

מדריך ל-LINUX

מאת: מוניק אשכנזי

עריכה: הלן לישר

עובד מחדש ע"י ידידיה ישראל וצוות התמיכה (נוב' 2000)

נערך ועודכן ע"י שמואל מקלר (נובמבר 2002)

המחלקה למתמטיקה והמחלקה למדעי המחשב

הפקולטה למדעים מדויקים

אוניברסיטת בר-אילן

© כל הזכויות שמורות. אין להעתיק, לצלם, להפיץ, לשכתב בכל צורה ודרך, אלקטרונית ואחרת בלא רשות

כתובה מאת בעלי זכויות היוצרים.

הכרת תודה

תודתנו נתונה למרכז המחשבים – אוניברסיטת בר-אילן, לד"ר דרורה כהן ולגב' מוניק אשכנזי, על שאפשרו לנו להיעזר בחוברת "מדריך ל UNIX".

כשם שאי אפשר לתבואה ללא תבן כך אי-אפשר לחוברת ללא שגיאות. כל הערה תתקבל בברכה ע"י צוות התמיכה.

הנחיות לקורא

- מדריך זה מיועד למשתמש המתחיל ב-UNIX ולמשתמש המתקדם יותר. למשתמש המתקדם מיועד "נספח ב - תוספות למתקדמים".
- רוב הנושאים והפקודות אופייניים לכל מערכת UNIX טבעי הדבר שבמערכות UNIX אחרות יהיו הבדלים, חלקם מזעריים וחלקם משמעותיים.
- במהלך המדריך נקטנו במספר כללים:
 1. הפקודות מופיעות כך: **Command**.
 2. בהגדרת מפרטי הפקודות השתמשנו בסימנים < > כדי לתחום את האופציות של הפקודות. סימנים אלו אינם חלק מתחביר הפקודות.
 3. מקשי פעולה, מקשים שהקשה עליהם גורמת לפעולה מסוימת מסומנים בהדגשה.

Smiley (-: חייכן (-:

כשתפגוש בחבורת בסימן החייכן (-: או (-: עליך למתוח את שפתיך ולפרוץ בצחוק מתפקיד החייכנים להפיח רוח חיים בטקסט מת. כדי להבין מדוע נבחר דווקא רצף תוים זה לציין בדיחות, סובב את הדף ב-90 מעלות והתבונן עליו שוב.

ההוראות בחוברת זו כתובות בלשון זכר ומיועדות כמובן גם לנשים.

תוכן

6	ברוכים הבאים לעולם ה - UNIX
6	הקדמה
6	1. התחלת העבודה
6	שם משתמש וסיסמא
6	התחברות למערכת
7	סביבת חלונות
7	ה - prompt
7	יציאה מהמערכת / kill logout
7	שינוי סיסמא passwd
9	2. היכרות עם פקודות UNIX
9	עצירת פקודה
9	תחביר פקודות UNIX
10	3. תיעוד מקוון (help on-line)
11	4. מערכת הקבצים
12	ארגון מערכת הקבצים
13	פקודות לטיפול בקבצים
13	הצגת רשימת קבצים ls
13	הצגת תוכן הקובץ
13	הפקודה cat
13	הפקודה more
14	העתקת קבצים cp (copy)
14	שינוי שם קובץ - הזזת קובץ mv (move)
15	מחיקת קובץ rm (remove)
15	חיפוש קבצים find
15	הדפסת קובץ (lp) וניהול הדפסות (lpstat, cancel)
16	הצגת המדריך הנוכחי pwd (present working directory)
16	שינוי מדריך cd (change directory)
16	יצירת מדריך mkdir (make directory)
16	מחיקת מדריך rmdir (remove directory)
16	תוים מיוחדים בשמות קבצים
17	קבצים דחוסים
17	הגנה על מערכת הקבצים
19	5. העורך - vi
19	הפעלת העורך vi
19	פקודות vi
19	הוספת טקסט
20	איחזור טעויות (Retrieve) חרטה (Undo) וחזרה (Redo)
20	החלפת טקסט
20	תנועה וחיפוש
21	העתקה והזזת טקסט
21	פקודות נוספות
21	סיום העבודה
22	6. דואר אלקטרוני
22	כתובות
22	mail
22	שליחת דואר
22	קריאת דואר
22	יציאה מהדואר
23	elm
23	קריאת דואר

23	pine
23	יתרונות pine
23	צורת העבודה עם pine
24	ממשקים נוספים
25	7. שירותי ה-Shell
25	הכוונת קלט-פלט
25	צינורות (pipes)
26	שמות קיצור לפקודות (Aliases)
26	אחסון וחזרה על פקודות (history)
27	8. שימוש בקומפיילרים - "מהדרים"
27	שפת אסמבלי
27	שפת C
27	שפת C++
27	אופציות של מהדרים
28	mainx
28	הודעות שגיאה של המהדר
28	הרצת תוכנית
28	dbx Debugger
29	פקודות dbx
29	gdb
29	ספריות
29	ספריות ar
30	10. עבודה ברשת (Networking)
30	פקודות הפעלה מרחוק (Remote) בין מחשבי UNIX
30	הפקודה rsh
30	הפקודה rlogin
30	הפקודה rcp
31	הפקודה telnet
31	הפקודה ftp
32	נספח א' X-WINDOWS מערכת חלונות
32	מרכיבים של מערכת חלונות:
32	שרת X
32	לקוח X
32	פרוטוקול X
32	מסוף X
32	מנהל חלונות
33	שימוש בעכבר
33	חלון פעיל
33	העתקת טקסט
33	הזזת חלון
33	שינוי גודל חלון
33	איקון ICON
34	שימוש בתפריטים
34	דפדוף בחלון
34	פתיחה וסגירה של חלונות
34	שינוי סביבת עבודה של X
36	נספח ב' - תוספות למתקדמים
36	מסלול חיפוש (Path Search):
36	משתנים סביבתיים (Variables Environment)
37	קבצים מיוחדים
37	הקובץ login
37	הקובץ cshrc
37	קבצים נוספים
38	עבודות (JOBS)

38.....	תהליך (process)
38.....	עבודה (JOB)
38.....	הרצת עבודות ב - background
39.....	דו"ח על העבודות
39.....	שימוש בדיסקטים של DOS תחת סביבת UNIX
40.....	הפקודה tar
40.....	Shells בהרחבה
41.....	C-Shell
41.....	הפקודה echo
42.....	הגדרת רשימה - "מערך" :
42.....	פעולות אריתמטיות
42.....	משפטי תנאי - if
43.....	פקודת while
43.....	פקודת foreach
43.....	Script Shell
45.....	Tcsh
45.....	השלמת שמות פקודות-קבצים
45.....	תיקון טעויות כתיב (spelling)
45.....	דפדוף במאגר הפקודות ועריכתו
46.....	טבלה 1. השוואה בין Shells ב- UNIX
47.....	נספח ג - רשימת פקודות
50.....	עשה ואל תעשה (DON'TS & DO'S)

ברוכים הבאים לעולם ה - UNIX

הקדמה

UNIX היא מערכת הפעלה רב-תהליך ורב-משתמש. באופן כללי - מערכת הפעלה היא תוכנית "המנהלת" את המחשב והציוד ההקפי, כגון מדפסות, ומשתמשת ממשק (interface) בינה המשתמש לבין המחשב. הרוצים הגדרה מדויקת יותר - מוזמנים להשתתף בקורס "מערכות הפעלה".

