מטלת מנחה (ממ"ן) 12

הקורס: "מערכות הפעלה"

חומר הלימוד למטלה: ראו פירוט בסעיף "רקע"

מספר השאלות: 5 | משקל המטלה: 10

סמסטר: 2025א | מועד אחרון להגשה: 28.12.2024

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

החלק המעשי (70%)

כללי

בתרגיל זה נכיר מרחבי משתמש מבודדים(קונטיינרים) במערכת הפעלה XV6. מרחבי משתמש מבודדים (Containers) משתייכים לשיטות וירטואליזציה ברמת מערכת הפעלה.

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 נכיר את מימוש מרחבי משתמש מבודדים במערכת הפעלה XV6 בכלל ומימוש pid namespace בפרט.

`

מטרות

- הכרת ההיבטים המעשיים של מימוש קונטיינרים
- היכרות עם מימוש של pid namespace במ"ה xv6
- היכרות עם unshare - קריאת מערכת (system call) ליצירת namespace חדש.
- היכרות עם שינויים בקוד של קריאת מערכת fork ליצירת תהליך חדש ושיוך של התהליך שנוצר למרחב משאבים.
- התאמה ושינוי קריאת מערכת למודעות ותמיכה במנגנון ההפרדה.

רקע

א) פרק Makefile מחוברת "Ubuntu 24.04 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 maman12.zip.

ג) פרקים 0(עד PIPES בע' 13) ופרק 1 מתוך https://pdos.csail.mit.edu/6.828/2018/xv6/book-rev11.pdf.
עם דגש על "Process overview" בסוף פרק 1 .

ד) חוברת "xv6 containers, namespaces and cgroups" עם דגש על מימוש פונקצית fork בפרק "PID 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 העכשווי.

תיאור המשימה

בקובץ maman12.zip תמצאו תיקיה עם מערכת ההפעלה xv6. תיקיה זו שונה מזו שקיבלתם בממ"ן 11 מכיוון שמדובר במערכת שנוספה לה תמיכה בקונטיינרים. יש להשתמש בתיקיה הזאת.

קבצי מקור עם הפונקציות שאין צורך לשנות לא כלולים בממ"ן.

הפעם נרחיב את קריאת המערכת מממ"ן 11 ונתאים אותה לקונטיינרים. הפרטים בהמשך.

ממ"ן זה מתבסס על הידע אשר נצבר במהלך העבודה על הממ"ן הקודם.

הסבר מפורט

1. הפעילו את xv6 (מממ"ן 12) כמתואר בסעיף ב' של "חומר קרע". הריצו משורת הפקודה את :
   make clean; make qemu
   תצרו ותתנסו בשימוש בקונטיינרים כמו שמתואר בסעיף ד' של חומר הרקע.
2. אם תעתיקו את **השינויים מהקבצים שהגשתם** בממ"ן 11 **לתוך הקבצים התואמים** בתיקייה של XV6 החדשה(**שימו לב שלא מדובר על העתקת הקבצים עצמם**) ותנסו לבנות(make qemu), הפלט יהיה(שגיאת הקומפילציה) :

```
proc.c: In function 'cps':
proc.c:826:88: error: 'struct proc' has no member named 'pid'
 printf("%s \t %d  \t SLEEPING \t %d \t \t %d \t %d \n", p->name, p->pid, extp
                                                                          ^
proc.c:827:86: error: 'struct proc' has no member named 'pid'
 cprintf("%s \t %d  \t RUNNING \t %d \t \t %d \t %d \n", p->name, p->pid, extp
                                                                          ^
<builtin>: recipe for target 'proc.o' failed
make: *** [proc.o] Error 1
user@ubuntu:~/xv6-ns$
```

