

האוניברסיטה הפתוחה

20594

## **מערכות הפעלה**

חוברת הקורס – סתיו 2015א

כתב: דוד שריאל

אוקטובר 2014 – סמסטר סתיו- תשע"ה

**פנימי – לא להפצה.**

© כל הזכויות שמורות לאוניברסיטה הפתוחה.

## תוכן העניינים

א	אל הסטודנט
ב	1. לוח זמנים ופעילויות
ד	2. תיאור המטלות
ד	3. התנאים לקבלת נקודות זכות
ה	4. הדרכה לפתרון מטלות התכנות
1	ממ"ץ 11
5	ממ"ץ 12
9	ממ"ץ 13



## אל הסטודנט,

אנו מקדמים את פניך בברכה עם הצטרפותך אל הלומדים בקורס " מערכות הפעלה".

בחוברת זו תמצא את לוח הזמנים, תנאים לקבלת נקודות זכות ומטלות.

לקורס קיים אתר באינטרנט בו תמצאו חומרי למידה נוספים, אותם מפרסם/מת מרכז/ת ההוראה. בנוסף, האתר מהווה עבורכם ערוץ תקשורת עם צוות ההוראה ועם סטודנטים אחרים בקורס. פרטים על למידה מתוקשבת ואתר הקורס, תמצאו באתר שה"ס בכתובת:

<http://telem.openu.ac.il>

מידע על שירותי ספרייה ומקורות מידע שהאוניברסיטה מעמידה לרשותכם, תמצאו באתר הספרייה באינטרנט [www.openu.ac.il/Library](http://www.openu.ac.il/Library)

אפשר לפנות אלי בדואר אלקטרוני [davidsa@openu.ac.il](mailto:davidsa@openu.ac.il) או בשעות הנחיה הטלפונית המפורסמות באתר הקורס. הפרטים הללו מצויים גם באתר המחלקה למדעי המחשב [telem.openu.ac.il/cs](http://telem.openu.ac.il/cs). פגישות יש לתאם מראש.

חשוב להדגיש כי התקשוב בקורס ישמש ערוץ רשמי בין צוות ההוראה של הקורס לבין הסטודנט, כלומר חובה על כל סטודנט להתעדכן באופן שוטף על הנעשה בקורס דרך אתר הבית. כל ההודעות - הן בנושאים אקדמיים והן בנושאים מנהליים - יועברו דרך אתר הבית בלבד, ולא יישלחו הודעות בדואר רגיל. סטודנטים אשר אין להם גישה לרשת האינטרנט יוכלו לגשת למרכז הלימוד הקרוב לביתם ולהשתמש במעבדת המחשבים שם. לפרטים מלאים על מרכזי הלימוד ושעות הפתיחה, ניתן להתקשר למוקד הפניות בטלפון : 09-7782222.

אל אתר הבית של הקורס ניתן לגשת מדף הבית של החטיבה למדעי המחשב:

<http://telem.openu.ac.il/cs>

בברכת לימוד פורה ומהנה,

דוד שריאל

מרכז ההוראה בקורס

1. לוח זמנים ופעילויות (20594/ 2015א)

תאריך אחרון למשלוח הממ"ן (למנחה)	מפגשי ההנחיה*	יחידת הלימוד המומלצת	תאריכי שבוע הלימוד	שבוע הלימוד
		הכרת UBUNTU יחידה 1 מספר הקורס	24.10.2014-21.10.2014	1
		יחידה 2	31.10.2014-26.10.2014	2
		יחידה 2 קריאה של יחידות 10.3 ו 11.4 מספר הקורס (באופן עצמאי)	7.11.2014-2.11.2014	3
		יחידה 2	14.11.2014-9.11.2014	4
		יחידה 6	21.11.2014-16.11.2014	5
ממ"ן 11 27/11/2014		יחידה 6	28.11.2014-23.11.2014	6
		יחידה 3	5.12.2014-30.11.2014	7
		יחידה 3	12.12.2014-7.12.2014	8
		יחידה 3	19.12.2014-14.12.2014 (ד-ו חנוכה)	9

\* התאריכים המדויקים של המפגשים הקבוצתיים מופיעים ב"לוח מפגשים ומנחים".