"רב-תהליך" פירושו שהמחשב יכול לבצע מספר דברים בו-זמנית ו-"רב-משתמש" פירושו שהמחשב יכול לטפל במספר משתמשים בו-זמנית. למעשה זה רק כאילו בו-זמנית - כפי שתלמד בקורס הנ"ל.

למעשה UNIX מדברת אתך דרך תכנית הנקראת shell. ניתן "לעצב" את ה-shell כך שתהווה סביבת עבודה אישית יותר - ניתן לתכנת את ה-shell ואף להחליפו בתכנית אחרת. רבות הן האפשרויות - וכמובן שלא נוכל להסביר את כולם. נעמוד על הפעולות הבסיסיות ביותר -

ונזכיר במילה או שתיים את קיומן של פעולות נוספות. הדרך הטובה ביותר ללמוד לשחות בימה של UNIX היא - לקפוץ פנימה ולנסות לשחות בעזרת ההדרכה שתינתן כאן. בהצלחה.

1. התחלת העבודה

לאחר הקדמה קצרה זו - אנו מוכנים לגשת לעבודה. מה אמרת? לא קראת את ההקדמה? שמע - אנחנו מבינים שעליך להגיש מחר את התרגיל הראשון ועוד לא התחלת! שאתה לבטח חושב לעצמך בשביל מה בכלל אני צריך את ההקדמה. בכל זאת - כדאי מאד להקדיש כמה דקות לקריאת ההקדמה - גם אם לא תבין כל מה שכתוב שם. בהמשך עבודתך ב-UNIX תגלה שאותן שתיים שלוש שורות עזרו לך להתחיל את עבודתך ב-UNIX ברגל ימין. בנוסף לכך - כבר הקדשת יותר זמן לקריאת הערה זו מאשר היית מקדיש לקריאת ההקדמה! (-):

שם משתמש וסיסמא

על מנת להתחבר למערכת - אתה זקוק לשם משתמש (login) ולסיסמא (password). שם המשתמש מונפק ע"י המחלקה ויפורסם בתחילת השנה. הסיסמא אישית ואין למסרה לאחרים. הסיסמא הראשונה תונפק ע"י המחלקה ותידרש להחליפה בפעם הראשונה שתתחבר למחשב וכמו כן מידי שלושה חדשים, ראה שינוי סיסמא בהמשך.

התחברות למערכת

תחילה ברר האם המסוף כבר דלוק ע"י לחיצה קלה על מקש מבין מקשי המקלדת. אם עדין לא הופיע כלום על המסך - הדלק אותו!

אם המסוף מבקש שם של מחשב הקש את שם המחשב אליו אתה מורשה להכנס או בחר אותו מהרשימה שהוצגה לפניך. בד"כ זה יהיה planet ולחץ על מקש ה-Enter/Return.

עם קבלת המילה login הקש את שם המשתמש שלך ולחץ על מקש ה-Enter/Return.

כעת הכנס את הסיסמא שלך ליד המלה: password ולחץ על מקש ה-Enter/Return.

הסיסמא לא תופיע בעת הקשתה על המסך מטעמי אבטחת מידע.

לאחר סיום מוצלח של התחברות למערכת - יתכן ותופענה על המסך הודעות למיניהן שעשויות להיות חשובות לך, זמנים בהם המחשב מושבת - מידע על תוכניות חדשות וכו'.

בעגה המקצועית נקראים הודעות אלו: message of the day (MOTD) קרא אותם.

כמו כן - תופיע הודעה על מכתבים שנשלחו לך ע"י משתמשים אחרים ונמצאים בתיבת הדואר שלך, ראה פרק 6. דואר אלקטרוני.

סביבת חלונות

בשלב זה סביר להניח שנכנסת לסביבת חלונות: על המסך מופיע לפחות חלון אחד - שטח מלבני בתוכו ניתן להקיש תוים. סביבת חלונות היא הסביבה המקובלת לעבודה עם UNIX - אך מהווה נושא בפני עצמו. נספח א' X-WINDOWS "מערכת חלונות" מסביר מהי מערכת חלונות וכיצד עובדים בה. לצורך לימוד UNIX תוכל לדלג בשלב זה על לימוד מערכת חלונות ולהתרכז בחלון אחד. לשם כך - הזז את העכבר כך שהמצביע על המסך ינוע לתוך החלון שהתקבל. הקש (לחץ ושחרר) על הכפתור השמאלי של העכבר. כעת החלון פעיל. מעתה ואילך תוכל לתרגל את הנאמר בחוברת בחלון זה.

ה - prompt

מחרוזת התוים בעזרתה UNIX מסמנת לך שהיא מוכנה ומזומנת לקבל ממך פקודות היא ה-prompt בהמשך תוכל ללמוד איך לשנותה ובכך לאפשר לעצמך סביבת עבודה אישית יותר. בינתיים ניתן לך prompt די נחמד הכולל את שם ה-login שלך, שם המחשב עליו אתה נמצא כעת, מספר הפקודה ונקודתיים. דוגמא:

levy david planet 74:

יציאה מהמערכת kill / logout

בכדי לסיים את העבודה במערכת מומלץ להקיש: `kill -1 -1`
שימו לב: פקודה זו מסיימת כל תהליך אשר אתם מריצים על המערכת. כגון חישובים ארוכים או תהליכים שבאחריותכם (לדוגמא עבור מנהלי המערכת (root) תור ההדפסות).
 כדי לצאת מבלי לסיים את כלל התהליכים יש להקיש `exit` או `logout`
 וללחוץ על מקש ה- Enter/Return.
 הערה: מעתה ואילך לא נציין את הצורך ללחוץ על מקש ה- Enter/Return לאחר הקשת פקודה מתוך הנחה שדבר זה ברור.

במערכת חלונות יש לצאת מהמערכת באמצעות פריט WM Exit או Logout בתפריט הראשי. את התפריט אפשר לקבל בלחיצה על אחד ממקשי העכבר כאשר המצביע לא נמצא בחלון מסוים. חשוב לציין שחוסר חלונות על המסך אינו אומר בהכרח שיצאת החוצה. ישנה חשיבות רבה לכך שתצא מן המערכת לאחר סיום עבודתך שאם לא כן יוכלו אחרים להשתמש בשם המשתמש האישי שלך ובכך לגרום נזק לך ולקבצ'ך. בנוסף לכך קיימת סכנה שמנהלי המערכת ימנעו את הגישה למשתמש המפקיר את שם המשתמש שברשותו לשימוש הכלל - דבר שיכול לגרום לאי נעימות.

זכור: אתה אחראי לנעשה בקוד המשתמש שלך.