3. סיבת השגיאה כמו שאפשר לראות בפלט היא שמבנה proc (PCB של תהליך) השתנה בעקבות הוספת התמיכה בקונטיינרים.
4. תבצעו התאמת קריאת המערכת מממ"ן 11 , תוסיפו "מודעות" לקונטיינרים ותרחיבו אותה. הפלט התקין אמור להיות :

```
user@ubuntu: ~/xv6
init: starting sh
$ ps
main-loop: WARNING: I/O thread spun for 1000 iterations
name     pid     state          extpid         ppid    cputime
init     1       SLEEPING       1              0       55965
sh       2       SLEEPING       2              1       15027
ps       3       RUNNING        3              2       6978
$ pouch start c1
Pouch: c1 starting
$ ps
name     pid     state          extpid         ppid    cputime
init     1       SLEEPING       1              0       55965
sh       2       SLEEPING       2              1       26633
ps       7       RUNNING        7              2       27202
pouch    5       SLEEPING       5              1       12803
sh       1       RUNNING        6              5       3771386
$ pouch connect c1

tty0 connected
ps
name     pid     state          extpid         ppid    cputime
init     1       SLEEPING       1              0       55965
sh       2       RUNNING        2              1       5162115
ps       2       RUNNING        9              1       6639
pouch    5       SLEEPING       5              1       12803
sh       1       SLEEPING       6              5       25363971
$
```

שדה EXTPID הוא PID "חיצוני"- מחוץ לקונטיינר. בשביל זה תבינו היטב את החוברת מסע' ד' של חומר הרקע ובמיוחד פרק **namespaces in xv6** ממנה. המידע הדרוש קיים ב PCB .

את מבנה ה PCB אפשר לראות בחוברת או ב struct proc בקובץ proc.h **. כעת תמיכה בקונטיינרים מקוננים לא מופעלת ב XV6 ואפשר להתבסס על זה בפתרון המטלה(אין קונטיינר בתוך קונטיינר).**

אפשר להתבסס על הפתרון ממ"ן 11 , להעתיק חלק מקבצים שהגשתם בו ולשנות חלק. הפעם הוספת קריאת המערכת צריכה להיות בדרך מקובלת, בעזרת שינוי בקובץ **syscall.h וצריך להגיש גם אותו!**

אפשר להיעזר ב https://github.com/raj-maurya/xv6-public_modifiedOS . לא צריך(והפעם אסור) לבצע את כל "המעקף" שבממ"ן 11 בעקבות איסור לשנות את **syscall.h** ולא לבצע שינויים בקבצים רלוונטיים שהיו בעקבות זה.

בקיצור, הפעם מוסיפים קריאת מערכת כמו שמקובל.

בשביל אחידות הפלטים בבדיקה הדפסת שורת הכותרת של הפלט צריך לבצע בעזרת:

שימו לב לרווח בין \t ל \t // ;("name \t pid \t state \t \t extpid \t ppid \t cputime \n")cprintf

5. לאחר תיקון הבעיה תבנו ותריצו את XV6 מחדש, תפעילו את PS , תצרו קונטיינר, תתחברו אליו, תפעילו את PS בתוך קונטיינר, תבדקו שהפלט תקין ב 2 המקרים. שימו לב לתקינות שדה EXTPID . תוודאו שאתם מבינים מתי ובאיזה מצב PID ו EXTPID שווים ומתי לא.

הערה לא חשובה: כתוצאה מצורת מימוש קלט-פלט עם תמיכה בקונטיינרים, ה TTY הלא פעיל יכול לגרום ל sh(shell) שמחובר אליו להיות פעיל(רץ) כל הזמן ולכן זמן CPU שלו עולה(סוג של בג).

- צריך להדפיס את כל התהליכים הקיימים(שנוצרו במערכת, ללא שורות ריקות בטבלת התהליכים). בשביל פשטות ואחידות הבדיקה **כל תהליך שלא נמצא במצב RUNNING אמיתי מודפס כ SLEEPING (במשמעות NOT RUNNING) .**
הסיבה לזה שבמימוש של קונטיינרים שנוספו ל XV6 חלק מתהליכים שלפי הגיון אמורים להיות SLEEPING , למעשה רוב הזמן לא במצב הזה.

