

מבחן במערכות הפעלה

ד"ר סיון טולדו, מועד א', סמסטר ב' תש"ס

הוראות כלליות

- חומר העזר המותר היחיד הוא דף בגודל A4 כתוב בכתב היד שלכם (כלומר בכתב היד של התלמיד הניגש למבחן).
- חובה לצרף את דף העזר למחברת המבחן! מבחן שלא יצורף אליו הדף יפסל.
- משך המבחן שלוש שעות.
- בהצלחה!

שאלה א' (20 נקודות)

בשאלה זו עליך להשתמש בידע ובהבנה שצברת בקורס על מנת לשער את דרך פעולתה של תוכנה בשם VMware. השאלה בוחנת רעיונות עקרוניים במערכות הפעלה ולא היכרות עם תוכנת VMware; היכרות כזו אינה נדרשת על מנת לענות על השאלה. כל השערה התואמת את מה שנלמד בקורס ומנומקת בהגיון תתקבל כתשובה נכונה.

תוכנת VMware מדמה מחשב אישי שלם והיא רצה תחת מערכת ההפעלה Linux. משתמש רגיל יכול להפעיל את התוכנה והיא רצה כהליך רגיל. המחשב הוירטואלי מכיל בקרים לדיסקים, למקלדת ועכבר, למסך, לרשת Ethernet, ולכרטיס קול. כל הבקרים וירטואליים והתוכנה משתמשת במשאבים של המחשב הפיזי על מנת לדמות אותם. כך למשל מידע שנשלח במחשב הוירטואלי לכרטיס המסך מוצג על חלון רגיל ב-Linux, מידע שנשלח לדיסק של המחשב הוירטואלי נשמר בתוך קובץ רגיל במערכת הקבצים של המחשב הפיזי, וכדומה.

הרכיב היחיד במחשב הוירטואלי שאיננו וירטואלי הוא המעבד. כאשר המחשב הוירטואלי מריץ תכנית, פקודות המכונה של התכנית מורצות על ידי המעבד הפיזי ללא התערבות של תוכנת VMware. אם תוכנת VMware היתה מתערבת בכל פקודות מכונה של המחשב הוירטואלי, תוכניות היו רצות עליו באופן איטי מנשוא.

תוכנת VMware מריצה במחשב הוירטואלי בדרך כלל מערכת הפעלה שלימה, כגון Windows או Linux, ומערכת ההפעלה במחשב הוירטואלי מריצה תהליכים ותוכניות נוספות. מערכת ההפעלה שמריץ המחשב הוירטואלי נקראת האורחת ומערכת ההפעלה שמריץ המחשב הפיזי נקראת המארחת.

1. מה לדעתך מתרחש כאשר תהליך במערכת ההפעלה האורחת מבצע קריאת מערכת שאמורה לשלוח מידע לכרטיס הקול? הנח(י) שמערכת ההפעלה האורחת מבצעת את השירות מייד ללא context switch. עליך לתאר את סדר ההתרחשויות ברמת שתי מערכות ההפעלה והחומרה. בפרט, תאר(י) מה קורה כאשר מערכת ההפעלה האורחת מנסה להעביר את המחשב הוירטואלי ל-kernel mode-- וכאשר היא מנסה לגשת לבקר של כרטיס הקול.

2. המחשבים בכיתה 004 פועלים תחת האילוצים הבאים: (1) כל משתמש יכול להפעיל את תכנת VMware, (2) רק ל-root יש הרשאות גישה לקובץ במערכת ההפעלה המארחת שמכיל את הדיסק של האורחת. כיצד מתאפשרת למערכת ההפעלה האורחת גישה לדיסק?

שאלה ב' (9 נק')

חברת הסטארט-אפ המבטיחה NetBlaf פיתחה רשת תקשורת יחודית המיועדת להחליף את רשתות ה-Ethernet הנפוצות. החברה פיתחה מתגים ובקרים לרשת החדשה, וכעת הגיעה השעה לפתח פרוטוקולי תקשורת וגם device drivers עבור הבקרים. היחודיות של הרשת נובעת מכך שברמת החומרה התקשורת ברשת אמינה וסדורה, כלומר חבילות (packet) שנשלחות ממחשב אחד לאחר מגיעות כולן לבקר התקשורת של המחשב המקבל, ללא שגיאות ולפי סדר שליחתן. החברה צריכה כעת לפתח פרוטוקול דמוי TCP על גבי הרשת החדשה. מכיון שהחומרה תומכת בתקשורת אמינה וסדורה, הפרוטוקול וה-drivers יכולים להיות פשוטים מהרבה מ-TCP וה-drivers שלו.

1. האם הפרוטוקול החדש צריך להשתמש בחלונות? נמק(י).
2. האם הפרוטוקול החדש צריך להשתמש ב--sequence numbers? נמק(י).
3. אם הפרוטוקול החדש צריך להשתמש ב--slow start? נמק(י).