שינוי סיסמא passwd

הפקודה לשינוי סיסמא היא: `passwd`
 לאחר הקשתה תתבקש להקיש את הסיסמא הנוכחית, לאחר מכן את הסיסמא החדשה, וכדי לאשר אותה תתבקש להקיש אותה שוב. בפעם הראשונה בה תתחבר למחשב תתבקש אוטומטית להחליף את סיסמאתך פעל בהתאם להוראות שתקבל.
 חשוב לציין שלאחר שינוי הסיסמא דרוש זמן מה לעדכן את כל המחשבים בסיסמתך החדשה. לכן אל תתמה אם לאחר שינוי הסיסמא - סיסמתך הישנה עדיין תקפה באי-אלו מחשבים.
 כדי להבטיח את סיבוכיות בסיסמא נקבעו הכללים הבאים:
 1. אורך הסיסמא יהיה 7-8 תוים.
 2. סיסמא תכלול תוים לפחות משלוש קבוצות תוים, ראה בהמשך הגדרת קבוצות.

3. לא יהיה בסיסמא רצף ארוך משלושה תוים מאותה קבוצת תוים.
 4. לא ניתן לעשות שימוש במילים מהשפה האנגלית או בשמכם.

הוגדרו ארבע קבוצות תוים:

א. האותיות הלטיניות הגדולות: A-Z.

ב. האותיות הלטיניות הקטנות: a-z.

ג. ספרות: 0 - 1

ד. שאר התוים שאינם אלפה-נומרים כגון:

() [] { } , . ; ! @ # % - + * = / \ \$ & < > | ?

זהירות: קרא בענין את הקטע האחרון - אי הצלחה לשנות את הסיסמא בפעם הראשונה יכול לגרום להפחתת נקודות ממבחן ה-IQ שלך. :-)

2. היכרות עם פקודות UNIX

מערכת UNIX היא אינטראקטיבית, לאחר כתיבת פקודה ולחיצה על Enter/Return יש המתנה לסיום בצוע הפקודה עד להופעת ה-prompt - הבא. נסה למשל את הפקודה `finger` המציגה פרטים על המשתמשים במערכת.

עצירת פקודה

לעתים קורה שפקודה שהקשת מתבצעת זמן ממושך מאוד. בזמן ביצועה אינך מקבל את ה-prompt ולכן אינך יכול להקיש פקודה נוספת. ניתן לצאת ממצב זה ע"י ביטול הפקודה. הקש על `c+ctrl`, לחיצה על מקש `ctrl` ובו זמנית לחיצה על תו `c`. אם זה לא עזר נסה `ctrl+z` להשהיית הפקודה וביטולה ע"י פקודת `kill`, ראה "נספח ב' - תוספת למתקדמים" בפרק "עבודות (JOBS)".

תחביר פקודות UNIX

פקודות UNIX מורכבות מ: פרמטרים <אופציות> שם-פקודה מרבית האופציות מתחילות בתו "-" . דוגמא:

`finger-l ben`

דוגמא הנ"ל האופציה 1 מורה לפקודה `finger` להציג מידע מפורט (long) על המשתמש `ben`. אם שם המשתמש `ben` לא קיים יוצגו סימני שאלה ??? . כאשר מציינים יותר מאופציה אחת ניתן בד"כ לצרף אותן יחד. למשל - את הפקודה `finger -l -p ben` ניתן לכתוב גם כך:

`finger -lp ben`

לא נפרט כאן את משמעות האופציה `p`.

ניתן לצרף מספר פקודות בשורה אחת על ידי הפרדתן באמצעות ";" ; דוגמא:

`finger ; date`

הפקודה `finger` מתבצעת תחילה ומציגה פרטים על המשתמשים המחוברים כרגע למערכת. לאחר בצועה מתבצעת הפקודה `date` המציגה את התאריך והשעה. ניתן לפצל פקודה ביותר משורה אחת ע"י כתיבת התו (`\` backslash) בסוף השורה ולחיצה על Enter/Return, שימושי בעיקר לפקודות ארוכות. דוגמא:

`finger -l \ ben`

ה-prompt לשורה המפוצלת ישתנה ל- "?" או ל- " " כדי לסמן שהקשת הפקודה לא הסתיימה.

3. תיעוד מקוון (help on-line)

למערכת ההפעלה UNIX יש תיעוד מקוון (on-line) לרוב הפקודות במערכת. התיעוד של כל פקודה מורכב ממספר סעיפים: בחלק הראשון ניתן שם הפקודה (NAME) ומשמעותה; בחלק השני מתואר תחביר הפקודה (SYNOPSIS).

בחלק הבא קיים תיאור של הפקודה והאופציות השונות שהיא מאפשרת (DESCRIPTION); בהמשך קיימת רשימה של קבצים בהם משתמשת הפקודה (FILES) ורשימת תוכניות רלוונטיות הקשורות לנושא (ALSO SEE). ייתכנו חלקים נוספים כגון: טיפול בבעיות (DIAGNOSTICS) תיאור שגיאות למיניהן (BUGS) וכו'.

<שם-פקודה>

העיון בתיעוד של פקודה מסויימת נעשה ע"י:
man לדוגמא:

man passwd

ניתן לקבל רשימה של פקודות על פי מילת מפתח ע"י שמוש באופציה -k

<מילת-מפתח> man -k

<מילת-מפתח> apropos

או באמצעות

man man

פרטים נוספים על פקודת man ניתן לקבל ע"י:

הפקודה: <שם-פקודה> whatis תתן הסבר מתומצת על הפקודה. זהו בעצם סעיף ה-SYNOPSIS ב-manual של הפקודה.

קיימת תוכנית נוספת לקבלת עזרה בשם news באמצעותה תקבל מידע עדכני על שינויים ותוספות מקומיות וכדומה, פשוט כתוב news לקבלת הוראות שמוש. יש גם עזרה ראשונה ע"י צוות התמיכה במחלקה. הקש news support למידע נוסף.

אם ה-shell שלך הוא tcsh, כפי שהוא בד"כ, תוכל להקיש ESC+h לאחר שם הפקודה כדי לקבל את שני הסעיפים הראשונים SYNOPSIS-NAME של ה-"man" שלה. על מעללים נוספים של tcsh תוכל לקרוא בנספח.