לוח זמנים ופעילויות - המשך

תאריך אחרון למשלוח הממ"ן (למנחה)	מפגשי ההנחיה*	יחידת הלימוד המומלצת	תאריכי שבוע הלימוד	שבוע הלימוד
ממ"ן 12 25/12/2014		יחידה 3 קריאה של יחידות 10.4 ו 11.5 מספר הקורס (באופן עצמאי)	26.12.2014-21.12.2014 (א-ד חנוכה)	10
		יחידה 4	2.1.2015-28.12.2014	11
		יחידה 4 קריאה של יחידות 10.6 ו 11.8 מספר הקורס (באופן עצמאי)	9.1.2015-4.1.2015	12
		יחידה 5	16.1.2015-11.1.2015	13
ממ"ן 13 22/1/2015		יחידה 5 קריאה של יחידות 10.5 ו 11.7 מספר הקורס (באופן עצמאי)	23.1.2015-18.1.2015	14
		יחידה 9 ושיעור חזרה קריאה של יחידות 10.7 ו 11.9 מספר הקורס (באופן עצמאי)	2.2.2015-25.1.2015	15

מועדי בחינות הגמר יפורסמו בנפרד

\* התאריכים המדויקים של המפגשים הקבוצתיים מופיעים ב"לוח מפגשים ומנחים".

## 2. תיאור המטלות

קרא היטב עמודים אלו לפי שתתחיל לענות על השאלות

חוברת זו מכילה מידע על המטלות ואת המטלות עצמן.  
פתרון המטלות הוא חלק בלתי נפרד מלימוד הקורס - הבנה מעמיקה של חומר הלימוד דורשת תרגול רב. המטלות יבדקו על-ידי המנחה ויוחזרו לך בצירוף הערות המתייחסות לתשובות.

לכל מטלה נקבע משקל. ניתן לצבור 36 נקודות. חובה להגיש את כל המטלות.

ללא צבירת 36 נקודות בהגשת מטלות  
לא ניתן יהיה לגשת לבחינת הגמר

**לתשומת לבכם!**

ציון סופי מחושב רק לסטודנטים שעברו את בחינת הגמר בציון 60 ומעלה והגישו את כל המטלות בציון 60 לפחות.

כל סטודנט יכין את הממ"נים לבדו. אין להגיש את הממ"נים בזוגות (או קבוצות)!

## 3. התנאים לקבלת נקודות זכות

- א. הגשת מטלות במשקל כולל של 36 נקודות לפחות עם ציון מינימלי של 60 נקודות בכל אחת מהמטלות שהוגשו.
- ב. ציון של לפחות 60 נקודות בבחינת הגמר.



#### 4. הדרכה לפתרון תרגילי התכנות

תרגילי התכנות בקורס זה דורשים מאמץ ניכר. התרגילים לכשעצמם אינם קשים באופן מיוחד אולם הם דורשים הכרה והבנה טובה של החומר המוצע כחומר רקע (ראו סעיף "חומר קרע" בגוף כל ממ"ן)

למרות שהקוד הנדרש בסופו של דבר בתרגילי התכנות איננו ארוך, סביר להניח כי תקדישו לתרגילים שעות רבות. תכנות מערכת הפעלה, דורש ניסיון, ולמרבה העצב רכישת הניסיון כרוכה לרוב גם בהקדשת זמן. עם זאת, התרגילים תוכננו כך שיעסקו מעט ככל האפשר בנושאים שמטבעם הם טכניים בלבד.

בפתרון התרגילים אנו מציעים את השלבים הבאים:

א. קראו היטב את דרישות התרגיל והבהירו לעצמכם מה הבעיות שעלולות להתעורר בעת יישומו.

ב. קראו את החומר המוצע כחומר רקע (ראו סעיף "חומר קרע" בגוף כל ממ"ן). לצורך זה מצויים

בידכם ארבעה מקורות, עיינו בהם על פי הסדר הבא:

1. ספר הקורס, Modern Operating Systems, המספק את הרקע התיאורטי.
2. המדריך למתכנת המערכת, [The GNU C library reference manual](#), מתאר את פעולת קריאות המערכת ברוב מערכות UNIX הקיימות
3. הפקודה "man command-name" ב-UNIX מאפשרת לקבל מידע על פקודות, פונקציות ספרייה, וקריאות מערכת, כפי שהן ממומשות במערכת שבידך.
4. מידע נוסף שמכיל דוגמאות קוד והסברים אפשר למצוא באינטרנט, בפרט באתרים שכתובותיהם מצויים בקטגוריה "אתרים ברשת" (ראו את הדף הראשי של אתר הקורס).

ג. בעת כתיבת הקוד, הקפידו על הכללים המקובלים, בהנדסת תוכנה. רוב הדרישות המפורטות כאן מוכרות לכם בודאי מקורסים קודמים אומנם ישנן דרישות ייחודיות לקורס במערכות הפעלה. לקיום הדרישות הללו קיימת השפעה על ציון הממ"ן:

1. מתן שמות משמעותיים למשתנים.
2. הימנעות משימוש במספרים שרירותיים.
3. כתיבת פונקציות קצרות.