6. שמות הפונקציות והקבצים

- **לקריאת המערכת צריך להיות שם cps1xx** , כש xx הן 2 הספרות האחרונות של ת״ז של הסטודנט. לדוגמא, אם ת״ז 3135678**92** אז שם קריאת המערכת צריך להיות **cps192 !!!**
- **מספר קריאת המערכת צריך להיות כמו(שווה) לספרות אחרי cps , 192 בדוגמא הנ״ל.**
- **לקובץ ps.c צריך להיות שם ללא תוספת ספרות(ps.c בלבד) !!!**

בדיקה סופית

1. לאחר תיקון באגים הריצו make clean; make qemu. וודאו בפעם נוספת שאתם מסוגלים
   - ליצור קונטיינר/ים
   - להתחבר אל הקונטיינר/ים הנוצרים ולהפעיל בהם פקודות
   - להשמיד קונטיינרים אשר נוצרו
2. כעת המשיכו לבדיקות regression שמטרתן לוודא כי כל הבדיקות (tests) עוברים בהצלחה. לשם כך כבו את QEMU.
3. הריצו משורת הפקודה של מערכת ubuntu 16.04 פקודה הבאה:

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

אם צריך, תתנו הרשאות הרצה לקובץ runtests.exp .

4. ודאו כי תוכנת סקריפט יצאה עם סטאטוס 0 מיד לאחר סיומה

```
$ echo $?
0
```

5. להכרות כללית עם expect מומלץ לקרוא את פרק ה׳ של ״חומר רקע״.

פתרון ביה״ס

להריץ מתוך תיקיית הממ״ן את make clean ו אחריו make qemuss

הגשה בזיף אחד ביחד עם החלק העיוני!

יש להגיש אך ורק את הקבצים שהיה צורך לשנות/להוסיף:

(ps.c , proc.c , syscall.h , syscall.c , usys.S , sysproc.c , user.h , defs.h ו Makefile) בלבד.

אין להגיש קבצים נוספים ו/או מקומפלים. ראה הוראות הגשה כלליות בחוברת הקורס. ראו הוראות הגשה כלליות בחוברת הקורס.

את הקובץ/הקבצים המוגש/ים יש לשים בקובץ ארכיון בשם exYZ.zip (כאשר YZ הנו מספר המטלה). עדיף להכין את הארכיון בפורמט זיפ ZIP ב WINDOWS . אם אין אפשרות, ע"י הרצת הפקודה הבאה משורת הפקודה של Ubuntu : zip exYZ.zip <exYZ files>

<u>הערה חשובה : בתוך כל קובץ קוד שאתם מגישים יש לכלול כותרת(בהערה) הכוללת תיאור הקובץ, שם הסטודנט ומספר ת.ז.</u>

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 ספריה 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%)

האם ניתן וכדאי להשתמש באלגוריתם LRU בצורתו הטהורה לצורך פינוי דפים (page eviction)?

שאלה 3 – (5%)

האם דף עם תוכן מסוים וזהה יכול להיות בו זמנית בשתי קבוצות עבודה (working sets) של 2 תהליכים שונים? נמקו.

שאלה 4 – (10%)

טבלת הדפים של תהליך במערכת עם זיכרון וירטואלי נראית כך. כל המספרים הם דצימליים, מתחילים מאפס, וכל הכתובות הן כתובות של בייט בזיכרון. גודל הדף הוא 1024 בתים.

Page Number	Valid bit	Frame Number
0	1	4
1	1	3
2	0	1
3	1	4
4	0	1
5	1	0

א. לאילו כתובות פיזיות, אם יש כאלו, ימופו הכתובות הוירטואליות הבאות: 942, 2211, 5399.

ב. האם יש שגיאות בטבלת הדפים, אם כן מה הן.

ג. האם הכתובת הפיזית המתאימה לכתובות הוירטואליות 942 תשתנה אם גודל הדף יהיה 2048 בתים?

שאלה 5 – (10%)

א. תארו את ההליך תרגום כתובת לוגית בעלת 32 סיביות לכתובת פיזית כשבמערכת גודל דפי זיכרון היא 4MB (4 מגה בית).

ב. חשבו את גודל טבלת הדפים בהנחה שאורך שורה בטבלה הוא 4B (4 בית) ושכל תהליך מקבל את מרחב זיכרון וירטואלי מרבי.

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!