חידה: אפרופו "מן" - כיצד תבחין בהבדל בין ה-man של פורים לזה של בני ישראל במדבר ? :-)

4. מערכת הקבצים

קובץ הוא רצף של תווים המאוחסן במחשב - לרוב על דיסק. כל קובץ מאופיין על ידי שם לפי הכללים הבאים:

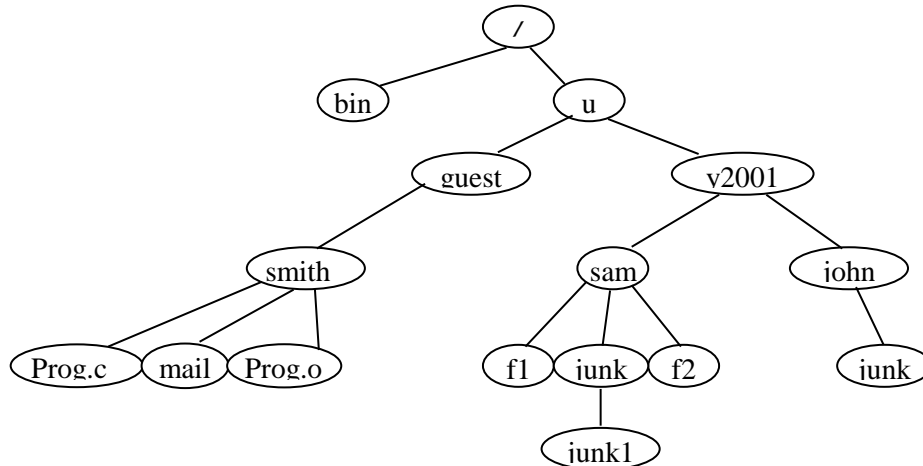
1. השם יכול להכיל תו אחד עד 255 תווים
2. כל התווים קבילים פרט ל -
3. רצוי להימנע משמוש בתווים הבאים:
 \backslash " ; [] () * & ' ^ ?
4. רצוי להמנע מלהשתמש בתווים: + - . כתו ראשון בשם קובץ.
5. מערכת UNIX מבדילה בין אותיות קטנות וגדולות - לכן למשל, שם קובץ File אינו זהה לשם קובץ file.

סיומות מקובלות עבור קבצים שונים:

Free Ascii text file	.txt
Script file – On screen log file	.scr
library	.a
relocatable object file	.o
awk script	.awk
Pascal source	.pas
Pascal include file	.i
C source	.c
perl script	.pl
C++ source	.cpp
C Header – Include - File	.h
PostScript program	.ps
Device Independent (TeX output)	.dvi
assembly source	.s
Fortran source	.f
shell script	.sh
GNU-compressed-file	.gz
tape archive image	.tar
compressed file	.Z
TeX source file	.tex
shared library	.so
lex source	.l
yacc source file	.y
log files	.log

ארגון מערכת הקבצים

מערכת הקבצים ב-UNIX מאורגנת בצורת עץ הפוך: בראשו נמצא השורש, בשם "/". ומתחתיו מסתעפים קבצים כמו ענפים ותתי ענפים, או לחילופין "אבות" ו"בנים". הקבצים שיש הסתעפות מתוכם נקראים מדריכים directories. לכל מדריך בעץ, פרט לשורש, קיים "אב" אחד ויכולים להיות לו מספר "בנים". קבצים רגילים, קבצי נתונים ותוכניות וכיו"ב שייכים למדריכים אך אין להם "בנים". ניתן להתייחס לשם קובץ ע"י ציון המסלול, path- באופן יחסי למקום הנוכחי בעץ או באופן מוחלט, כלומר- יחסית לשורש. נדגים זאת בעזרת התרשים הבא המתאר חלק ממערכת קבצים:



לכל משתמש מוגדר מדריך ראשי (directory home) שהוא המדריך בו הוא נמצא מיד לאחר ההתחברות למערכת. כאשר משתמש בשם sam מבצע login המערכת מכניסה אותו למדריך הראשי שלו שהוא נניח /u/y2001/sam/ -ה, /- מפריד בין הרמות בעץ וגם מסמן את מדריך השורש. על מנת לעיין בקובץ f1 עליו להקיש: f1 cat אך כדי לעיין בקובץ junk1 עליו להקיש:

```
cat junk/junk1
```

למעשה השם המלא של junk1 הוא: /u/y2001/sam/junk/junk1 אך כיון שהמדריך הנוכחי הוא /u/y2001/sam ניתן להסתפק במסלול על העץ יחסית אליו.

סימנים מוסכמים

המקום הנוכחי בהרכיית מערכת הקבצים	.
"אב" של המדריך הנוכחי	..
המדריך הראשי של המשתמש.	~
שורש העץ בהרכיית מערכת הקבצים	/

דוגמאות:

המשתמש sam העובד במדריך הראשי שלו יכול לעיין בקובץ junk של המשתמש john ע"י:

```
cat ../john/junk
```

אם sam שינה את המדריך הנוכחי שלו ל - /u/y2001/sam/junk/ (על שינוי מדריכים בהמשך) הוא יוכל לעיין בקובץ f1 ע"י:

```
cat ~/f1
```

מדריכי מערכת עקריים

יש מספר מדריכים הנמצאים דרך קבע ברמה אחת מתחת לשורש ומכילים קבצי מערכת:

- bin מכיל תוכניות של מערכת ההפעלה
- etc מכיל תוכניות וקבצים לניהול המערכת
- lib מכיל ספריות עבור תוכניות ושפות תכנות
- tmp מכיל קבצים זמניים אשר כל משתמש יכול ליצור
- dev מכיל קבצים מיוחדים המייצגים יחידות קלט פלט הקפיות כגון: מדפסות, מסופים, דיסקים, כונני סרטים ועוד.

פקודות לטיפול בקבצים

הצגת רשימת קבצים ls

<שם-קובץ או שם-מדריך> <אופציות> ls

פקודה המציגה על המסך את רשימת הקבצים במדריך המצוין. אם לא צוין שם מדריך אזי המערכת מציגה את רשימת הקבצים במדריך הנוכחי. ניתן גם לציין מספר מדריכים ולקבל את רשימת הקבצים בכל מדריך. כמו כן, ניתן לציין שם קובץ עם האופציות המתאימות לקבלת פרטים על הקובץ. רשימת אופציות עקריות:

- l (long) נותן מידע מפורט על כל קובץ, תאריך עדכון זכויות גישה לקובץ וכו'
- a (all) כולל ברשימת הקבצים גם קבצים ששםם מתחיל ב-'.' אלה קבצים שהמערכת משתמשת בהם, למשל login.
- t (time) ממיין את רשימת הקבצים לפי תאריך עדכון של הקבצים במקום לפי סדר א-ב.
- F מוסיף את התו ' - ' בסוף שם של מדריך. 'x' בסוף שם של קובץ ריצה.
- R מציג רשימת כל התתי מדריכים בצורה רקורסיבית עד תחתית העץ.

דוגמא:

המשתמש sam הפעיל במדריך הראשי שלו את הפקודה `ls -t -l`

המסך שהתקבל:

```
-rw-r--r--  1 sam y2001   60      Aug 07   13:24   f1
drwxr-xr-x  2 sam y2001  512      May 09   10:55   junk
-rwxr----- 1 sam y2001   46      Feb 18   15:40   f2
```

10 התווים הראשונים בשורת המידע נקראים ה-mode. התו הראשון מזהה את סוג הקובץ: 'd' עבור מדריך, '.' עבור קובץ רגיל. 9 התווים הבאים מתייחסים להרשאות קריאה, כתיבה וביצוע לקובץ, עליהן נדון בהרחבה בהמשך. המספר הבא הוא מספר ה-links לקובץ, לא נרחיב על כך את הדיבור. אחר כך רשום שם הבעלים של הקובץ, הקבוצה אליה משתייך הקובץ, גודל הקובץ בבתים, זמן עדכון אחרון של הקובץ ושם הקובץ.