4. תיעוד סביר. הכוונה לתיעוד מתומצת של פעולות התוכנית, של פונקציות ושל משתנים. כמו כן, יש לרשום בתחילת כל קובץ קוד שמוגש את הפרטים האישיים (שם מלא ומספר סטודנט) ותיאור קצר של תוכן הקובץ.
5. יש להקפיד על שימוש בשמות המוגדרים במטלה.
6. אין להשתמש ב goto. ליציאה מלולאות ניתן להשתמש במידת הצורך ב continue או break.
7. מבנה מדורג. מודולים ופונקציות קצרות וללא אפקטים משניים.
8. Indentation.
9. משפטי תנאי קצרים.
10. כל יציאה בגלל שגיאה חייבת להיות מתועדת. למשל, באמצעות הפונקציה perror().
11. בכל מקרה יש לבדוק את הערך המוחזר על ידי קריאות מערכת.
12. בכל מקרה יש לבדוק את נכונות הקלט.
13. התוכנית לא תיפול עקב שגיאה/תקלה כלשהי. במידה וקורה אירוע בלתי צפוי, על התוכנית להודיע על כך ולסיים את עבודתה.
14. אין להשתמש בפונקציה system().
15. יש לשחרר את כל המשאבים שאינם בשימוש.
16. הוראות קומפילציה יש לכתוב בשפת ההוראות של תוכנית השירות make ולהגישם בקובץ בשם makefile.
17. חובה להשתמש בדגל "Wall" (flag) בזמן קומפילציה התוכנית.

## בונוס

במקרים יוצאי דופן, כאשר מוגשת תוכנית טובה במיוחד או כזו שעושה למעלה ממה שנדרש, תישקל האפשרות להוסיף עד 5 נקודות בונוס. בכל מקרה שהנכם מתכוונים להגיש תוכנית מעין זו, שימו לב כי:

1. כל הדרישות מהתוכנית המקורית יתקיימו.
2. כל תוספת תהיה מתועדת היטב.
3. תוספות המכילות שגיאות עלולות להוריד מהניקוד הסופי גם אם התוספות לא נדרשו במטלה. כוונת טובות אינן מובילות בהכרח לתוצאה הרצויה.

# מטלת מנחה (ממ"ן) 11

הקורס: "מערכות הפעלה"

חומר הלימוד למטלה: ראו פירוט בסעיף "רקע"

משקל המטלה: 12

מספר השאלות: 7

מועד אחרון להגשה: 27/11/2014

סמסטר: 2015א

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

## החלק המעשי (70%)

### כללי

בממ"ן זה עליכם לממש שתי ספריות לעבודה עם תהליכונים (threads) ברמת המשתמש (user-level). אחת הספריות תממש סמפורים בינאריים לעבודה עם קטעים קריטיים וספרייה שנייה תממש מספר פונקציות המאפשרות יצירה והרצה של תהליכונים ברמת המשתמש ומדידת זמן הריצה ל profiling של תוכניות המשתמשות בספרייה זו.

### מטרה

- הכרת ההיבטים המעשיים של מימוש תהליכונים ברמת המשתמש
- שימוש בסינגלים
- שימוש ב-non-local branching
- timers
- profiling
- קטעים קריטיים

### רקע

א) פרקים 2.3.5, 2.5.1, 2.2.1, 2.2.2, 2.2.3 בספר של Tanenbaum, "Modern operating systems".  
ב) [פרק 24.3](#) של The GNU C library  
ג) [פרק 23.4](#) של The GNU C library  
ד) פרק "Libraries" מחוברת "Ubuntu 12.04 programming environment, making first steps"  
ה) man pages של Linux - מידע על קריאות מערכת ופונקציות הבאות: alarm, sigfillset, sigaction, swapcontext, getcontext, makecontext, steitimer, kill, getpid

### תיאור המשימה

בממ"ן זה עליכם לממש שתי ספריות סטטיות:

1) libut.a - ספרייה פשוטה לעבודה עם תהליכונים ברמת המשתמש, שה-API שלה מוגדר בקובץ ut.h. קובץ זה מכיל תיאור מפורט לגבי תפקידה של כל פונקציה שעליכם לממש (אין לשנות קובץ זה, אך כמובן

שבמידת הצורך ניתן להגדיר פונקציות עזר בקובץ C). הספרייה תתמוך רק בפעולות הבסיסיות ביותר, שהן יצירת התהליכונים, הרצתן ותזמונן. על מנת שלא להפוך את המשימה למסובכת מדי, הספרייה תממש רק מודל פשוט של שימוש בתהליכונים המבוסס על ההנחות הבאות:

א. כל תהליכון מריץ פונקציה אינסופית שמקבלת פרמטר יחיד מטיפוס `int` ומחזירה `void`. לא נטפל בסיום תהליכונים ובבדיקת סטטוס היציאה.

