

מטלת מנחה (ממ"ן) 13

הקורס: "עקרונות מערכות הפעלה"

חומר הלימוד למטלה: ראו פירוט בסעיף "רקע"

מספר השאלות: 5

סמסטר: 2024א

משקל המטלה: 10

מועד אחרון להגשה: 20.01.2024

הגשת המטלה: שליחה באמצעות מערכת המטלות המקוונת באתר הבית של הקורס.
הסבר מפורט ב"נוהל הגשת מטלות מנחה".

החלק המעשי (70%)

במערכת הפעלה xv6 מימוש הקונטיינרים מתבסס על אפשרויות ניהול של שני מרחבי משאבים (namespaces) הקיימים ב xv6 והם pid namespace ו mount namespace. בתרגיל זה נצלול למימוש מרחבי משאבים מבודדים במערכת הפעלה xv6 ובפרט למימוש של mount namespace.

מטרות

- הכרה של מערכת הפעלה xv6
- הכרת ההיבטים המעשיים של מימוש קונטיינרים
- היכרות עם מימוש של mount namespace במ"ה xv6
- תיקון באג שגורם ל kernel panic בעת השימוש ב mounted file system

רקע

(א) פרק Makefile מחוברת "Ubuntu 16.04 programming environment, making first steps" (הורידו את החוברת מאתר הקורס).

(ב) "Running and debugging xv6.pdf" (באנגלית, כולל הוראות דיבוג משורת הפקודה) ו/או "XV6 Instalation and EclipseConfig.pdf" (בעברית, מאחד התלמידים, כולל דיבוג ב ECLIPSE) מתוך maman13.zip.

(ג) פרק 6 מתוך <https://pdos.csail.mit.edu/6.828/2018/xv6/book-rev11.pdf>.

(ד) חוברת "[xv6 containers, namespaces and cgroups](https://pdos.csail.mit.edu/6.828/2018/xv6/book-rev11.pdf)" עם דגש על מימוש mount namespaces בפרק Mount namespaces in xv6.

(ה) expect - שפת סקריפט אינטראקטיבית: <https://likegeeks.com/expect-command>

(ו) במידת הצורך סרטונים על שימוש ודיבוג ב XV6 מאתר הקורס (בחלק ממ"נים). מספרי הממ"נים והדוגמאות בהם לא זהים לתוכן המטלות.

תיאור המשימה

בקובץ `maman13.zip` תמצאו תיקיה עם מערכת ההפעלה `xv6`. תיקיה זו שונה במקצת מזו שקיבלתם בממ"ן קודם מכיוון ששתלנו במערכת באג.

הפעם הבעיה מתגלה בעת ניסיון להריץ סדרת טסטים משורת הפקודה של `xv6` המופעלים ע"י קובץ הרצה `mounttest`.

בהמשך הפרטים למציאה ותיקון הבאג.
שימו לב, ממ"ן זה מתבסס על הידע אשר נצבר במהלך העבודה על הממ"נים קודמים.

הסבר מפורט

1. הפעילו את מערכת ההפעלה `xv6` כמתואר בסעיף ב' של "חומר קרע".
2. הריצו משורת הפקודה של `XV6` את התוכנית `mounttest` **פעמיים!!! (אחת אחר השנייה)** הבעיה תתגלה בהרצה השנייה. היא תתגלה גם בהרצת טסט מלא של `EXPECT`. בפתרון ביה"ס הבעיה לא קיימת לא בהרצה השנייה ולא בהרצות נוספות.

```
user@ubuntu: ~/xv6-13-ns-ss-n
5
sb: size 80 nblocks 34 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 4
5
sb: size 80 nblocks 34 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 4
5
sb: size 80 nblocks 34 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 4
5
sb: size 80 nblocks 34 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 4
5
sb: size 80 nblocks 34 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 4
5
sb: size 80 nblocks 34 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 4
5
umounta: umount returned -1
sb: size 80 nblocks 34 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 4
5
mounttest tests passed successfully
$ mounttest
Running all mounttest
umounta: umount returned -1
lapicid 1: panic: out of mount_list objects
80103c9e 8010435d 80104623 80104dce 80107f13 80106ca7 80108759 8010849e 0 0
```

- ניתן לראות כי טסט גורם לקריסת גרעין ומתקבלת הודעה של `kernel panic`.
- כדי לסיים ולחזור לשורת הפקודה צריך לסגור את חלון `XV6` של `QEMU`.

3. כמו שאפשר לראות מההודעה, מערכת לא מוצאת mount_list object שפנוי להקצות.
 4. הבעיה היא שבעת עזיבת namespace mount בפונקציה namespaceleave מקובץ namespace.c(אין ולא צריך קוד מקור בשביל הפתרון) שקוראת ל mount_nsleave מקובץ mnt_ns.c, הדברים לא התעדכנו כשורה, מדובר על בעיה בפונקציה mount_nsleave מקובץ mnt_ns.c .
 5. תקנו את הבעיה. התיקון לא ארוך. בסה"כ צריך להוסיף לפונקציה שכמעט ריקה 3 פעולות שונות(אחת מהן בעזרת קריאה לפונקציה אחרת שכבר קיימת בתוך אותו קובץ עם השם שמראה מה התפקיד שלה). אם תעשו את התיקון בשלבים, יכולות להתגלות בעיות נוספות, צריך לתקן את כולם.
- ברמה לוגית פונקציה mount_nsleave כשהתהליך האחרון עוזב את mount_ns אמורה לרוקן את ה mounts הקיימים ולרשום שאין mounts. אם תהליך שאינו אחרון עוזב את mount_ns צריך לבצע שינוי רישום בהתאם. את המידע הנוסף אפשר למצוא בחוברת ["xv6 containers, namespaces and cgroups"](#) ובקובץ mnt_ns.h .