הצגת תוכן הקובץ

הפקודה `cat` (הצגת קובץ `cat` -l) (less

<שם-קובץ> <אופציות> cat

מדפיסה תוכן קובץ על הפלט הסטנדרטי, המסך אך אינה מאפשרת דפדוף. הפקודה יעילה לקבצים קצרים שכל תוכנם יכול להיות מוצג על מסך אחד.

הפקודה `more`

<שם-קובץ> <אופציות> more

מדפיסה תוכן קובץ על המסך ומאפשרת דפדוף. בתחתית המסך מודפס אחוז התווים שכבר הוצגו. דפדוף בקובץ:

רווח (bar space)	דפדוף למסך הבא
b	דפדוף מסך אחורה
Enter/Return	דפדוף שורה שורה
/pattern	חפש את המחרוזת pattern.
n	מצא את המופע הבא של הביטוי (next).
h	הדפסת מסך עזרה
q	סיום עיון בקובץ

הפקודה Less:

`less <שם-קובץ> <אופציות>`

less היא תוכנית חדישה יותר. less יעצור לאחר הצגת חלון אחד ויפיע נקודתיים : בתחתית המסך.
 אוסף הפקודות העיקריות דומה לזה של more להלן אוסף פקודות 'טיולים' המיוחדים ל-less:
 d טיול קדימה. ברירת מחדל 10 שורות, down, חצי מסך.
 u כמו d אך לאחור, up חצי מסך.
 k, y לך שורה אחת אחורה.
 כמו כן ניתן לגלל קדימה אחורה במסמך באמצעות החיצים.
 לסיכום: less is more than more

העתקת קבצים cp (copy)

`cp <שם-קובץ-מטרה> <שם-קובץ-מקור> <אופציות>`

א

`cp <שם-מדריך-מטרה> <שמות-קובץ-מקור> <אופציות>`

(מדריך = ספרייה = folder)

הפקודה cp מאפשרת:

1. העתקת קובץ מקור לקובץ מטרה.

אם קובץ המטרה לא קיים - הוא יוצר אם הוא קיים - תוכנו הישן יימחק! דוגמא:

`cp f1 f2`

העתקת קובץ f1 ל-f2 במדריך הנוכחי.

2. העתקת קובץ למדריך.

הקובץ יועתק בשמו המקורי. אם קיים במדריך קובץ בעל אותו שם - תוכנו הישן יימחק! דוגמא:

`cp /file/temp ~/.`

העתקת קובץ /file/temp למדריך הראשי של המשתמש.

3. העתקת מדריך למדריך:

א. העתקת כל הקבצים ממדריך אחד למדריך שני, תתי מדריכים של המדריך הראשון לא יועתקו.

דוגמא: sam יכול להעתיק למדריך הנוכחי שלו את כל הקבצים במדריך /sam/tmp

`cp -x /sam/tmp/ .`

ב. אם מציינים את האופציה -r (recursive) מועתק המדריך הראשון וכל תתי המדריכים שלו

למדריך השני. דוגמא:

`cp -r /sam .`

המדריך sam וכל הנמצא בו מועתקים למדריך הנוכחי.

האופציה -i מבקשת אישור מהמשתמש לפני מחיקת קובץ קיים כתוצאה מפעולת ההעתקה. דוגמא:

`cp -i f1 f2`

שינוי שם קובץ - הזזת קובץ mv (move)

`mv <שם-הקובץ-החדש> <שם-קובץ-הישן> <אופציות>`

א

`mv <שם-המדריך-החדש> <שם-המדריך-הישן> <אופציות>`

הפקודה mv משמשת להזזת קובץ ממקום מסויים במערכת הקבצים למקום אחר.

כמו כן - ניתן להשתמש בפקודה mv לשינוי שם קובץ.

דוגמאות:

```
mv f1 f2
```

בדוגמא זו - שם הקובץ שונה מ- f1 ל- f2. אם הקובץ f2 היה קיים **תוכנו יימחק!**

```
mv f1 ./junk
```

בדוגמא זו - מיקום הקובץ f1 שונה מהמדריך הנוכחי למדריך junk. ניתן להשתמש באופציה `-i` - אשר מבקשת אישור מהמשתמש לפני מחיקת קובץ קיים כתוצאה מפעולת ההזזה או שינוי השם.

מחיקת קובץ rm (remove)

<שם-מדריך או שם-קובץ> <אופציות> rm

מומלץ מאד להשתמש בפקודה זו מדי פעם כדי להפטר מקבצים מיותרים - אחרת תוצר בעיה של חוסר מקום בדיסק.

דוגמא: על מנת למחוק את הקובץ junk1 הנמצא במדריך junk רשום:

```
rm junk/junk1
```

האופציה `-i` - מבקשת אישור מהמשתמש לפני מחיקת הקובץ.

האופציה `-r` - מאפשרת מחיקה **רקורסיבית** של מדריכים ותוכנם - כאשר הפרמטר של `rm` הוא מדריך.

דוגמא: מחיקת כל הקבצים והמדריכים ב- junk , תוך כדי בקשת אישור:

```
rm -ir junk
```

המחיקה מתבצעת רק לאחר אישור עבור כל קובץ ומדריך, בגלל האופציה `-i` . לאחר מחיקת קובץ לא ניתן לשחזרו בקלות לכן שונתה הפקודה במערכת שלנו כך שלמעשה תעביר את הקובץ למדריך מיוחד בשם **TrashCan** משם אפשר לשחזר את הקובץ באמצעות הפקודה:

<שם-הקובץ> unrm

מדריכי ה-**TrashCan** מנוקים ע"י המערכת כל כמה ימים לכן אין לסמוך עליהם יותר מדי.

כדי לנקות בעצמך את ה- **Trashcan** שלך השתמש בפקודה:

```
clean.trashcan
```

כדאי לבצע מחיקת מלאה עבור אותם קבצים התופסים הרבה מקום (על חשבון המשתמש שלך).

את פקודת ה-`rm` המקורית עדיין אפשר לבצע ע"י `\rm` . אל תצפה ש-`-i` ו-`-r` יעבדו ב-`rm` החדש.

הערה: השתמש בפקודה `du` או בפקודה `quota -v` כדי לדעת כמה מקום תופסים קבצריך.

ניצול (usage) גבוהה רומזים על כך שעליך להשתמש בפקודה `rm` .

חיפוש קבצים find

הפקודה `find` מאפשרת למצוא את מיקומם של קבצים במערכת הקבצים. לפקודה זו אופציות רבות , פירוט ניתן לקבל ע"י הקשת `man find` צורתה השמושית :

```
find <שם-קובץ> <-name> <מסלול-חיפוש>
```

המערכת מחפשת את הקובץ החל ממסלול החיפוש ועם מציאתו מדפיסה את המסלול (path) שלו , ז"א את מיקומו בעץ. האופציה `-print` נדרשת על מנת להציג את תוצאות החיפוש. דוגמא:

```
find . -name a.out -print
```

המערכת מחפשת את הקובץ a.out החל מהמדריך הנוכחי ". - ולאחר שמצאה אותו - מדפיסה על המסך את המסלול המלא שלו.

הדפסת קובץ (lp) וניהול הדפסות (lpstat, cancel)

<שם-קובץ> <אופציות> lp

הפקודה `lp` מאפשרת הדפסת קובץ במדפסת מקומית.