ב. אין הוספה דינאמית של תהליכונים. המשתמש קודם יצור את כל התהליכונים, וא"כ יקרא ל-`ut_start()` כדי להריץ את כל התהליכונים.

ג. כל התהליכונים הם בעלי אותה עדיפות. תזמון התהליכונים יהיה בשיטת `round-robin`, כאשר גודל ה-`quantum` הוא שנייה אחת.

ד. שימו לב שלא הגדרנו מצב `blocked` לתהליכונים. זאת מפני שבמודל שלנו ההנחה היא שתהליכונים לא מבצעים פעולות הגורמות לחסימה (`blocking calles`). לאחר הביצוע של `ut_start()`, כל תהליכון יכול להיות באחד משני המצבים - רץ או מוכן לריצה. וודאו שאתם מבינים כי בהנחה כזאת כלל לא נצטרך לשמור את מצב התהליכונים מכוון שמנגנון התזמון שלנו תמיד יבחר את התהליכון הבא בתור ויריץ אותו.

בשלב ראשון של הכנת הממ"ן קראו את הסעיפים א, ב, ג) מחומר רקע והריצו והבינו את התוכניות `demo1.c`, `demo2.c`, `demo3.c` שסיפקנו לכם. התוכנית הראשונה מדגימה כיצד מתאפשר לשים "שעון מעורר" לתהליך ב `Linux`. התוכנית השנייה מרחיבה את הראשונה ומדגימה כיצד אפשר ליצור 2 ניבים של ריצה בתוכנית באמצעות המנגנון המכונה `non-local jumping`. התוכנית השלישית מדגימה כיצד אפשר לבצע רישום של זמן ריצה של תוכנית לצורך ה-`profiling`.

בשלב שני עליכם לממש את הממשק המוגדר בקובץ `ut.h`. הממשק מגדיר פונקציות לאתחול הספרייה, ליצירת תהליכון חדש ולהרצת התהליכונים שנוצרו. `ut.h` מממשת את מודל התהליכונים הפשוט שתיארנו לעיל. שימו לב ש `demo2.c` מדגימה כיצד ליצור 2 תהליכונים. אתם מתבקשים להכליל את הפתרון למספר תהליכונים. לכן, לאחר שהשלמתם את שני השלבים הקודמים כל שנותר לעשות הוא להעביר חלקים של הקוד מ `demo2.c` ל `ut.c` עם שינויים מינוריים.

ב `ut.h` עליכם לממש את `ut_get_vtime` המשמשת למדידת זמן הריצה של תהליכון. השתמשו בקוד של `demo3.c` שמשמשת בבשעון מסוג `ITIMER_VIRTUAL` שישלח סיגנל `SIGVTALRM` כל `100msec` (10 פעמים לשנייה). בכל פעם שהסיגנל מתקבל, יש להוסיף `100msec` לשדה הזמן הווירטואלי של התהליכון האקטיבי בזמן קבלת הסיגנל.

(2) `libbinsem.a` - ספרייה של סמפורים בינאריים שנועדו לשימוש ע"י התהליכונים מהסעיף הראשון. הקובץ `binsem.h` מגדיר את הטיפוס של סמפור בינארי ומתאר את הפונקציות הרלוונטיות (אין לשנות קובץ זה). עליכם לממש את הפונקציות שמוצגות בקובץ זה, תוך כדי שימוש במקרו `xchg()` המוגדר בקובץ `atomic.h`. כמו כן, תסתמכו על העובדה שהחלפת התהליכונים מתבצעת כתוצאה מקבלת הסיגנל `SIGALRM` כדי לממש את ההמתנה ב- `binsem_down()` (כפי שפורט בסעיף הקודם, לתהליכונים שעליכם לממש לא מוגדר מצב `blocked`. יש לדמות את המצב ע"י כך שתהליכון ה"ממתין" בסמפור מייד לאחר קבלת ה-CPU ישלח סיגנל `SIGALRM` שיגרום להפעלת המתזמן ומעבר לתהליכון הבא).

