

האוניברסיטה הפתוחה

20594

## **מערכות הפעלה**

חוברת הקורס – אביב 2013ב

כתב: דוד שריאל

מרץ 2013 - סמסטר אביב- תשע"ג

**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"ן 11
7	ממ"ן 12
15	ממ"ן 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"ם 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)

ניתן לפנות אלי בשעות הקבלה ביום ה' בין השעות 08:00-10:00 בטלפון 02-6773316 או בפקס' 02-6510125 או בדואר אלקטרוני [davidsa@openu.ac.il](mailto:davidsa@openu.ac.il). פגישה 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 / ב'2013)**

שבוע לימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח ממ"ן (למנחה)
1	8.3.2013-3.3.2013	הכרת UBUNTU יחידה 1 מספר הקורס		
2	15.3.2013-10.3.2013	יחידה 2		
3	22.3.2013-17.3.2013	יחידה 2 קריאה של יחידות 10.3 ו 11.4 מספר הקורס (באופן עצמאי)		
4	29.3.2013-24.3.2013 (ב-ו פסח)	יחידה 2		
5	5.4.2013-31.3.2013 (א-ב פסח)	יחידה 6		
6	12.4.2013-7.4.2013 (ב יום הזכרון לשואה)	יחידה 6		מועד הגשת ממ"ן 11 11.04.2013
7	19.4.2013-14.4.2013 (ב יום הזכרון) (ג יום העצמאות)	יחידה 3		
8	26.4.2013-21.4.2013	יחידה 3		
9	3.5.2013-28.4.2013 (א ל"ג בעומר)	יחידה 3		

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

לוח זמנים ופעילויות - המשך

שבוע הלימוד	תאריכי שבוע הלימוד	יחידת הלימוד המומלצת	מפגשי ההנחיה*	תאריך אחרון למשלוח הממ"ן (למנחה)
10	10.5.2013-5.5.2013 (ד יום ירושלים)	יחידה 3 קריאה של יחידות 10.4 ו 11.5 מספר הקורס (באופן עצמאי)		מועד הגשת ממ"ן 12 09.05.2013
11	17.5.2013-12.5.2013 (ג-ד שבועות)	יחידה 4		
12	24.5.2013-19.5.2013	יחידה 4 קריאה של יחידות 10.6 ו 11.8 מספר הקורס (באופן עצמאי)		
13	31.5.2013-26.5.2013	יחידה 5		
14	7.6.2013-2.6.2013	יחידה 5 קריאה של יחידות 10.5 ו 11.7 מספר הקורס (באופן עצמאי)		מועד הגשת ממ"ן 13 06.06.2013
15	14.6.2013-9.6.2013	יחידה 9 ושיעור חזרה קריאה של יחידות 10.7 ו 11.9 מספר הקורס (באופן עצמאי)		

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 שהיו בשימוש.
16. הוראות קומפילציה יש לכתוב בשפת ההוראות של תוכנית השירות make ולהגישם בקובץ בשם makefile.
17. חובה להשתמש בדגל (flag) "-Wall" בזמן קומפילצית התוכניות

## בנוס

במקרים יוצאי דופן, כאשר מוגשת תוכנית טובה במיוחד או כזו שעושה למעלה ממה שנדרש, תישקל האפשרות להוסיף עד 5 נקודות בנוס. 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.
4. כוונת טובות אינן מובילות בהכרח לתוצאה הרצויה.
4. ציון המטלה לא יעלה על 100 נקודות.