<אופציות> lpstat

הפקודה `lpstat` מציגה את ההדפסות הממתנות בתור להדפסה. לכל הדפסה מוצמד מספר בקשה.

cancel <אופציות> <מספר-בקשה>

הפקודה **cancel** מוחקת קבצים מהתור להדפסה לפי מספר הבקשה אופציה שימושית היא שם מדפסת או אולי **-d** באמצעותה קובעים לאיזה מדפסת אנו מעונינים לשלוח את ההדפסה. דוגמא:

```
lp -d lab4 targil.cpp
```

תשלח את התרגיל שלך להדפסה במדפסת lab4, הפקודה תדווח לך על מספר הבקשה. לדוגמא: request id is lab4-1234 (1 file)

תוכל לבטל את ההדפסה אם היא עדין בתור ע"י הפקודה:

```
cancel lab4-1234
```

הצגת המדריך הנוכחי pwd (present working directory)

הפקודה **pwd** מציגה את שם המדריך הנוכחי.

דוגמא: המשתמש sam המפעיל את הפקודה **pwd** יקבל: /u/y2001/sam/

שינוי מדריך cd (change directory)

הפקודה **cd** משנה את המדריך הנוכחי. **cd** ללא ציון מדריך מחזיר את המשתמש ל-מדריך הראשי שלו.

דוגמאות:

```
cd ..
```

הפקודה הנ"ל מעלה את המשתמש רמה אחת למעלה בעץ

```
cd ../john
```

הפקודה הנ"ל מאפשרת ל-sam לשנות את המדריך שהוא כרגע עובד בו ל- /u/y2001/john/

יצירת מדריך mkdir (make directory)

mkdir <שם-מדריך> <אופציות>

הפקודה **mkdir** יוצרת מדריך חדש. דוגמא:

```
mkdir d1
```

דוגמא זו יוצרת מדריך חדש בשם **d1** במדריך הנוכחי.

מחיקת מדריך rmdir (remove directory)

rmdir <שם-מדריך> <אופציות>

הפקודה **rmdir** מוחקת מדריך קיים, במידה ואין בו קבצים. דוגמא:

```
rmdir d1/d2
```

הפקודה הנ"ל מוחקת את המדריך d2 הנמצא במדריך d1 (יש למחוק תחילה את כל הקבצים במדריך d2).

תוים מיוחדים בשמות קבצים

ניתן להתייחס לשמות קבצים בצורה מקוצרת ע"י שמוש בתוים: * ? []

* מתאים לתו אחד או יותר

? מתאים לתו יחיד

[] מתאים לאחד מהתוים המופיעים בתוך הסוגריים.

דוגמאות:

ls f* הפרמטרים המועברים לפקודה **ls** הם כל שמות הקבצים ששםם מתחיל באות **f**, למשל: **f1 f2**

[mf]* שמות קבצים המתחילים באות **m** או באות **f**, למשל: **file**, **mbox**, **manual.1**

ch?? שמות קבצים המתחילים ב-'ch' ונגמרים בשני תוים כלשהם. למשל: chap ch.1, ch.[0-9] שמות קבצים המתחילים ב-'ch.' ונגמרים באחת הספרות 0 עד 9. למשל: ch.1 ch.3, *file* שמות קבצים המכילים את המילה file, למשל: profiles, myfile file

קבצים דחוסים

קובץ דחוס הוא קובץ שהופעל עליו אלגוריתם של דחיסה כדי להקטין את נפחו על הדיסק. בדרך כלל דוחסים קבצים גדולים שאינם בשימוש שוטף. לפני השימוש בקובץ דחוס - יש לפרוס אותו.

compress <שם-קובץ> הפקודה לדחיסת קובץ היא:

לאחר הדחיסה נוספת סיומת של Z. לשם הקובץ, והוא יתפוס פחות מקום בחשבונכם. כדי לשוב ולעשות בו שימוש יש לפרוס אותו מחדש.

uncompress <שם-קובץ> פקודה לפריסת קובץ היא:

הפעל **man compress** לקבלת עזרה על פקודות אלו.

פקודת דחיסה נוספת היא **gzip** הדוחסת קובץ בפורמט של GNU zip, הסיומת לקובץ דחוס בפורמט זה הוא **.gz**. (אגב, דחיסה זו נפתחת גם בדוחסים הנפוצים מעולם ה-PC "winzip" וכן "winrar")

דחיסת קובץ: **gzip <שם-קובץ>**

פריסת קובץ: **gzip -d <שם-קובץ>**

עזרה: **gzip -h**

עיון בקובץ ללא פריסה: **zmore <שם-קובץ>**

הגנה על מערכת הקבצים

לכל קובץ במערכת הקבצים מוגדרים שלושה סוגי הרשאות לשלוש קבוצות משתמשים. סוגי ההרשאות הם:

(-r read) לקריאת תוכן הקובץ, כולל העתקת הקובץ

(-w write) לכתיבה ושינוי תוכן הקובץ.

(-x execute) לביצוע או הרצת תוכנית-פקודה או הצגת רשימת קבצים בספריה.

שלוש קבוצות המשתמשים הן:

(-u user) המשתמש שיצר את הקובץ

(-g group) הקבוצה אליה משתייך בעל הקובץ, נקבע ע"י מנהלי המערכת

(-o other) שאר המשתמשים.

ראינו שניתן לראות את רשימת ההרשאות של הקבצים באמצעות הפקודה **ls -ls** ההרשאות מקובצות בקבוצות של שלוש - כשהשלישיה הראשונה מציגה הרשאות עבור המשתמש, השלישיה השניה מציגה הרשאות עבור הקבוצה אליה משתייך המשתמש והקבוצה השלישית מציגה הרשאות עבור שאר המשתמשים. התו " - " מסמן שאין הרשאה.

דוגמא:

המשתמש sam מפעיל את הפקודה: **ls -l f2** מקבל את הפלט הבא על המסך:

```
-rwxr----- 1 sam y2001 46 Feb 18 15:40 f2
```

התו הראשון מזהה את סוג הקובץ. '-' פרושו שהקובץ הוא קובץ רגיל.

שלושת התוים הבאים 'rwx' מציגים הרשאות קריאה כתיבה וביצוע עבור sam על הקובץ f2.

שלושת התוים שאחריהם 'r--' מסמנים שכל המשתמשים השייכים לקבוצה של sam, יכולים לקרוא את הקובץ, אך הם אינם יכולים לכתוב לתוכו או לבצע אותו. שלושת התוים האחרונים '---' מסמנים שלכל שאר המשתמשים אין הרשאת קריאה - כתיבה או ביצוע לקובץ f2. בעל הקובץ בלבד יכול לשנות את ההרשאות לקובץ.

שינוי הרשאות chmod (change mode)

chmod <שם-מדריך או שם-קובץ> <סוג-הרשאה><קבוצה>

ישנן שתי צורות עיקריות לשימוש בפקודה זו. נסביר להלן את הצורה הנוחה יותר למתחילים.