לצורך הבדיקה של שתי הספריות סיפקנו לכם פתרון של בעיית הפילוסופים הסועדים בקובץ ph.c. בעיית הפילוסופים הסועדים מתוארת בפרק 2.5.1 בספר של Tanenbaum. כל פילוסוף רץ כהליכון נפרד (לצורך זה משתמשים בספריית התהליכונים שהממשק שלה הוגדר ב ut.h. התהליכונים משתמשים בסמפורים שהוגדרו ב binsem.h). התוכנית תופעל ע"י הפקודה "ph <N>", כאשר N (בטווח מ-2 עד 32) הוא מספר התהליכונים (פילוסופים). התוכנית תופסק ע"י הקשת "Ctrl-C", לפני היציאה יודפסו זמני השימוש ב-CPU של כ"א מהתהליכונים.

כדי לקמפל את תוכנית הפילוסופים עם הספריות שתכתבו, תשתמשו ב Makefile שסיפקנו. שימו לב שעליכם לשנות את ה Makefile לפני ההגשה (ראו סעיף "הגשה" בהמשך).

### טיפול בשגיאות

יש תמיד לבדוק את ערכי החזרה של קריאות מערכת ופונקציות סטנדרטיות של C. במקרה של כשלון, יש לפעול כפי שמוגדר בקבצים ut.h ו-binsem.h. בנוסף, במקרה של כשלון המערכת תוך כדי ביצוע של signal handler(s) בספריית התהליכונים, יש להודיע על השגיאה באמצעות perror() ולהפסיק את הביצוע ע"י exit(1).

### הגשה

יש להגיש **פ**ל קבצי הקוד Makefile המייצר שתי ספריות סטטיות: libbut.a ו-libbinsem.a. אין להגיש קבצים מקומפלים. ראה הוראות הגשה כלליות בחוברת הקורס.

את הקבצים המוגשים יש לשים בקובץ ארכיון בשם exYZ.zip (כאשר YZ הנו מספר המטלה). הכנת קובץ ארכיון מתבצעת ע"י הרצת הפקודה הבאה משורת הפקודה של Ubuntu:

```
zip exYZ.zip <ExYZ files>
```

**הערה חשובה: בכל קובץ קוד שאתם מגישים יש לכלול כותרת הכוללת תיאור הקובץ, שם הסטודנט ומספר ת.ז.**

### פתרון ביה"ס

קיבלתם את שתי הספריות, libbut.a ו-libbinsem.a, כפי שמומשו על ידינו. תוכלו להיעזר בהן בהכנת הממ"ן/ למשל לקמפל את תוכנית הבדיקה ph עם ספרייה אחת משלכם (שאותה אתם רוצים לבדוק) וספרייה השנייה של פתרון ביה"ס.

**הערה: תוך כדי העבודה על הממ"ן תצטרכו להכיר ולהבין מספר נושאים שאינם פשוטים - זהו הקושי של ממ"ן זה. יחד עם זאת, הממ"ן לא ידרוש מכם הרבה עבודת תכנות. ניתן לממש את שתי הספריות בכ-100 שורות קוד בסה"כ.**

### החלק העיוני (30%)

שאלה 2 (5%)

תארו את הסוגים הבאים של מערכות ההפעלה:

- א) מערכת הפעלה הפועלת באצווה (batch system).
- ב) מערכת הפעלה עם ריבוי תהליכים (multiprogramming system).
- ג) מערכת הפעלה עם חלוקת הזמן (time-sharing system).
- ד) מערכת הפעלה מבוזרת (distributed system).

שאלה 3 (5%)

- א) מהי פעולת ה TRAP (TRAP instruction). תארו מתי היא מתבצעת ומה קורא בעת ביצועה.
- ב) מהו ההבדל בין פעולת ה TRAP לפסיקת החומרה (hardware interrupt)?

שאלה 4 (5%)

הסבר מהו ההבדל בין תוכנית לתהליך.

שאלה 5 (5%)

הסבר את מדוע פתרון התור (strict alternation), איננו מהווה פתרון סביר. איזה תנאי/ים הוא מפר.

שאלה 6 (5%)

האם מדיניות הוצאת תהליכונים מתור המתנה של סמפור יכולה להיות שונה מ first in first out? אם כן, הבר מדוע. אם לאו, תאר את הבעיה.

שאלה 7 (5%)

הוכיחו כי בפתרון של Peterson תהליכים אינם ממתניים זמן אינסופי על מנת להיכנס לקטע קריטי. בפרט הוכיחו כי תהליך שרוצה להיכנס לקטע קריטי לא ממתין יותר ממה שלוקח מתהליך אחר להיכנס ולעזוב את הקטע הקריטי.

### הגשת החלק העיוני

החלק העיוני יוגש כקובץ Word או כקובץ pdf. שם הקובץ צריך להיות exYZ.pdf או exYZ.doc (כאשר YZ הנו מספר המטלה).