# מטלת מנחה (ממ"ן) 11

הקורס: "מערכות הפעלה"

חומר הלימוד למטלה: ראו פירוט בסעיף "רקע"

משקל המטלה: 12 נקודות

מספר השאלות: 7

מועד אחרון להגשה: 11.04.2013

סמסטר: 2013ב

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 בסיגנלים
- 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)

[http://www.gnu.org/software/libc/manual/html\\_mono/libc.html#toc\\_Signal-Handling](http://www.gnu.org/software/libc/manual/html_mono/libc.html#toc_Signal-Handling) (פרק 24.3)

ג)

[http://www.gnu.org/software/libc/manual/html\\_mono/libc.html#system-V-contexts](http://www.gnu.org/software/libc/manual/html_mono/libc.html#system-V-contexts)

ד) פרק "Libraries" מחוברת "Ubuntu 9.10 programming" environment

ה) man pages של Linux - מידע על קריאות מערכת ופונקציות הבאות:  
alarm, sigfillset, sigaction, swapcontext, getcontext,  
makecontext, steittimer, 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 ותזמון. 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 חוספה 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 calls). 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 מכיוון שמנגנון התזמון שלנו תמיד יבחר את התהליכון הבא בתור ויריץ אותו.

בשלב ראשון של הכנת הממ"ן קראו את הסעיפים א), ב) וג) מחומר רקע והריצו והבינו את התוכניות demo1.c, demo2.c, demo3.c שסיפקנו לכם. התוכנית הראשונה מדגמה 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 הקובץ ut.h. הממשק מגדיר פונקציות לאתחול הספרייה, ליצירת תהליכון חדש ולהרצת התהליכונים שנוצרו. ut.h מממש 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 שנוותר 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 שמוצהרות 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"מתין" 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 כתהליכון 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 **כל** קבצי הקוד Makefile המייצר שתי ספריות סטטיות:

`libbinsem.a` ו-`libut.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, `libut.a` ו-`libbinsem.a`, כפי שמומשו על ידינו. תוכלו להיעזר בהן בהכנת הממ"ן/ למשלי 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 מכס 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 קורה 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 הבעיות של של ה blocking I/O בקטע קריטי כאשר מדובר בתהליכונים הממומשים ברמת הגרעין במערכת עם מעבד אחד.

### **שאלה 6 (5%)**

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

### **שאלה 7 (5%)**

הוכיחו כי בפתרון של Peterson תהליכים אינם ממתינים 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 נקודות

מספר השאלות: 4

מועד אחרון להגשה: 09.05.2013

סמסטר: 2013ב

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

**החלק המעשי (80%)**

בחלק זה של המטלה נממש `command interpreter` עם פונקציונליות מוגבלת.

מטרת התרגיל: תהליכים ותקשורת בין התהליכים.

**רקע**

(1) סיפקנו את הקובץ `shell.c` אותו תירשו לשנות ולהרחיב. קמפלו והריצו את התוכנית.

(2) [http://www.gnu.org/software/libc/manual/html\\_node/Execu](http://www.gnu.org/software/libc/manual/html_node/Execu)

[http://www.gnu.org/software/libc/manual/html\\_node/Execu](http://www.gnu.org/software/libc/manual/html_node/Execu) `ting-a-File.html` הסבר עם פונקציות ממשפחת `exec` (אפשר להשתמש בפונקציה

לבחירתכם). כמו כן סיפקנו את הקובץ `exec.c` שאפשר לקמפל ולהריץ.

(3) פרקים 24.7.2, 24.7.3 מ

[http://www.gnu.org/software/libc/manual/html\\_mono/libc](http://www.gnu.org/software/libc/manual/html_mono/libc).

[http://www.gnu.org/software/libc/manual/html\\_mono/libc](http://www.gnu.org/software/libc/manual/html_mono/libc) עם הסבר על פונקציות `signal`, `sigemptyset`, `sigfillset`,

`sigaddset`, `sigprocmask`, `sigsuspend`. סיפקנו קובץ `suspend.c`.

תקמפלו תריצו והבינו.

(4) פרק

[http://www.gnu.org/software/libc/manual/html\\_mono/libc](http://www.gnu.org/software/libc/manual/html_mono/libc).

[http://www.gnu.org/software/libc/manual/html\\_mono/libc](http://www.gnu.org/software/libc/manual/html_mono/libc) עם הסבר על פונקציות `pipe` לתקשורת בין התהליכים.

סיפקנו קובץ `pipe.c`. תקמפלו תריצו והבינו.

