

אוניברסיטת בן-גוריון בנגב, המחלקה למדעי המחשב

מועד א' במערכות הפעלה (202-1-3031) (8 ביולי 2012)

מרצים: דני הנדלר ורועי זיוון; מתרגלים: אלון גרובשטיין, איליה מירסקי, אמיר מנצ'ל ודולב פומרנץ

משך המבחן: שלוש שעות
חומר עזר: אסור
פתרי את כל השאלות: סה"כ 100 נקודות.

1. מערכת הקבצים (25 נק')

א. (15 נק') שלומציונה משתמשת במערכת קבצים יוניקס מבוססת i-nodes אשר בה מנגנון ה-cache buffer מנוטרל. הניחו כי במערכת זו גודלו של כל בלוק הוא 1KB. שלומציונה כותבת תוכנית המגרילה בכל פעם באופן רנדומלי ויוניפורמי מיקום (file position, offset) בקובץ נתון וקוראת 100 בתים של אינפורמציה ממיקום זה בקובץ.

הניחו כי לכל קובץ ממנו קוראת התוכנית של שלומציונה מוקצה מקום רציף בדיסק.

a. (8 נק') לאחר הרצת התוכנית מספר רב של פעמים על קבצים שונים שלומציונה מבחינה בתופעה הבאה: משך הזמן הנדרש להרצת התוכנית שונה עבור קבצים בגדלים שונים. עבור קבצים קטנים (לכל היותר 100KB) משך ההרצה קצר בצורה משמעותית מקבצים גדולים (יותר מ-1GB). כיצד ניתן להסביר תופעה זו בהסתמך על מאפייני המערכת? הסבירו בקצרה אך במדויק.

b. (7 נק') זרובבל טוען כי אם שלומציונה תריץ את התוכנית לעיל במערכת קבצים FAT ההבדל בזמן ההרצה בין קבצים גדולים לקטנים יקטן משמעותית וכי במערכת קבצים NTFS כלל לא יהיה הבדל. האם זרובבל צודק? הסבירו בקצרה אך במדויק.

ב. (10 נק')

a. (5 נק') במערכת הקבצים NTFS ישנו וקטור של ביטים (bitmap) הממפה את הבלוקים הפנויים בדיסק. אם מבנה נתונים זה נמחק ואין עותק נוסף שלו, האם יש דרך לשחזר אותו (ואם כן, כיצד)? ענו בקצרה אך במדויק.

b. (5 נק') מאיזה מבנה נתונים במערכת הקבצים FAT ניתן להסיק אילו בלוקים פנויים בדיסק? אם מבנה נתונים זה נמחק ואין עותק נוסף שלו, האם יש דרך לשחזר את מספרי הבלוקים הפנויים בדיסק? ענו בקצרה אך במדויק.

2. ניהול זיכרון (30 נק')

שימו לב: בשאלה זאת סעיפים א' ו-ב' בלבד מתייחסים לעבודה 3 ולמערכת ההפעלה xv6.

א. (5 נק') . ראינו כי מערכת ההפעלה xv6 תומכת בריבוי תהליכים (multiprogramming), ומכאן שבזמן נתון יכולים לרוץ מספר תהליכים שונים. בנוסף, אנו יודעים כי לכל תהליך טבלת דפים יחודית משלו (2-level page table). כאשר המעבד מריץ את פקודות התהליך, הוא עושה שימוש בחומרה על מנת לתרגם כתובות וירטואליות לפיזיות, וזאת בעזרת טבלת הדפים.

כיצד מודיעה מערכת ההפעלה למעבד מהי טבלת הדפים הנוכחית איתה עליו לעבוד? תארו בקצרה אך באופן מדויק וברור.

ב. (5 נ') . כאשר המעבד מריץ את פקודות התהליך ומתבצע שימוש בחומרה על מנת לתרגם כתובות וירטואליות לפיזיות, עלול להיווצר מצב בו מתבצעת גישה לכתובת לא חוקית בזיכרון הוירטואלי. במקרה כזה, המעבד מעביר שליטה למערכת ההפעלה xv6 על מנת שתטפל במצב. כיצד מודיע המעבד למערכת ההפעלה מהי הכתובת הוירטואלית שהגישה אליה אינה חוקית? תארו בקצרה אך באופן מדויק וברור.

