

מבחן במערכות הפעלה

ענת ברמלר וסיון טולדו

מועד ב' סמסטר ב' תשס"ב, 7/10/2002

הוראות

יש לענות על כל השאלות.
יש לצרף את טופס המבחן ודף העזר למחברת הבחינה. מחברת ללא טופס או ללא דף עזר תפסל.
בשאלות אמריקאיות, יש לסמן את התשובה הנכונה בעיגול על טופס הבחינה ולנמק כשנדרש נימוק. בשאלות שבהן יש צורך לנמק, תשובה ללא נימוק לא תזכה באף נקודה.
יש לענות על כל השאלות בגוף הבחינה במקום המיועד לכך. המקום המיועד מספיק לתשובות מלאות.
תשובות במחברת הבחינה לא תבדקנה.
יש למלא מספר ת"ז על כל דף של טופס הבחינה.

אסור השימוש בחומר עזר כלשהו, כולל מחשבוניו או כל מכשיר אחר פרט לעט.

הצלחה!

שאלה 1 (24%):

עבור כל אחד מהסעיפים הבאים ענה נכון לא נכון ונמק/נמקי:

1. במצב מיוחס המעבד אינו מגיב לפסיקות. נכון/לא נכון
2. ממשקים אחידים לגישה לחומרה מאפשרים להפיץ תוכנה שרצה על מגוון גדול של חומרות. כן/לא
3. כאשר מפתחים מנהל התקן להתקן זכרון מסוג חדש, למשל סוג חדש של דיסק, ניתן להגדיר למנהל ההתקן ממשק שרירותי, כל זמן שניתן להשתמש בו משפת התכנות שבה כתובה מערכת ההפעלה (בדרך כלל שפת C). כן/לא
4. בפרוטוקול - Ethernet (פרוטוקול בשכבת המיקשר, link layer) ניתן להשתמש בערוצי תקשורת שבהם כל מחשבים שומע הודעות מכל מחשב אחר, אבל אם שני מחשבים משדרים ביחד, לא ניתן להבין אף אחת מההודעות. אם מחשב מנסה לשדר ושומע שבזמן השידור שלו מחשב אחר ניסה גם כן לשדר, הוא מפסיק את השידור ומנסה לשדר שנית לאחר פרק זמן אקראי. העובדה שפרק הזמן הוא אקראי מיועדת למנוע:
 - (1) Deadlock, כלומר מצב שבו חלק מהמחשבים מפסיקים לשדר ונתקעים כן/לא
 - (2) Livelock, חלק מהמחשבים ממשיכים לשדר אך לא מצליחים להעביר מנות כן/לא
 - (3) חוסר הגינות בין המחשבים כן/לא

5. עליך לכתוב תוכנית שאמורה לבצע חישוב מתמטי מסובך עם כמות עצומה של נתונים כך שתרוץ במחשב בעל מעבד אחד בצורה מהירה. במחשב זה אין DMA לדיסק, וה-controller של הדיסק אינו מצויד ברכיב זיכרון. לכל אחד ממטרות ציין האם המטרה חשובה או לא ומדוע.

- (1) להקטין את מספר ה-PF כן/לא
- (2) לממש את התוכנית בעזרת מספר תהליכים ובכך להגדיל את המקביליות של התוכנית כן/לא

שאלה 3 (29%):

מערכת הפעלה מסויימת תומכת בשגרות הבאות לתיאום בין חוטים (threads):

- CreateMutex() מחזירה מזהה של מנעול (משתנה מניעה הדדי)
- DestroyMutex(m) מוחקת מנעול על פי מזהה
- Lock (m) נועלת מנעול m, החוט ירדם עד שישתחרר המנעול
- Unlock (m) משחררת מנעול m
- CreateEvent(e), DestroyEvent(e) יצירה ומחיקה של אירועים
- Wait(e) כניסה להמתנה לאירוע e
- Broadcast(e) שחרור כל מי שממתין לאירוע e

מערכת ההפעלה מבטיחה הגינות לנסיונות נעילה, כלומר, אם מספר חוטים נכנסים להמתנה למנעול בזה אחר זה, הם יקבלו את המנעול בסדר שבו נכנסו להמתנה. מתכנני המערכת גם החליטו, מתוך שאיפה למנוע קיפאון (deadlock), להגביל את ההמתנה למנעול (כלומר המתנה בשגרה Lock) לדקה אחת. לאחר פרק זמן זה מערכת ההפעלה מעיפה את החוט הנועל ומשחררת את המנעול (ומודיעה על כך לשאר החוטים בתהליך במנגנון שלא נתאר כאן). כמו כן, על מנת למנוע תעופות כאלה כאשר אין קיפאון, חוט שקיבל מנעול ירוץ על המעבד עד לשחרור המנעול ולא יוחלף בחוט אחר.

חלק א'

האם ניתן בכל זאת להכנס לקיפאון על ידי שימוש בשגרות הללו? נמק/נמקי

חלק ב'

עליך לממש במערכת זו תוכנית מרובת חוטים. החוטים צריכים, בין היתר, להסדיר גישה למשאב מסויים. מותר לכל היותר לחוט אחד לגשת למשאב בכל זמן נתון. ההמתנה למשאב צריכה להיות הוגנת (חוט שקרא לשגרת נעילת המשאב לפני חוט אחר צריך לקבל גישה למשאב לפני האחר). חוטים עתידיים להשתמש במשאב זמן ארוך (מעל דקה) ואסור שתגרם בשל כך תעופה. כמו כן, משיקולי יעילות יש למנוע מצב שבו חוט מתעורר מהמתנה ונכנס שוב להמתנה ללא התקדמות משמעותית (כלומר אסור ש-Broadcast יעיר יותר מחוט אחד אם רק חוט אחד יבצע דבר מה מועיל והשאר יכנסו שוב להמתנה מסוג כלשהו). אין הגבלה על מספר החוטים ואין להם מזהה מספרי.

ממש את השגרות הבאות **בגוף המבחן**:

- CreateResource() מחזירה struct שייצג משאב; הגדר/הגדירי את ה-struct
- Acquire(r) נועל את המשאב r, אולי לאחר המתנה שיתפנה
- Release(r) משחרר את המשאב r

רמז: על מנת למנוע תעופות, יש להקפיד שחוט שנועל מנעול של מערכת ההפעלה משחרר אותו אחרי פרק זמן קצר (מספר קבוע של פעולות). מותר להניח שבמימוש כזה לא יהיו תעופות גם אם מספר גדול של חוטים נכנס להמתנה למנעול של מערכת ההפעלה בפרק זמן קצר, מכיון כל אחד מהם מחזיק במנעול רק פרק זמן קצר.

דף למילוי התשובה לשאלה 3 חלק ב'. כתוב/כתבי קוד. פתרון שיחרוג מדף זה לא יבדק.

שאלה 4 (12%)

מנה (packet) ברשת תקשורת נושאת נתונים שכל אחד מהם שייך לאחת השכבות במודל השכבות של רשתות תקשורת. בכיתה למדנו את מודל שבע השכבות והסברנו שבמציאות משתמשים במודל מנוון יותר, בעל חמש שכבות. עבור כל אחד מהשכבות ציין האם נתב משנה את הנתונים של השכבה הנ"ל כאשר הוא מנתב מנה, והדגם:

השכבה הפיסית:

שכבת המקשר (link)

שכבת הרשת (Network):

שכבת ההעברה (transport):

שכבת היישום (application):