בדיקה סופית

1. לאחר תיקון הבעיה הריצו make clean; make qemu. וודאו בפעם נוספת שאתם מסוגלים לבצע רצף פקודות אשר גרם לקריסתה של המערכת בתחילת הדרך.
2. כעת המשיכו לבדיקות regression שמטרתן לוודא כי כל הבדיקות (tests) עוברים בהצלחה. לשם כך כבו את QEMU.
3. התקינו expect (אם עדיין לא הותקן) - שפת סקריפט למיכון (automation) אינטראקציה עם פקודות shell ותוכנות אשר מורצות משורת הפקודה. לשם כך הריצו משורת הפקודה

```
sudo apt-get install expect
```

4. הריצו משורת הפקודה של מערכת ubuntu 16.04 פקודה הבאה:

```
./runtests.exp my.log
```

5. ודאו כי תוכנת סקריפט יצאה עם סטטוס 0 מיד לאחר סיומה

```
$ echo $?
0
```

6. להכרות כללית עם expect מומלץ לקרוא את פרק ה' של "חומר רקע".

פתרון ביה"ס

להריץ מתוך תיקיית הממ"ן את `make clean` ו `make qemuss`. להרצת טסט לסגור את החלון של QEMU לפני ההרצה (לא צריך וחבל לסגור את הלון הטרמינל של UBUNTU).

הגשה

יש להגיש את הקובץ `mnt_ns.c` בלבד. אין להגיש קבצים נוספים ו/או מקומפלים. ראה הוראות הגשה כלליות בחוברת הקורס.

את הקובץ/הקבצים המוגשים יש לשים בקובץ ארכיון בשם `exYZ.zip` (כאשר YZ הנו מספר המטלה). עדיף להכין את הארכיון בפורמט ZIP ב WINDOWS. אם אין אפשרות, ע"י הרצת הפקודה הבאה משורת הפקודה של Ubuntu: `zip exYZ.zip <exYZ files>`
הערה חשובה: בתוך כל קובץ קוד שאתם מגישים יש לכלול כותרת(בהערה) הכוללת תיאור הקובץ, שם הסטודנט ומספר ת.ז.

בדיקה לאחר הגשה

לאחר ההגשה יש להוריד את המטלה (חלק מעשי/עיוני) משרת האו"פ למחשב האישי לבדוק תקינות של הקבצים המוגשים (לדוגמא, שניתן לקרוא אותם). בנוסף, הבדיקה של החלק המעשי תכלול את הצעדים הבאים:

- פתיחת ארכיון `exXY.zip` בספרייה חדשה (`new folder`).
- יצירת ספרייה חדשה עם הקוד המקורי של `xv6`.
- העתקת הקובץ המוגש לספרייה עם הקוד המקורי של `xv6`.
- הרצת `make qemu` ווידוא שכל ה `target` נוצר ללא שגיאות וללא `warnings`.
- הרצת בדיקות רלוונטיות: וידוא תקינות הריצה של החלק המעשי.

החלק העיוני (30%)**שאלה 2 (5%)**

לפי מדיניות חדשה של תזמון זרוע הדיסק, בקשות מוחזקות בתור לפי סדר הגעתן. הראשונה המטופלת היא הבקשה אשר הגיע האחרונה. מדיניות זו נקראת LIFO – last in first out.

(א) מהו היתרון של המדיניות הזאת?

(ב) מהו החיסרון של המדיניות הזאת?

שאלה 3 (10%)

בכל רגע נתון גודלו של קובץ יכול לנוע בין 4Kb ל 4Mb. באיזו מבין 3 מדיניות הייתם בוחרים בשביל ש:

א. בזבוז מקום יהיה מינימאלי.

ב. זמני גישות סדרתיות יהיו מינימאליים.

- הקצאה רציפה

- רשימה מקושרת

- i-node

הניחו הנחות סבירות נוספות אם נדרש. הדגימו חישובים עליהם התבססה ההחלטה.

שאלה 4 – (10%)

תקראו את ההסבר של מודל האבטחה וניהול תהליכים במערכת אנדרואיד (פרקים 10.8.10-10.8.12 בספר הקורס, אין צורך להתייחס לפרטים בפרק 10.8.11, אלא רק למודל עצמו).

תענו על השאלות הבאות:

(א) במה המודל המתואר דומה לקונטיינרים הנידונים בחלק המעשי של מטלות הקורס ובמה הוא שונה ממנו?

(ב) באיזה סוג בידוד (הפרדה) מדובר?

(ג) מה היתרון בשיטת ניהול זיכרון ראשי COW (Copy On Write) שמערכת אנדרואיד משתמשת בה בשביל מימוש מודל תהליכים שלה?

שאלה 5 (5%)

תארו בקצרה את 2 שיטות מניעת ניסיון שינוי ה capability שניתנה למשתמש במערכת ללא תמיכת חומרה. ציינו בקצרה את היתרון והחיסרון של כל אחת מהן.

הגשת החלק העיוני

החלק העיוני יוגש כקובץ Word או pdf עם שם exYZ.docx או exYZ.pdf (כאשר YZ הנו מספר המטלה) בתוך אותו zip עם החלק המעשי. אין להגיש יותר מזיפ אחד בסה"כ!