קבוצה היא קבוצת המשתמשים (o, g, u) לה נותנים-מסירים הרשאה:

+ מסמן הוספת הרשאה

- מסמן הסרת הרשאה

סוג-הרשאה הוא r, w, x.

שם-קובץ שם-מדריך עבור משנים את ההרשאה.

דוגמאות:

```
chmod g+w f2
```

הפקודה הנ"ל מאפשרת למשתמשים השייכים לקבוצת בעל הקובץ לכתוב על הקובץ f2.

```
chmod go+rw f2
```

הפקודה הנ"ל מאפשרת לכל המשתמשים לקרוא ולכתוב בקובץ f2.

ניתן לשנות הרשאות גם למדריכים. משמעות ההרשאות במקרה של מדריך מעט שונה:

-r מייצג רשות לראיית אלו קבצים מופיעים במדריך, כגון ע"י פקודת ls

-w מייצג רשות ליצירה או מחיקה קבצים מהמדריך

-x מייצג רשות ל"טיול" דרך המדריך, כגון ע"י הפקודה cd

כרגע ברירת המחדל של המערכת גורמת לכל הקבצים להווצר עם הגנה מקסימלית שאינה מאפשרת דבר, אף לא לחברי קבוצתך. במידת הצורך אפשר "לפתוח" קובץ זה או אחר באופן זמני לקריאה או כתיבה ע"י אחרים. תוכנית שפותחה במחלקה בשם **secureme** בודקת פרטים שונים בהגנה על קוד-המשתמש שלך ועל קבציו ואף מתקנות את הטעון תיקון. אחד מתפקידיה של תוכנית זו הוא "לסגור" חזרה קבצים שהיו "פתוחים" לגישה לאחרים. רצוי להריץ את secureme אחת לשבוע עד אחת לחודש.

למעשה אם ברצונך לתת לחברך להעתיק קובץ שלך רצוי לעשות זאת או במשלוח בדואר אלקטרוני, ראה בהמשך, או באמצעות שימוש במדריך /tmp עליך לבצע:

```
cp my-file /tmp
```

```
chmod a+r /tmp/my-file
```

לאחר מכן חברך יבצע את הפקודה, באותו מחשב:

```
cp /tmp/my-file ~/his-file
```

ובסוף תוכל למחוק את הקובץ ע"י:

```
\rm /tmp/my-file
```

צורה שניה לשימוש בפקודה chmod משתמשת בייצוג הבינארי של הרשאות הקובץ.

ע"י התייחסות להרשאות הקובץ כמספר בינארי, וחלוקת המספר ל-3 מספרים בני 3 ספרות כ"א, כך שכל מספר מייצג קבוצה (המשתמש u, הקבוצה g, והאחרים o) וכל סיפורה מייצגת הרשאה, לדוגמא:

$rwX = 111 = 7$, $r-- = 100 = 4$, $r-x = 101 = 5$, $rw- = 110 = 6$

עתה, אם ברצונך להעניק הרשאות מלאות לעצמך, קריאה וכתיבה לקבוצה וקריאה בלבד לאחרים.

$rwX, rw-, r-- = 111, 110, 100 = 764$

הרשאות הקובץ צריכות להיות:

נסביר זאת בעזרת דוגמא:

```
ls -l f1
```

הפלט שהתקבל:

```
-rw-r--r-- 1 sam y2001 60 Aug 07 13:24 f1
```

ניתן לראות שהקובץ f1 ניתן לקריאה ע"י כל המשתמשים. sam רוצה לבטל את הרשאת הקריאה לקבוצה ולאחרים, ולאפשר רק לעצמו קריאה וכתיבה לקובץ. כלומר לשנות את ה-mode ל: rw-----. סימון זהה

הינו 110,000,000 או 600, והפקודה אותה יש להריץ היא: **chmod 600 f1**

5. העורך – vi

- קיימות מספר דרכים ליצירת קובץ במערכת. בפרק זה נדון בדרך אחת - עבודה עם העורך vi. vi כשמו כן הוא "וי". למען האמת אינך חייב להשתמש בו. אנו ממליצים על שימוש בעורך המהולל הידוע לתפארה לכל קבצי תבל הלא הוא emacs אם כן תשאלו למה vi? התשובה היא:
- compatibility Backward.
 - vi נמצא בכל מערכת unix בסיסית - emacs הוא תוספת שאינה כלולה במערכת. על emacs תוכל לקרוא בתיעוד נפרד של המחלקה.

הפעלת העורך vi

בכדי להפעיל את העורך מקישים:

vi <שם-קובץ>

vi f1

לדוגמא:

הפקודה הנ"ל יוצרת קובץ חדש בשם f1 במדריך הנוכחי, במידה והקובץ f1 קיים אז הקובץ f1 יוצג לעיבוד.

לעורך vi קיימים שני מצבים של עבודה - mode insert ו mode command. זהו מצב העבודה של העורך מיד לאחר mode command מפורשים התווים המוקשים לפקודות. זהו מצב העבודה של העורך מיד לאחר הפעלתו.

פקודות vi

רוב הפקודות בתוכנית vi מיוצגות ע"י אחד או שני תווים. אם ברצוננו שפקודה של vi תתבצע מספר פעמים אנו מקדימים לפקודה מספר המציין את מספר החזרות שיש לבצע את הפקודה או ע"י הקשת ".". בתוכנית vi יש חשיבות רבה להגדרת ה"מקום הנוכחי". אנו נשתמש בהמשך בהגדרת ה"תו הנוכחי", ה"מילה הנוכחית" וה"שורה הנוכחית". קיימת חשיבות רבה להקשת הפקודה באותיות קטנות או גדולות. לפקודות המוקשות באותיות קטנות או גדולות יש משמעות שונה כפי שנראה בהמשך.

הפקודות בתוכנית vi מחולקות לקבוצות הבאות:

1. הוספת טקסט - מחיקת טקסט
2. איחזור טעויות - חרטה וחזרה
3. החלפת טקסט
4. חיפוש ותנועה בקובץ
5. העתקה והזזת טקסט
6. פקודות שונות
7. גמר עבודה

הוספת טקסט

כל פקודה להוספת טקסט מעבירה את התוכנית vi ל-mode insert במצב עבודה זה כל תו מוקש יופיע כפי שהוצג על גבי המסך בקובץ. בכדי לחזור ל-mode command יש להקיש ESC.

פקודות הכנסת הטקסט הן כדלקמן:

a (append)	הוספת טקסט לאחר התו הנוכחי
A	הוספת טקסט בסוף השורה
i (insert)	הוספת טקסט לפני התו הנוכחי
o (open)	הוספת טקסט בשורה חדשה מתחת לשורה הנוכחית
O	הוספת טקסט בשורה חדשה מעל לשורה הנוכחית

R (replace) הדפסה על הטקסט הנוכחי, במקומו. בדומה למצב overwrite.

מחיקת טקסט

X	מחיקת התו הנוכחי
dw (delete word)	מחיקת המילה, מחרוזת הנוכחית
dd	מחיקת השורה הנוכחית
d\$	מחיקת הטקסט מהמקום הנוכחי עד לסוף השורה.