ג. (5 נ') . מהי תופעת ה-thrashing? כיצד מתמודדים מנגנוני ניהול הזיכרון של מערכות הפעלה עם תופעה זו? הסבירו בקצרה אך במדויק.

ד. (15 נ') . נתונה מערכת הפעלה העובדת עם paging ואלגוריתם החלפת דפים LRU, אשר מקבלת את סדרת הגישות (ה-reference string) הבאה: 1, 3, 8, 7, 1, 3, 5, 6, 3, 9, 1, 7, 1, 3, 6.

סדר בקשת הדפים הוא משמאל לימין.

i. (7 נ') חשבו את ה-distance string המתאים ל-reference string שלעיל ולאלגוריתם LRU.

ii. (8 נ') נסמן ב-F את מספר דפי הזכרון הפיסיים (page frames) במערכת. נסמן ב-P(F) את מספר ה-page faults לו תגרום סדרת הגישות שלעיל באלגוריתם LRU. חשבו את הערך של F שיביא למינימום את הביטוי הבא: $F+P(F)$. צריך להיות ברור מתשובתכם כיצד הגעתם לערך זה.

3. סינכרוניזציה – סמפורים ובעיית המניעה ההדדית (30 נק')

שני הסעיפים הראשונים מתייחסים לבעייה הבאה.

במערכת ישנם שלושה חוטים – A, B ו-C. יש לכתוב אלגוריתם המממש קטע קריטי בעל התכונות הבאות:

- אם A נמצא בקטע הקריטי, אזי הוא נמצא בו לבדו.
- חוטים B ו-C יכולים להימצא בקטע הקריטי בו זמנית. בפרט, אם B (C) נמצא בקטע הקריטי אזי C (B) יכול להיכנס לקטע הקריטי גם כן, אפילו אם A ממתין.
- על האלגוריתם לקיים חופש מקיפאון (deadlock freedom).

א. (10 נק') ממשו את הפרוצדורות A-enter, A-exit, B-enter, B-exit, C-enter ו-C-exit המממשות את קטעי הכניסה והיציאה של שלושת ההליכים.

בסעיף זה מותר להשתמש במימוש אך ורק במשתנים הבאים:

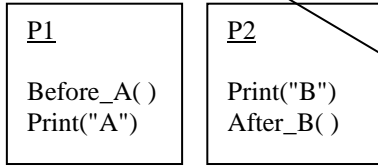
סמפורים (סמפורים מונים או בינאריים – כרצונכם) m1 ו-m2 המאותחלים שניהם לערך 1 ומשתנה c התומך בקריאות וכתובות בלבד ומאותחל לערך אפס. אין להשתמש ב-busy waiting. כלומר, תהליכים יכולים להמתין רק ע"י שינה על סמפור.

ב. (10 נק') גם בסעיף זה עליכם לממש את הפרוצדורות A-enter, A-exit, B-enter, B-exit ו-C-enter ו-C-exit. בסעיף זה מותר להשתמש אך ורק במשתנים הבאים:

משתנים m1 ו-m2 התומכים בפעולות קריאה, כתיבה ו-test-and-set בלבד ומאותחלים לאפס ומשתנה c התומך בקריאות וכתובות בלבד ומאותחל לערך אפס גם הוא. תהליכים הנאלצים להמתין יעשו זאת באמצעות busy-waiting (כלומר, פקודת await). להזכירכם, אם משתנה v תומך בפעולת test-and-set אזי פעולת v.test-and-set() מתבצעת באופן אטומי, כותבת ערך 1 למשתנה ומחזירה את ערכו הקודם.

ג. (10 נק') .

במערכת ישנם שני תהליכים, P1, P2 העשויים לבצע מדי פעם את הקוד להלן. עליכם לכתוב מימוש לפרוצדורות Before_A() ו-After_B() המבטיח את התכונות הבאות:



- לעולם אין ארבע הדפסות רצופות של "A". במילים אחרות, בין כל ארבע הדפסות של "A" יש לפחות הדפסה אחת של "B".
- אם תהליך P1 רץ לבדו (כלומר, P2 אינו מבצע את הקוד שלו באותו זמן) וההדפסה הבאה שלו אינה הרבעית ברציפות, אזי הוא יכול לבצע.
- תהליך P2 יכול תמיד לבצע עוד ועוד הדפסות.
- לעולם לא קורה מצב בו P1 ו-P2 ממתנים זה לזה (deadlock).