מ

13.12

(5) פרק

[http://www.gnu.org/software/libc/manual/html\\_mono/libc](http://www.gnu.org/software/libc/manual/html_mono/libc).

[html](#) עם הסבר על פונקציות dup. תקמפלו ותריצו את dup.c שספקנו כדי לראות שימוש ב dup.c. dup מהווה וריאציה של pipe.c מהסעיף הקודם ומדגימה כיצד ניתן ליצור ערוץ תקשורת בין שני תהליכים בצורה שהיא שקופה לתהליכים עצמם.

מ

14.1

(6) פרק

[http://www.gnu.org/software/libc/manual/html\\_mono/libc](http://www.gnu.org/software/libc/manual/html_mono/libc).

[html](#) עם הסבר על הפונקציה chdir.

(7) כמו כן יש להיזכר בפונקציות wait, fork.

כמו ניתן לקבל מידע על הפונקציות הנ"ל מה man של LINUX.

## תיאור המשימה

- עליכם לממש `command interpreter` בשם `smash` (`small shell`) שיהיה מסוגל:
- (א) להריץ תוכניות ברקע ובזמן אמת (`background` ו `foreground`).
  - (ב) לאפשר שרשור של 2 פקודות.
  - (ג) לתמוך בפקודות פנימיות `exit` ו `cd`.

### הרצת תוכניות ברקע ובזמן אמת

הרצת תוכנית (תוכניות) בזמן אמת (`foreground`) גורמת ל `command interpreter` להמתין עד סיום התוכנית (תוכניות). למשל

```
smash:> ls
```

```
smash:> ps | wc -l
```

הן דוגמאות להרצת תוכניות בזמן אמת.

הרצת תוכנית (תוכניות) ברקע (`background`) לא גורמת ל `command interpreter` להמתין עד לסיום התוכנית (התוכניות) שרצות ברקע. הרצת תוכניות ברקע תתבצע ע"י הוספת "&" בסוף שורת הפקודה. למשל:

```
smash:> find /home -name Makefile -print &
```

```
smash:> chown -R root:root /tmp &
```

### שרשור של פקודות

`smash` יאפשר שרשור של עד לשני פקודות בשורת פקודה אחת. השרשור מתבצע ע"י סימן "|" (pipeline) בין הפקודות. משמעות השרשור היא שפלט של הפקודה הראשונה מהווה קלט לפקודה השנייה. כך למשל הרצת

```
smash:> cat /etc/passwd | wc -l
```

גורמת לספירת כמות השורות בקובץ `/etc/passwd`. הפקודה `"cat /etc/passwd"` מדפיסה את תוכן הקובץ `/etc/passwd` ל `stdout`. באמצעות ה "|" אפשר "לומר" ל `smash` להפנות את הפלט של `cat` לתוכנית `wc`. והתוצאה שהיא כמות השורות בקובץ תודפס על הצג.

## פקודות פנימיות

smash יתמוך בשתי פקודות פנימיות:

- (א) exit – בעקבות הקשת הפקודה יסיים smash את פעולתו.
- (ב) cd - בעקבות הקריאה לפקודה זו ישנה smash את ספרית העבודה הנוכחית שלו.

## הנחיות נוספות

(1) בעת הרצת תוכנית (תוכניות) בזמן אמת ישהה smash את עצמו עד לסיום התוכנית (התוכניות) באמצעות sigsuspend כארש הסיגנל היחיד שיועבר ל smash יהיה SIGCHLD. smash ישהה את עצמו עד לסיום התוכנית או עד לסיום התוכנית השנייה במקרה של שרשור פקודות.

(2) יש להקפיד "לאסוף" את הבנים עם סיומם באמצעות הפונקציה wait. לשם כך שנו את הדיספוזיציה (disposition) של סיגנל SIGCHLD ואספו את הבנים שהסתיימו ב signal handler routine.

(3) לשם פשטות ה parsing של שורת הפקודה הסימן "&" יהיה מופרד ברווח. כלומר הפקודה

```
smash:> emacs &
```

תרוץ ברקע ואילו

```
smash:> emacs
```

תרוץ בזמן אמת.

