

מבחן במערכות הפעלה

חזי ישורון וסיון טולדו, מועד ב' סמסטר ב' תשס"ד
8 באוקטובר 2004

הוראות

משך הבחינה ארבע שעות. לא תינתן הארכה.
יש לענות על כל השאלות.

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. יש לצרף את טופס המבחן למחברת הבחינה. מחברת ללא טופס עזר תפסל. תשובות במחברת הבחינה לא תבדקנה.

יש למלא מספר סידורי (מספר מחברת) ומספר ת"ז על כל דף של טופס הבחינה.

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

בהצלחה!

שאלה 1

א. במערך קבצים RAID3 קצב ההעברה קטן מאשר במערך RAID1. נכון או לא? לנמק...

ב. התקורה של מימוש מדיניות LRU מדויקת לפיניו דפים בתוכנה (כלומר ללא תמיכת חומרה) הוא כ-30%. נכון או לא? לנמק...

ג. יתכזו ששתי כתובות וירטואליות שונות ימופו לאותה כתובת פיזית. נכון או לא? לנמק...

ג. במחשב עם שני מעבדים, לא יתכן שבאותו רגע ממש כל אחד מהמעבדים ימפה את אותה כתובת וירטואלית לכתובת פיזית שונה.

ד. מדוע לא כדאי להשתמש בפרוטוקול בשכבת הרשת (network layer) שפשוט שולח מנות ומחכה לאישור אחרי כל מנה, לפני שהוא שולח את המנה הבאה? יש לנמק.

שאלה 2 (20 נקודות)

א. תוכנית חלונות צריכה לפתוח קובץ ואז להריץ תוכנית אחרת שתשתמש בקובץ הפתוח (בלי לפתוח אותו שוב). תארי כיצד מבצעים הליך כזה:

ב. תארי את סדרת קריאות המערכת שתוכנית צריכה לבצע בחלונות על מנת ליצור שקע, להתחבר לשרת, לשלוח לו בקשה ולקבל תשובה, ולסיים את פעולתה. התוכנית אינה מבצעת שום דבר פרט למשימה שתוארה. אין צורך להזכיר את קריאות המערכת שבונות את הכתובת של השרת (כגון `gethostbyname`):

שאלה 3 (30 נקודות ועוד 5 נקודות בonus)

100%

נתון מחשב עם זיכרון לא-גדיף מסוג פלאש, (כמו ברכיבי DiskOnKey). הזיכרון הזה מחולק לבלוקים בגודל 512 בתים, והוא משמש לאיחסון מערכת קבצים. בגלל תכונות החומרה, לא ניתן לשכתב בלוק בודד; הפעולות היחידות המותרות הן מחיקת קבוצה של 16 בלוקים ביחד, שגורמת לכל הביטים בקבוצה לקבל את הערך 0, ואיפוס ביטים (בודדים או

בקבוצות). על מנת להתגבר על מגבלות החומרה, מערכת ההפעלה מתייחסת לזיכרון בעזרת מספר בלוק לוגי, שמתורגם בכל גישה למספר בלוק פיזי. טבלה בזיכרון הראשי שומרת את מספר הבלוק הפיזי שבו מאוחסן כל בלוק לוגי. כאשר צריך לשכתב בלוק, משכתבים אותו על בלוק פנוי (שנמחק לאחרונה וכל הביטים שלו דלוקים) ומשנים את המצביע המתאים בטבלה. כאשר כל הבלוקים בקבוצה מסוימת משוכתבים, מוחקים את הקבוצה כולה. כדי שאפשר יהיה לשחזר את הטבלה כאשר מדליקים את המחשב, שומרים יחד עם כל בלוק בזיכרון הפלאש כמות קטנה של מידע נוסף, מידע שכולל את מספר הבלוק הלוגי המאוחסן באותו בלוק, האם הוא בשימוש או שוכתב, וכולי.

מספר הבלוקים הפיזיים בהתקן הזיכרון הוא 2^{20} , והוא משמש לאחסון 16- 2^{20} בלוקים לוגיים. כמות המידע הנוסף שמשויך לכל בלוק הוא 24 סיביות.

א. המטרה הראשונה בתרגיל היא לאפשר שכתוב אטומי של בלוקים. אם המחשב נופל בזמן שכתוב של בלוק, אחרי העלייה מחדש הבלוק הלוגי צריך להכיל את ערכו או אחרי או לפני ניסיון השכתוב, אבל לא ערך אחר. הפעולה היחידה שהיא אטומית ברמת החומרה היא איפוס של סיבית בודדת. את המטרה משיגים על ידי שימוש חכם בסיביות המידע הנוסף ועל ידי סדר פעולות מסוים בזמן השכתוב. תארי/י כעת כיצד יש להשתמש בסיביות הללו. כלומר, כיצד 24 הסיביות מחולקות לשדות ומה גודל כל שדה. אין בהכרח צורך להשתמש בכל הסיביות.

ב. כעת תארי/י את סדר הפעולות בזמן שכתוב.

ג. טבלת המיפוי תופסת מקום רב בזיכרון הראשי. טבלה פשוטה תתפוס 20 סיביות לכל בלוק, כלומר כשניים וחצי מליון בתים. תאר מנגנון מיפוי מתוחכם יותר שיאפשר לדחוס את טבלת המיפוי לתוך 16 סיביות בלבד לבלוק, כלומר כשני מליון בתים בסך הכל.

ד. (סעיף בונוס) כעת נוסף דרישה חדשה, והיא שאם בזמן עלייה מחדש של המחשב מערכת ההפעלה מזהה מספר עותקים פיזיים של בלוק לוגי מסוים, ואם פעולת הכתיבה של כל העותקים הללו הסתיימה, מערכת ההפעלה תזהה את השכתוב האחרון (כלומר הנתונים העדכניים ביותר) ותמפה את הבלוק הלוגי אליו. יש לשנות או לשפר את חלוקת המידע הנוסף לסיביות ואת תהליך השכתוב.

שאלה 4 (30 נקודות)

בכרטיסים מסך (כרטיסים גרפיים) מתקדמים יש מעבד ניתן לתכנות, שניתן להשתמש בו על מנת להאיץ משימות גרפיות מורכבות. המטרה של מתכנני הכרטיסים היא לאפשר לכל תוכנית להשתמש במעבד של הכרטיס כרצונה, בהתאם לאלגוריתמים הגרפיים שהיא צריכה לבצע.

א. האם לדעתך תוכנית רגילה יכולה להוריד תוכנית לביצוע על ידי הכרטיס על ידי גישה ישירה להתקני החומרה של הכרטיס? יש לנמק.

ב. נניח שקוד שרץ ישירות על המעבד של הכרטיס הגרפי מסוגל לשנות את הצבע של כל פיקסל במסך (פיקסל הוא חלק מסך מינימלי שניתן לשלוט בצבע שלו באופן עצמאי משאר חלקי המסך). האם יש מצבים שבהם מערכת ההפעלה יכולה להרשות לתוכנית להריץ קוד שרירותי על הכרטיס?

ג. הצעתי שיפור במנגנוני החומרה של כרטיסים כאלו ביחס להנחה בסעיף הקודם, כך שבדרך כלל, תוכניות יוכלו להשתמש במעבד שעל הכרטיס. אין צורך לתאר את מנגנון החומרה, אלא רק מה הוא צריך לעשות.