שאלה ג' (9 נק')

השלם(י):

1. מערכת קבצים המבוססת על לוג (יומן) צריכה לשמור בלוג מידע undo עבור פעולות מול הדיסק למקרה שבו
2. מערכת קבצים המבוססת על לוג (יומן) צריכה לשמור בלוג מידע redo עבור פעולות מול הדיסק למקרה שבו
3. (הגדר) פעולה אידמפוטנטית (idempotent) היא פעולה ש

שאלה ד' (22 נק')

שאלה זו דנה בנעילות יעילות לעצי חיפוש שבהם רוב הפעולות עוברות על מסלול בעץ מהשורש לעלה או צומת פנימי. ניתן לתמוך בפעולת מספר חוטים על ידי צירוף מנעול (mutex) לכל צומת בעץ. חוט נועל כל צומת לפני שהוא מבקר בו ומשחרר אותו כאשר הוא ממשיך במורד העץ. נעילה של צומת מונעת מחוטים אחרים לבקר את הצומת וגם לרדת לתת העץ שתלוי עליו. הבעיה במימוש כזה שכל פעולה על העץ נועלת את המנעול של השורש לזמן מה. זה מונע רמות גבוהות של מקביליות. מכיון שביקור בשורש הוא תכוף ואילו שינוי השורש הוא נדיר, אנו מעוניינים לממש מנגנון סינכרון שיקיים את התכונות הבאות:

- מספר רב של "קוראים" יכולים לקרוא את השורש בו זמנית (קוראים הם חוטים המבקרים את השורש ואינם משנים אותו).
- קוראים שונים אינם נועלים את אותו מנעול. כלומר אם חוט i וחוט j מבקרים את השורש (בו זמנית או זה אחרי זה), אין שום מנעול בתכנית שננעל על ידי שניהם.
- כותב יכול לנעול את השורש בצורה שתמנע מקוראים לבקר את השורש. כמובן שהנעילה יכולה להצליח רק אחרי שאין קוראים קודמים שמבקרים את השורש.

- מכיון שכתובה היא נדירה, מותר לנעילה עבור כותב להיות יקרה. קריאות הן שכיחות ולכן צריכות להיות זולות.

1. תאר מנגנון העונה על הדרישות. מותר להשתמש $O(P)$ זכרון, כאשר P הוא מספר החוטים בתכנית. מותר להניח שלכל חוט יש מזהה בין 0 ל- $P-1$. מותר להגדיר את המשתנים של מנגנון הסנכרון כמשתנים גלובליים. עליכם לממש ארבע פונקציות, `reader_lock()` ו-`writer_lock()`, `writer_unlock()` ו-`reader_unlock()`, שינעלו וישחררו את השורש עבור קורא ועבור כותב. יש לכתוב קוד C ולהסבירו במילים (לא יותר מחצי עמוד).

שאלה ה' (40 נק')

בשאלה זו עליך להפגין שליטה בפתרון התרגיל "שרת אינטרנט לתוכן דינמי". בשאלה זו לא יורדו נקודות על שגיאות הנובעות בבירור מהקושי לכתוב תכנית נכונה על נייר (כלומר ללא מהדר ו-`debugger`).

כמו כן, על מנת להקל על פתרון השאלה אין לבדוק האם קרתה שגיאה אחרי כל קריאה לשגרת ספריה או קריאת מערכת, ויש להניח שהקלט שהתקבל מהמשתמש תקין. מותר להניח שאורך הקלט שמקבל השרת מהשקע חסום על ידי 999 בתים ושארורך הפלט שהוא ישלח לשקע חסום על ידי 999. כמו כן, אין צורך לכלול שום קבצי `include`. להלן תבנית הקריאה למספר קריאות מערכת שאולי תזדקקו להן:

```
int socket(int domain, int type, int protocol);
int connect(int sockfd, struct sockaddr* server_name, int server_name_len);
struct hostent* gethostbyname(const char *name);
struct hostent* gethostbyaddr(const char *addr, int len, int type);
int bind (int socket, struct sockaddr* my_addr, int addr_len);
int listen(int socket, int queue_size);
int accept(int socket, struct sockaddr* client_addr, int* addr_len);
int pipe(int filedes[2]); /* [0] for reading, */
                        /* [1] for writing */
int dup2(int oldfd, int newfd);
```

מותר להניח את קיומן של שלוש שגרות עזר נוספות. השגרה

```
void get_sockaddr_in(int port, struct sockaddr_in* output);
```

מחזירה בארגומנט `output` מבנה שמייצג את כתובת המחשב שעליו התכנית רצה ואת מספר הפורט המבוקש. השגרה

```
void get_request(char* http_req, char* argv[]);
```

מפענחת את בקשת ה-`http` שבארגומנט `http_req` ומחזירה במערך `argv` את שם התכנית שיש להפעיל ואת הארגומנטים שלה. השגרה

```
void put_reply(char* content, char* http_reply);
```

בונה תשובת `http` המורכבת מ-`header` ומהתוכן שמועבר בארגומנט `content`.

1. עליך לכתוב תכנית בשם `ex-serve` שתתפקד כשרת `http` המריץ תכניות ושולח את פלטן ללקוח. לתכנית יש ארגומנט אחד, מספר `port` שלו יש להאזין. הטיפול בכל בקשה צריך להתבצע בתהליך נפרד.
2. ציין שתי סיבות נפוצות לכך שקריאת המערכת `bind` תכשל.
3. ציין סיבה נפוצה לכשלון בקריאה ל-`execv`.