(4) שרשור של הפקודות יתבצע באמצעות הפונקציות pipe ו dup. נדגים כיצד יתבצע הדבר במקרה של "smash:> ps | wc -l":

תהליך האב (שהוא smash) יקרא לפונקציה pipe ויבצע fork בשביל wc. המצב שיווצר יהיה כמתואר בציור 1 (ראה בהמשך). לפני שעושים exec ל wc בתהליך הבן, יש לסגור את קצה ה pipe המיותר ולבצע "חיבור" של הקצה השני ל stdin (דוגמא כיצד עושים "חיבור" של קצה Pipe ל stdin מופיעה בקובץ dup.c). לאחר שבוצעו הפעולות הנ"ל תמונת המצב תהיה כמו בציור 2 (ראה בהמשך).

כעת יבצע ה smash עוד fork בשביל ps. תמונת המצב מתוארת בציור 3 (ראה בהמשך). לפני שעושים exec ל ps בתהליך הבן יש לסגור את קצה ה pipe המיותר ולבצע "חיבור" של הקצה השני ל stdout. כעת יהיו שני תהליכים ps ו wc מחוברים בצינור. כל מה שנותר זהו

לסגור את קצוות הצינור אצל תהליך האב (smash). לאחר שבוצעו הפעולות הנ"ל תמונת המצב תהיה כמתואר בציור 4 (ראה בהמשך).

(5) בעקבות פקודת exit יקרא ה smash לפונקציה exit(0). שאלה: מה יקרה לתהליכי בנים שרצו ברקע? מי "יאסוף" אותם?

(6) הפקודה הפנימית cd תקבל פרמטר אחד שהוא שם ספריית העבודה החדשה.

### טיפול בשגיאות

smash צריכה לתת הודעות שגיאה על:

(1) כשלון של קריאות מערכת או פונקציות שמכילות קירות מערכת. במקרה של שגיאות פאטאליות יש לצאת עם סטטוס 1 (ע"י exit(1)).

(2) שגיאות סינטקטיות (למשל במקרה של פקודה "ps | wc | wc -l" ישנם יותר מ 2 פקודות משורשרות). במקרה של שגיאות סינטקטיות תודיע ה smash הודעה:

```
Error : syntax error
```

### הגשה

יש להגיש כל קבצי הקוד ו Makefile המייצר קובץ הרצה smash. 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 Knoppix:

