

# מבחן במערכות הפעלה

ד"ר סיון טולדו ופרופ' חזי ישורון

סמסטר ב' תשנ"ט, מועד ב', 16 ספטמבר 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 (30 נקודות)

- עליך לממש מנגנון של סמפור לא חסום באמצעות מנגנוני mutex ו-manual-reset event של חלונות. עליך לממש את הפונקציות up ו-down ולהגדיר את המשתנים שמייצגים את הסמפור. אפשר להניח שיש בתכנית רק סמפור אחד ולהשתמש במשתנים גלובליים ובפונקציות ללא ארגומנטים. ניתן לממש את הפונקציה up ב-7 שורות ואת down ב-11. יורדו נקודות על מימושים מסובכים ו/או ארוכים. המימוש אינו צריך לענות על דרישת הוגנות כל שהיא.
- הסבר בקצרה כיצד ניתן לשפר את המימוש כך שיתנהג פחות או יותר כ-fifo. כלומר, שחוטים שקוראים ל-down בזמן שערך הסמפור 0 ישתחררו מהמתנה פחות או יותר לפי סדר הקריאה. האם ניתן להשיג התנהגות fifo מושלמת? נמק.

### שאלה 2 (20 נקודות)

נכון או לא נכון? נמק!

- ניהול הזכרון החפשי (free space) במערכת קבצים על ידי bitmap מונע בעיות אמינות עקב נפילות מתח.
- אין חסם עליון על גודל קובץ ב-Unix.
- הקטנת ה-quantum בשיטת התזמון round robin, שבה יש תור יחיד של התהליכים המוכנים לריצה וכל תהליך מקבל קוונטת זמן קבוע וחוזר לסוף התור, משפרת את זמן התגובה והנצילות.
- עקרונית, לא ניתן למנוע לחלוטין אחיזת מוות (deadlock).

### שאלה 3 (30 נקודות)

- מערכת מחשב משתמשת בדפים בגודל 4096 בתים ודיסק לדפדוף שקצב העברת הנתונים המירבית שלו 10MB/sec וזמן seek ממוצע שלו הוא 10ms (10 אלפיות שנייה). מיפוי הדפים הוא רציף, כך שדפים רצופים בזכרון הוירטואלי ממופים למסגרות רצופות בדיסק. המערכת מבוססת על demand paging, כלומר, דף בודד מובא לזכרון האמיתי בכל שגיאת דף.
- עליך מוטל להוסיף למערכת זו מנגנון prepaging שיהיה מסוגל לזהות באופן אוטומטי מצבים שבהם כדאי להביא יותר מדף בודד לזכרון.

1. הנח שבכל שגיאת דף המערכת החדשה תביא דף בודד או 10 דפים. תאר מנגנון פשוט לקבלת החלטה בזמן טיפול בשגיאת דף האם להביא דף בודד או 10. המנגנון צריך לצרוך לא יותר ממספר קטן וקבוע של בתים לכל תהליך ולא יותר ממספר קטן וקבוע של פעולות חישוב בכל שגיאת דף. המנגנון חייב לקבל החלטה נכונה במצבים שבו מתבצעות גישות אקראיות לזכרון (למשל גישות לטבלת hash) ובמצבים שבהם מתבצעות גישות רצופות לזכרון (למשל חיפוש לינארי בטבלה גדולה). מותר למנגנון לקבל החלטות שגויות במקרים אחרים.
2. האם מנגנון ה-prepaging עשוי להאיט תכניות לעומת המערכת המקורית? נמק. אם ענית שכן, מהי ההאטה המירבית?
3. האם המנגנון שתיראת בחלק א' פועל בצורה נכונה כאשר התכנית מחשבת מכפלה פנימית של שני ווקטורים ארוכים? (כלומר עבור הלולאה:  $s += a[i] * b[i]$ ;  $for (i=0; i < N; i++)$ ). אם לא, כיצד ניתן לשפר את המנגנון כך שיפעל נכון גם במקרה זה?

## שאלה 4 (20 נקודות)

מינו אותך לאחראי(ת) לטיפול במספר שרתי ftp. השרתים מחוברים לאינטרנט בקוי תקשורת מהירים (1.5 מליון סיביות לשניה) ואמינים. גולשים באינטרנט יכולים להתחבר לשרתים ולהוריד קבצים גדולים. התקשורת היא בפרוטוקול tcp. עיקר תפקידך הוא לכוון את הפרמטרים של פרוטוקול tcp במערכות ההפעלה של השרתים. הפרמטרים שניתן לכוון אותם הם: הזמן שלאחריו הודעה שלא אושרה נשלחת שוב (retransmit timer). אורך החוצץ המשמש לקבלת הודעות ואורך החוצץ המשמש לשליחת הודעות.

1. חלק מהמשתמשים של שרת מסוים התלוננו על קצב העברת נתונים נמוך מאוד. הם דווחו על קצב העברת נתונים גבוה משאר השרתים. בבדיקה הסתבר שמדובר רק במשתמשים שמחוברים לאינטרנט בקוי תקשורת מהירים עם קצב שגיאות גבוה יחסית. איזה פרמטר יש לכוון וכיצד?
  2. לאחר זמן מה התעוררה בעיה חדשה עם כל השרתים. משתמש יחיד דווח שהוא מחובר לאינטרנט בקוי תקשורת אמין ומהיר של 1.5 מליון סיביות לשניה, אך קצב העברת הנתונים מהשרתים הוא פחות מעשירית הקצב. בבדיקה עם כל המשתמשים האחרים, כולל משתמשים בעלי קוי תקשורת איטיים (56 אלף סיביות לשניה) ומשתמשים אחרים בעלי קוים מהירים לא התגלתה כל בעיה. כל המשתמשים האחרים דווחו על קצב העברת נתונים קרוב לקצב קוי התקשורת שלהם.
- ממה יכולה לנבוע הבעיה של המשתמש הבודד וכיצד ניתן לפתור אותה על ידי כוונת הפרמטרים?