# מטלת מנחה (ממ"ן) 12

הקורס: "מערכות הפעלה"

חומר הלימוד למטלה: ראו פירוט בסעיף "רקע"

משקל המטלה: 12

מספר השאלות: 5

מועד אחרון להגשה: 25/12/2014

סמסטר: 2015א

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

## החלק המעשי (80%)

בממ"ן זה עליכם לכתוב ספרייה להקצאה דינמית של זיכרון התומכת בריבוי תהליכונים. את הממשקים שיש לממש, ניתן לראות בקובץ `mtmm.h`.

### מטרה

- ניהול הזיכרון
- ריבוי תהליכונים

### רקע

(א) הסבר ליצירת ספריות סטטיות בפרק "Libraries" מתוך החוברת, `Ubuntu 12.04 programming environment`, `Making first steps`

(ב) פרק 8.7, ספר מאת Ritchie & Kernighan "C Programming Language"

(ג) הסבר לניהול זיכרון בסביבה מרובת תהליכונים בתוך "Implementation of multithread memory management"

### תיאור המשימה

תכנות מקבילי בשפות C ו C++ בתוכניות כגון שרתי אינטרנט, מנהלי מסדי נתונים, שרתי חדשות, ויישומים מדעיים הופכים הקצאה ושחרור של זיכרון לצוואר בקבוק שמגביל ביצועי מערכת ומגביל יכולת ניצול של מספר גדול של מעבדים במערכות שהן מרובות מעבדים.

בממ"ן זה נממש ספרייה להקצאה דינמית של זיכרון שתנצל בצורה טובה את קיום ריבוי המעבדים. קראו את חומר הרקע בסדר המופיע לעיל. בפרק 8.7 של K&R תראו פתרון פשוט לבעיית הקצאה ושחרור של זיכרון המשתמש בערמה (heap) אחד. בהסבר המופיע ב Implementation of multithread memory management תראו כיצד לממש את ספרייה להקצאת זיכרון בצורה המתאימה לריבוי תהליכונים. במשפט אחד אנהנו נשתמש במספר ערמות – אחת לכל תהליכון ועוד ערמה נוספת עבור כל התהליך כולו.

## טיפול בשגיאות

יש לבדוק את ערכי החזרה של קריאות מערכת ופונקציות סטנדרטיות של C. במקרה של כשלון, יש לפעול כפי שמוגדר בקובץ mtmm.h.

## הגשה

יש להגיש כל קבצי הקוד ו Makefile המייצר ספרייה סטטית libmtmm.a. את הקבצים המוגשים יש לשים בקובץ ארכיון בשם exYZ.zip (כאשר YZ הנו מספר המטלה). הכנת קובץ ארכיון מתבצעת ע"י הרצת הפקודה הבאה משורת הפקודה של Ubuntu:

```
<zip exYZ.zip <ExYZ files
```

הערה חשובה: בכל קובץ קוד שאתם מגישים יש לכלול כותרת הכוללת תיאור הקובץ, שם הסטודנט ומספר

ת.ו.

## פתרון ביה"ס

קיבלתם ספרייה סטטית libmtmm.a, כפי שמומשו על ידינו. תוכלו להיעזר בה בהכנת הממ"ן, למשל-לקמפל איתה את תוכניות הבדיקה שלנו וכמובן עם תוכניות בדיקה שלכם.



## החלק עיוני (20%)

שאלה 1 – (5%)

מהי טבלת דפים מהופכת. תארו כיצד משתמשים ב TLB בשילוב עם הטבלה המהופכת ומדוע תמיכת החומרה חיונית במקרה של תרגום כתובת וירטואלית כאשר משתמשים בטבלת דפים מהופכת

שאלה 2 – (5%)

תארו את שיטות הערכוב (hashing) אשר נועדה להתמודד עם המקרה של TLB miss בעת תרגום כתובת וירטואלית לכתובת פיזית כאשר משתמשים בטבלת דפים מהופכת

שאלה 3 – (5%)

קראו פקד 2.2 של p39-huck.pdf המצורף ל maman12.zip. בפרק זה מובא חישוב של גודל ה storage הדרוש למיפוי של זיכרון פיזי בגודל של 32 Mb. מהו ה storage הדרשו למיפוי של זיכרון פיזי בגודל 1Gb?

שאלה 4 – (5%)

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 בייטים.

Page Number	Valid bit	Frame Number
0	1	4
1	1	7
2	0	-
3	1	2
4	0	-
5	1	0

לאילו כתובות פיזיות, אם יש כאלו, ימופו הכתובות הוירטואליות הבאות: 1052, 2221, 5499.