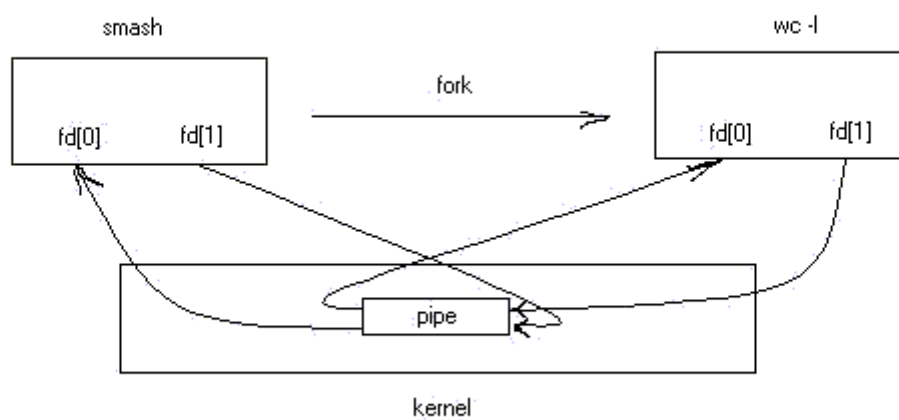
```
zip exYZ.zip <ExYZ files>
```

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

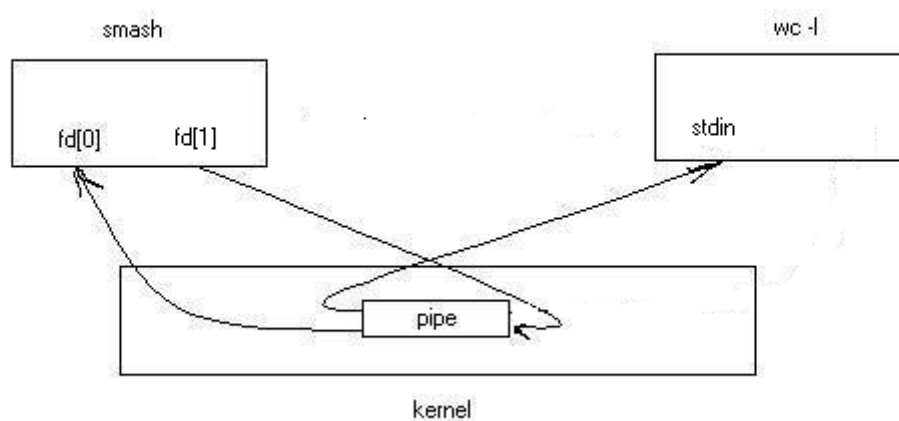
### פתרון ביה"ס

קיבלתם את קובץ smash כפי שמומש על ידינו.

