בסכמה מופיעות גם טבלאות הניתוב בשני המחשבים. מחשב B הוא שרת המאזין על TCP port 80. הניחוי כי כל כתובות ה-MAC וכתובות ה-IP כבר ידועות למחשבים ולנתבים (אין צורך בהודעות DNS או ARP).



1. תארו את רצף ההודעות על הרשתות שמופיע כאשר מחשב A מגדיר לעצמו socket s ומבצע קריאה לפונקציה connect(s, 2.2.2.35). רשמו את ההודעות בטבלא עם העמודות הבאות:

#	רשת	MAC Source	MAC Destination	IP Source	IP Destination	TCP Source Port	TCP Destination Port	TCP SYN	TCP ACK
---	-----	------------	-----------------	-----------	----------------	-----------------	----------------------	---------	---------

2. האם היה חל שינוי ברצף ההודעות מהסעיף הקודם אילו טבלת הניתוב של מחשב A הייתה נראית כך:

Destination	Mask	Gateway
Default	0.0.0.0	10.10.10.2
2.2.2.0	255.255.255.0	10.10.10.1
127.0.0.1	255.255.255.255	127.0.0.1

אם כן – איזה הודעות היו משתנות ומה היו תכניהן החדשים?

שאלה 7 (10 נקודות)

קטע הקוד הבא אמור להיות פתרון לבעיית מניעה הדדית בין 2 חוטים (threads). המשתנים turn, t1WantstoEnter ו-t2WantstoEnter משותפים לשני החוטים. הקטע הקריטי נמצא בתוך הפונקציה Do_critical_section(). בקוד הנ"ל ישנה בעיה משמעותית – הסבירו מהי, ותארו רצף מאורעות שגורם להופעה הבעיה.

Initialization
<pre>int turn=1; Boolean t1WantstoEnter = FALSE; Boolean t2WantstoEnter = FALSE; startThreads()</pre>

Thread 1	Thread 2
<pre>while (!done) { t1WantstoEnter = TRUE; while (turn != 1) { while (t2WantstoEnter) /* loop */ ; turn = 1; } Do_critical_section(); t1WantstoEnter = FALSE; }</pre>	<pre>while (!done) { t2WantstoEnter = TRUE; while (turn != 2) { while (t1WantstoEnter) /* loop */ ; turn = 2; } Do_critical_section(); t2WantstoEnter = FALSE; }</pre>