## הגשת החלק העיוני

החלק העיוני יוגש כקובץ Word או כקובץ pdf. שם הקובץ צריך להיות exYZ.pdf או exYZ.doc (כאשר YZ הנו מספר המטלה).



# מטלת מנחה (ממ"ן) 13

הקורס: "מערכות הפעלה"

חומר הלימוד למטלה: ראו פירוט בסעיף "רקע"

משקל המטלה: 12

מספר השאלות: 5

מועד אחרון להגשה: 22/1/2015

סמסטר: 2015א

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

## החלק המעשי (80%)

### כללי

בחלק המעשי נכתוב שתי תוכניות קטנות המממשות פונקציונאליות של מנהל מערכת קבצים ext2 על דיסקט 1.44 Mb.

### מטרה

הכרת מערכת קבצים ext2

### רקע

- (א) פרק 13.2 ב [Glibc manual](#) המתייחס לפונקציות lseek, open, close, read, write.
- (ב) קובץ ext2.pdf המתאר את ה layout של ext2.
- (ג) [The second extended file system](#) המכיל פירוט של כל הקבועים, המבנים והאלגוריתמים. חלק זה נועד לסימוכין ולא נדרש לקרוא אותו במלואו.

### תיאור המשימה

עליכם לכתוב 2 תוכניות:

```
my_dir <absolute_path_to_directory_residing_on_the_floppy_disk>
```

ו

```
my_rm <absolute_path_to_directory_or_file_residing_on_the_floppy_disk >
```

אשר הראשונה מביניהן (my\_dir) מדפיסה את תוכן הספרייה שנתיבה מצוין כפרמטר התוכנית. התוכנית השנייה מבצעת מחיקה של הספרייה או של הקובץ שנתיבם מצוין כפרמטר התוכנית.

## תוכנית my\_dir

- (1) בעקבות הרצתה של הפקודה `my_dir < abs_path_to_directory_residing_on_the_floppy_disk >` התוכנית תבדוק אם הספרייה נמצאת במערכת הקבצים ext2 אשר על גבי הדיסקט. במילים אחרות `my_dir` בודקת האם נתיב המצוין כפרמטר של התוכנית `my_dir` נמצא על הדיסקט.
- (3) לשם פשוטות נניח שהפרמטר `dir_name` מציין נתיב מלא המתחיל ב `./`. לדוגמא, אם בשורש של ext2 נמצאת ספרייה בשם `a` ובתוכה נמצאת ספרייה `b`, אז הרצת הפקודה `my_dir /a/b`, תסתיים בהצלחה והתוכנית תדפיס את תוכן הספרייה ותחזיר סטאטוס 0.
- (4) הפורמט המודפס חייב להיות זהה לפורמט של פתרון ביה"ס.
- (5) במידה והנתיב `abs_path_to_directory_residing_on_the_floppy_disk` אינו נמצא על ה ext2, התוכנית `my_dir` תודיע הודעת שגיאה ותחזור עם סטאטוס 1.
- (5) במקרה של כישלון של קריאת מערכת כלשהי התוכנית תחזור עם סטאטוס 1.

## תוכנית my\_rm

- (1) בעקבות הרצה של `my_rm < abs_path_to_dir_or_file_residing_on_the_floppy_disk >` התוכנית תמחק את הקובץ (או את הספרייה עם כל תכולתה) ממערכת הקבצים ext2.
- (2) כמו במקרה של התוכנית `my_dir`, גם כאן נניח שהנתיב הוא נתיב יחסית לשורש של ext2 ושהנתיב מתחיל ב `./`.
- (3) אם המחיקה הצליחה, התוכנית `my_rm` תחזור עם סטאטוס 0. אחרת התוכנית תחזיר סטאטוס 1.
- (4) במקרה של כישלון של קריאת מערכת כלשהי, התוכנית תחזור עם סטאטוס 1.

קראו בעיון את הקובץ `ext2.pdf`. `ext2.pdf` מסביר את מהו `layout` של מערכת הקבצים ext2. במידת הצורך תוכלו לעיין ב [The second extended file system](#) המכיל פירוט של כל הקבועים, המבנים והאלגוריתמים של ext2.