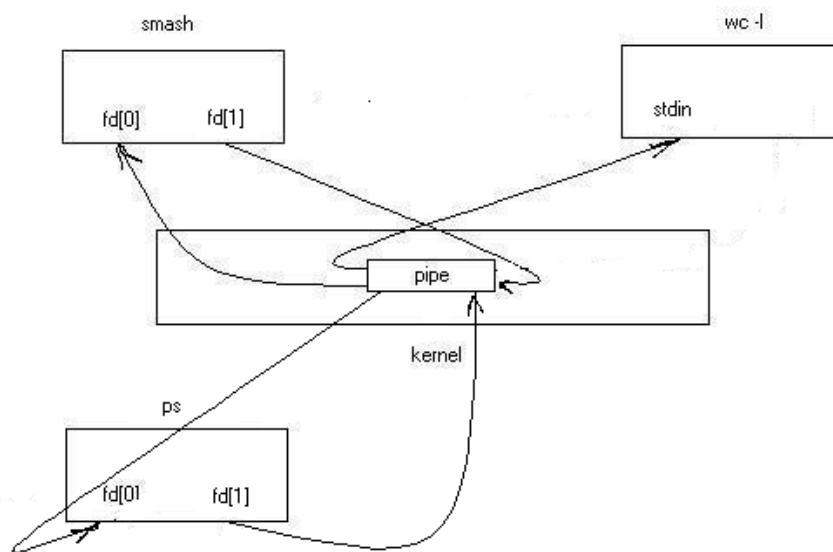
## ציור 1



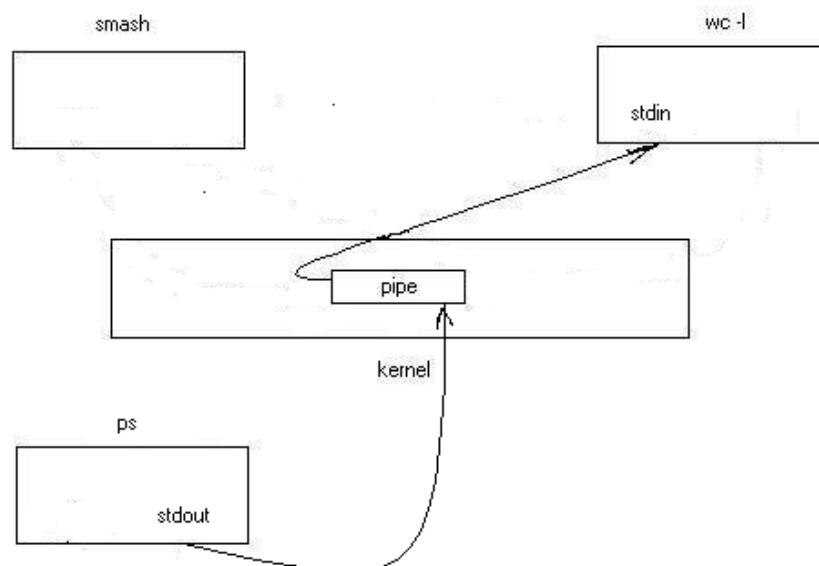
## ציור 2



### ציור 3



### ציור 4



### **החלק העיוני (20%)**

#### **שאלה 1 – (5%)**

מהי טבלת דפים מהופכת. כיצד מתמודדים עם מיפוי כתובת וירטואלית לכתובת פיזית באמצעותה? תארו איזה תמיכת החומרה נדרשת בתרגום ומדוע היא חיונית.

#### **שאלה 2 – (5%)**

מהי תופעת סחרור (threshing) ומה השפעותיה על תפקוד המערכת?

#### **שאלה 3 – (5%)**

תארו כיצד מטפלת מערכת ההפעלה בפסיקת דף.

#### **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 נקודות

מספר השאלות: 4

מועד אחרון להגשה: 06.06.2013

סמסטר: 2013ב

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

החלק המעשי (80%)

## כללי

בחלק זה של המטלה נכתוב תוכנית גיבוי פשוטה מאוד.

## מטרה

הכרת מערכת קבצים file types.

## רקע

- (1) יש לקרוא פרק 4.3.4 מספר "Modern operating systems" של A. Tanenbaum להבנה מהו (soft/symbolic link).
- (2) פרקים מ [http://www.gnu.org/software/libc/manual/html\\_mono](http://www.gnu.org/software/libc/manual/html_mono) המכסים את קריאות המערכת stat ו lstat. שימו לב להבדל בניהם. כמו כן בין הקבצים שסופקו יחד עם פתרון בה"ס נצמא קובץ filetype.c. באפשרותכם לקמפל ולהריץ את הדוגמא.
- (3) פרקים 4.2 מ [http://www.gnu.org/software/libc/manual/html\\_mono](http://www.gnu.org/software/libc/manual/html_mono) המכסים את mkdir, opendir, readdir, closedir.
- (4) פרקים מ [http://www.gnu.org/software/libc/manual/html\\_mono](http://www.gnu.org/software/libc/manual/html_mono) המספרים על 2 פונקציות symlink ו readlink. ראשונה יוצרת קישר (soft/symbolic link) והשנייה מאפשרת לקרוא את התוכן (ראה סעיף 1 דלעיל). מומלץ כמו כן לקרוא man על פקודה ln ליצירת קישורים

מ command line של shell. פקודה זו יוצרת גם קישורים קשיחים (hard links) על כן יש להיזהר אתה. השתמשו בה רק עם אופציה s— (ליצירת קישורים soft links).

(5) פרקים מ [http://www.gnu.org/software/libc/manual/html\\_mono](http://www.gnu.org/software/libc/manual/html_mono) המכסות פונקציות הבאות umask, chmod chown, lchown, utime. פונקציות הללו (פרט ל umask) נועדו לשינוי תכונות קובץ. תצטרכו אותם בעת שחזור מידע על הקבצים (ראה סעיף 5 של "הנחיות נוספות" מטה). כמו כן בין הקבצים שסופקו יחד עם פתרון בה"ס נצמא קובץ changemod.c המדגים שימוש ב umask ו chmod.

כמו ניתן לקבל מידע על הפונקציות הנ"ל מה man של LINUX.

### תיאור המשימה

עליכם לכתוב לכתוב תוכנית בשם mkbkp:

```
mkbkp <-c|-x> <backup_file> [<backupid_file | backupid_directory>]
```

כאשר מופיע דגל -c התוכנית מגבה קובץ backup\_file או לחילופין קבצים שנמצאים בתת-העץ של backupid\_directory אל הקובץ backup\_file. ניתן להניח כי אין מעגלים כתוצאה מקישורים (links).

כאשר מופיע דגל -x התוכנית משחזרת קבצים מקובץ backup\_file. אם מצוין דגל זה אין צורך בפרמטר [<backupid\_file | backupid\_directory>].