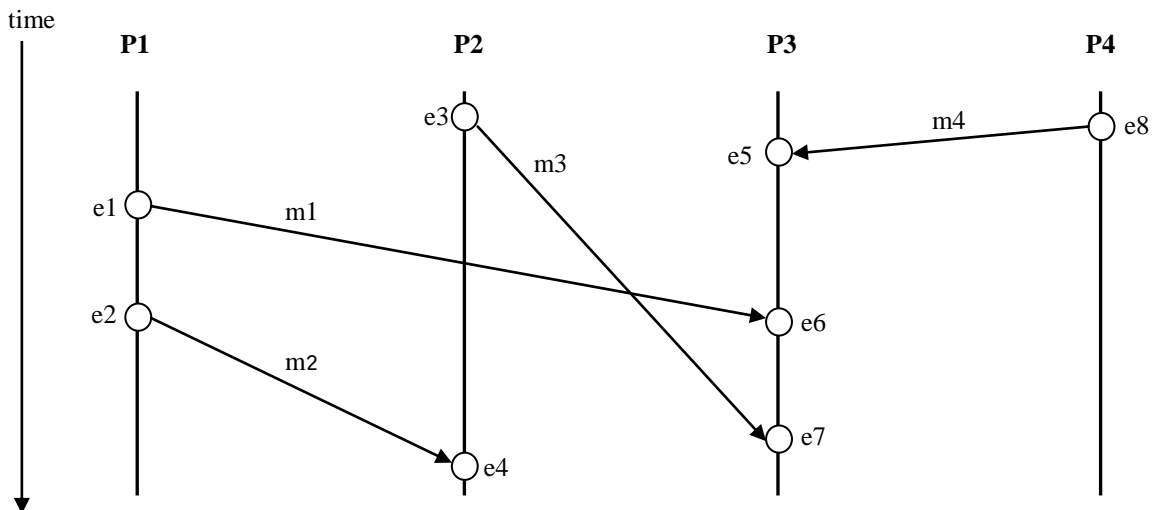
המימוש שלכם יכול להשתמש אך ורק בסמפורים (מונים או בינאריים – כרצונכם) ובמשתנים התומכים בקריאות וכתובות בלבד. אין להשתמש ב-busy waiting. כלומר, תהליכים יכולים להמתין רק ע"י שינה על סמפור.

סעיף זה שגוי ואין פתרון הממלא את כל התנאים הנדרשים בו. הבעיה היא זו: נניח כי P1 עומד לבצע את ההדפסה השלישית שלו ונעצר ממש לפנייה (כלומר, מייד לאחר ביצוע Before_A). כעת, P2 מבצע את After_B. P2 חייב לאפשר לP1 שלוש הדפסות נוספות שכן עד כמה שP2 יכול לדעת יתכן שP1 כבר ביצע את ההדפסה השלישית. לפיכך P1 יוכל לבצע ארבע הדפסות ברציפות.

עם הסטודנטים הסליחה. מלוא 10 הנקודות ניתנו בסעיף זה לכל הסטודנטים.

4. סינכרוניזציה מבוזרת ותיאום שעונים (15 נק')

נתונה הדיאגרמה הבאה של צירי הזמן של 4 מעבדים (Processors) P1...P4 ומשלוח הודעות ביניהם. נשלחות ארבע הודעות הגורמות לשמונה מאורעות (events) אצל המעבדים. את התשובות לשאלות שלהלן יש לכתוב במחברת ולא על גבי הדיאגרמה.



א. (5 נק') עבור כל אחד משמונה המאורעות (events), כתבו את חותמת הזמן של Lamport (Lamport timestamp).

ב. (5 נק') עבור כל אחד משמונה המאורעות, כתבו את חותמת הזמן הווקטורית (Vector timestamp).

ג. (5 נק') האם קיים מקרה של causality violation בתסריט המתואר בדיאגרמה לעיל? אם כן, בין אילו הודעות? נמקו תשובתכם בקצרה אך במדויק.

בהצלחה!