## הגשה

יש להגיש קבצי קוד וקובץ Makefile שמייצר קבצי הרצה בשם `my_dir` ו `my_rm`. אין להגיש קבצים מקומפלים. את הקבצים המוגשים יש לשים בקובץ ארכיון בשם `exYZ.zip` (כאשר YZ הנו מספר המטלה). הכנת קובץ ארכיון מתבצעת ע"י הרצת הפקודה הבאה משורת הפקודה של Linux:

```
zip exYZ.zip <ExYZ files>
```

**הערה חשובה: בכל קובץ קוד שאתם מגישים יש לכלול כותרת הכוללת תיאור הקובץ, שם הסטודנט ומספר ת.ז.**

**פתרון ביה"ס**

קיבלתם את התוכניות my\_rm ו my\_dir כפי שמומשו על ידינו. שימו לב שאתם צריכים כונן דיסקטים וירטואלי עם מערכת הקבצים ext2. להכנת הדיסקט בצעו את השלבים הבאים:

- 1) save floppy.iso (from maman13.zip) to some folder on Ubuntu. E.g. in /tmp/floppy.iso
- 2) sudo rm /dev/fd0
- 3) sudo ln -s /tmp/floppy.iso /dev/fd0
- 4) sudo losetup /dev/loop0 /tmp/floppy.iso
- 5) sudo mkdir /media/floppy
- 6) sudo mount /dev/loop0 /media/floppy

כעת /dev/fd0 יהיה מקושר לקובץ floppy.iso ו /media/floppy יכיל את הקבצים והספריות הבאות:

```
a b foo1 foo2 foo3 lost+found
```

ואם תריצו את my\_dir תקבלו:

```
27-Jun-2009 16:34 .
27-Jun-2009 16:34 ..
27-Jun-2009 16:00 lost+found
27-Jun-2009 16:33 a
27-Jun-2009 16:33 b
27-Jun-2009 16:33 foo1
27-Jun-2009 16:33 foo2
27-Jun-2009 16:34 foo3
```

אתם יכולים גם ליצור ספריות וקבצים חדשים בדיסקט ע"י יצירתם ב /media/floppy. אל תשכחו להריץ sync כדי לגרום למערכת הפעלה לכתוב את השינויים בפועל על גבי הדיסקט הווירטואלי. אחרת התוכנית my\_dir "תראה" את השינוי כעבור דקה כאשר מערכת ההפעלה תחליט לשמור את ה buffer cache.

כדי שתוכלו להריץ את הפיתרון, יש לוודא שהרשאות x במחרוזת ההרשאות של הקבצים של פיתרון בה"ס נמצאות במצב "דלוק". כדי "להדליק" אותה במידה והיא "כבויה" יש להריץ משורת הפקודה של UNIX את הפקודה:

```
chmod +x my_rm my_dir
```

החלק העיוני (20%)

**שאלה 1 (5%)**

מהן ההשגות כלפי מדיניות LRU לתזמון זרוע הדיסק?

**שאלה 2 (5%)**

מערכי דיסקים RAID level 2 ו RAID level 3 מסוגלים להמשיך לעבוד כאשר אחד מהדיסקים במערך מתקלקל. יחד עם זאת, Level 2 דורש מספר רב יותר של דיסקים עודפים. אז מדוע יש בכלל עניין כלשהו בשיטה הזאת?

תזכורת - קוד המינג :

בהנתן מילה בת 4 סיביות :

סיבית b1	סיבית b2	סיבית b3	סיבית b4
----------	----------	----------	----------

קוד המינג שלה הוא :

P1	P2	B1	P3	B2	B3	B4
----	----	----	----	----	----	----

כאשר

P1 = Even Parity of b1, b2, b4

P2 = Even Parity of b1, b3, b4

P3 = Even Parity of b2, b3, b4

לדוגמא : המינג קוד של מילה בת 4 סיביות (משמאל לימין – מ – most significant bit ל least significant bit) 1101 יהיה 1100110.

**שאלה 3 (5%)**

מערכת הקבצים של מערכת הפעלה מסוימת משתמשת בשיטת ה I-node.

- גודל הבלוק במערכת הקבצים הוא 0.5 Kbyte
- כתובת הבלוק היא 4 בתים (bytes)
- 12 שדות של ה I-node יכולים להחזיק ישירות כתובת הבלוק בדיסק
- שדה נוסף אחד נועד להחזיק כתובת של ה single indirect block
- עוד שדה נוסף אחד נועד להחזיק כתובת של ה double indirect block
- ועוד שדה נוסף אחד נועד להחזיק כתובת של ה triple indirect block

גודלו של קובץ מסוים במערכת 316 Kbyte. מהי כמות הבלוקים שדרושה להחזקת קובץ זה במערכת הקבצים (לא כולל את הבלוק שמכיל את ה i-node של הקובץ)?

#### שאלה 4 (5%)

תארו את שיטת ה ACL (access control lists) ל domain control.

#### **הגשת החלק העיוני**

החלק העיוני יוגש כקובץ Word במערכת הפעלה Windows. שם הקובץ צריך להיות exYZ.doc (כאשר YZ הנו מספר המטלה).