### הנחיות נוספות

- אם מגבים קובץ יחיד, אזי בעת הגיבוי לקובץ backup\_file יועתק הקובץ backupid\_file ומידע אודותיו.
- אם מגבים את ספרייה backupid\_directory, אזי לקובץ backup\_file יועתקו כל הקבצים אשר מכילה הספרייה backupid\_directory. עליכם לחשוב על מבנה עבור קובץ backup\_file שיאפשר שיחזור מלא של ספרייה backupid\_directory בעת השחזור (mkbkp עם אופציה -x). למשל, מבנה אפשרי לקובץ backup\_file יכול להכיל רשומות באורך קבוע לפני כל קובץ שיועתק ל backup\_file. כל רשומה כזו תכיל מידע שדרוש לשחזור נתונים מקוריים של הקובץ המגובה.

- הספרייה `backupid_directory` יכולה להכיל תתי-ספריות, קבצים רגילים (ordinary files) וקישורים (soft/symbolic links) בלבד. כמו כן הקובץ `file_backupid` יכול להיות קובץ רגיל או קישור בלבד. ניתן להניח כי אין מעגלים כתוצאה מהקישורים.
- שימו לב, כי השחזור צריך להתבצע יחסית לספריית העבודה הנוכחית (ז"א אין לרשום נתיב מלא לרשימת המידע ב `backup_file` אלא נתיב יחסי בלבד).
- המידע שישוחזר על הקבצים (ספריות וקישורים) צריך להכיל: `uid, gid, permissions` ואת ה `modification time` למקרה של ספריות וקבצים רגילים.
- אם ברצונכם לראות מה רושם פתרון ביה"ס בקובץ הגיבוי אפשר להריץ את התוכנית שלנו ולעיין בתוכן של קובץ הגיבוי הנוצר בעזרת מעבד תמלילים כלשהו (pico, vi וכו').

### **טיפול בשגיאות**

תוכנית הגיבוי צריכה לתת הודעות שגיאה על:

- 1) ניסיון לגבות קבצים שאינם קובץ רגיל, ספרייה או קישור `soft link`.
- 2) במקרה של `system calls` אשר לא הצליחו.
- 3) אם מתגלה בעת השחזור שקובץ/ספרייה/קישור המשוחזר כבר קיים בנתיב הנוכחי.

יש לתת הודעות שגיאה אינפורמטיביות.

### הגשה

יש להגיש כל קבצי הקוד וקובץ Makefile שמייצר קובץ הרצה בשם mkbkp. 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 Knoppix:

<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 התוכנית mkbkp כפי שמומשה על ידינו.

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

**chmod +x mkbkp**

החלק העיוני (20%)

שאלה 1 (5%)

מידת היעילות של מערכת קבצים מושפעת במידה רבה ע"י hit rate של ה buffer cache ( hit rate ) זהו יחס של כמות הבלוקים אשר נמצאו נוכחים ב buffer cache לכמות הבלוקים שנתבקשו כלל). הבאת בלוק מ buffer cache לוקחת 1 msec. הבאת בלוק במקרה של קריאה מדיסק לוקחת 40 msec. יהי h ה hit rate של buffer cache. מהו הזמן הממוצע הנדרש להבאת בלוק?

שאלה 2 (5%)

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

$P1 = \text{Even Parity of } b1, b2, b4$

$P2 = \text{Even Parity of } b1, b3, b4$

$P3 = \text{Even Parity of } b2, b3, b4$

לדוגמא: המינג קוד של מילה בת 4 סיביות (משמאל לימין – מ least significant bit ל most significant bit) יהיה 110110.

### שאלה 3 (5%)

לפי מדיניות חדשה של תזמון זרוע הדיסק, הבקשות מוחזקות בתור לפי סדר הגעתן והראשונה שמטופלת היא הבקשה שהגיע אחרונה. מדיניות זו נקראת LIFO (last in first out).

(א) מהו היתרון של המדיניות הזאת?

(ב) מהו החיסרון של המדיניות הזאת?

### שאלה 4 (5%)

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

### 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).