לדוגמא:

5X	מוחק חמישה תוים מהתו הנוכחי.
dd3	מוחק 3 שורות כולל את השורה הנוכחית.

איחזור טעויות (Retrieve) חרטה (Undo) וחזרה (Redo)

u	בטל פקודה אחרונה והחזר את מצב הטקסט לקדמותו
U	שחזר שורה נוכחית
. (נקודה)	חזור על פעולה-פקודה אחרונה
" p n	איחזור המחיקה ה-ח-ית האחרונה.

החלפת טקסט

R (Replace)	החלפת התוים הנוכחיים בתוים המוקשים (overwrite).
r (replace)	החלפת התו הנוכחי בתו הבא שמוקש לאחר התו r.
s (substitute)	החלפת התו הנוכחי בקבוצת התוים הבאה. כאן אנו עוברים ל-mode insert ולכן יש לסיים את הכנסת הטקסט בהקשת ESC.
S או cc	החלפת השורה הנוכחית בקבוצת התוים הבאה. כאן אנו עוברים ל-mode insert ולכן יש לסיים את הכנסת הטקסט בהקשת ESC.
cw	החלפת המילה-מחרוזת הנוכחית בתוים הבאים. התוכנית עוברת ל-mode insert יש להקיש ESC בסיום הקשת הטקסט.
C או \$c	החלפת טקסט ממקום הסמן עד סוף השורה. תוכנית vi עוברת ל-mode insert ויש להקיש ESC בסיום הקשת הטקסט החדש.

תנועה וחיפוש

בתחנות העבודה מקשי החיצים מתוכנתים בד"כ לעבודה בתוכנית vi.

תנועה של תו בודד

h או חץ שמאלי	תנועה של תו אחד שמאלה
l או חץ ימיני	תנועה של תו אחד ימינה
j או חץ למטה	תנועה של שורה אחת למעלה
k או חץ למעלה	תנועה של שורה אחת למטה

תנועה של מסך (Paging)

(up u) + ctrl	הזזת חצי מסך כלפי מעלה
(down d) + ctrl	הזזת חצי מסך כלפי מטה
(forward f) + ctrl	הזזת מסך שלם קדימה
(backward b) + ctrl	הזזת מסך שלם אחורה
G	הבאת הסמן לסוף הקובץ
n	הבאת הסמן לתחילת שורה מספר n.

חיפוש מחרוזת

חפש את המחרוזת קדימה מהמקום הנוכחי והבא את הסמן לתחילת המחרוזת שמצאת	/<מחרוזת>
חפש את המחרוזת אחורה מהמקום הנוכחי והבא את הסמן לתחילת המחרוזת שמצאת	<מחרוזת?>
חזור על החיפוש האחרון באותו כיוון חיפוש	n (next)
חזור על החיפוש האחרון בכיוון חיפוש הפוך	N

העתקה והזת טקסט

הטקסט האחרון שנמחק נשמר באוגר register של התוכנית. למעשה קיימים מספר אוגרים. ניתן להשתמש בזכרון האוגר באמצעות הפקודות p ו-P (put).	
ממקם את תוכן האוגר לאחר מקום הסמן. במידה ובאוגר קיימות שורות אזי שורות אלו יופיעו מתחת לשורה הנוכחית.	p
ממקם את תוכן האוגר לפני הסמן. במידה והאוגר מכיל שורות שלמות, אזי שורות אלו תופענה מעל לשורה הנוכחית.	P
בשני המקרים, תוכן האוגר אינו משתנה ולכן ניתן לחזור על הפעולה מספר פעמים.	
העתקת השורה הנוכחית לאוגר.	Y (yank)
חבר את השורה הבאה לשורה הנוכחית.	J (join)

פקודות נוספות

בתוכנית vi קיימות פקודות נוספות אשר ניתנות מהשורה האחרונה של המסך. פקודות אלו מתחילות עם התו " : " .	
אנו נציין רק מספר פקודות:	
<שם-קובץ> r העתק את "שם-קובץ" אל תוך הטקסט הקיים החל מהשורה הבאה.	
<שם-פקודה> !: בצע את הפקודה. הכוונה לפקודת מערכת ההפעלה.	
<Ctrl>+g או f: הצג את שם הקובץ הנוכחי ומספר השורה בו נמצא הסמן.	
לדוגמא:	
!ls : הצגת רשימת הקבצים במדריך הנוכחי וחזרה לתוכנית vi.	
!pwd : הצגת שם המדריך הנוכחי.	

סיום העבודה

כל השינויים שביצענו נשמרו בקובץ זמני, אך לא בקובץ המקורי. בכדי לשמור את השינויים על גבי הקובץ המקורי עלינו להשתמש בפקודת write. בכדי להמנע מבעיות בשמירת הקובץ, לדוגמא עקב הפסקת חשמל, כדאי מפעם לפעם לשמור בזמן העבודה את השינויים שביצעת בקובץ על גבי הדיסק. במידה והתוכנית מודיעה הודעת שגיאה בעת שהנך מנסה לשמור את הקובץ, מומלץ לפנות לתמיכה לקבלת הדרכה לפני שמסיימים לעבוד בתוכנית vi.	
w : שמור את השינויים שביצעת עד כה על גבי הדיסק.	
q : סיים את העבודה בתוכנית vi.	
wq : שמור את השינויים על גבי הדיסק וסיים את העבודה ב-vi.	
q! : סיים את העבודה בתוכנית vi ללא שמירת השינויים.	
<שם-קובץ> w : שמור את הטקסט בקובץ ששמו "שם-קובץ".	
<שם-קובץ> e : ערוך את "שם-קובץ". פקודה זו משמשת לעריכת קובץ חדש בתוך תוכנית vi במידה ולא שמרת את הקובץ הנוכחי, עליך לשמור אותו לפני השימוש בפקודה זו.	
סיום זמני של עריכת הקובץ ללא שמירתו. פקודה זו משמשת לדוגמא, במקרה בו אינך יכול לשמור את הקובץ ע"ג הדיסק, המערכת לא מאפשרת את שמירת הקובץ.	z + ctrl
על מנת לחזור לעריכת קובץ זה ב-vi יש להקיש: fg, קיצור של foreground.	
ללא נקודתיים, קיצור של wq .	ZZ

6. דואר אלקטרוני

דואר אלקטרוני הינו שירות למשלוח מכתבים בין מחשבים. ב- UNIX קיימות מספר תוכניות לדואר אלקטרוני. בפרק זה נדון בהרחבה בתוכנית ה- mail שהיא תוכנית הדואר הסטנדרטית של UNIX. כמו כן ניתן סקירה כללית על elm ו- pine שהם תוכניות "מצג מלא" - full (screen).
אנו ממליצים להשתמש בתוכנית pine הנוחה והקלה לשימוש.

כתובות

כדי לשלוח מכתב למשתמש יש לדעת את הכתובת שלו בדואר אלקטרוני. מבנה של כתובת היא:
<שם משתמש> @ <כתובת מחשב>

דוגמא:

הכתובת המלאה של משתמש במחלקה היא: sam@macs.biu.ac.il
הסיומת ac.il מופיעה בכתובות של כל האוניברסיטאות בארץ. לכן, משתמש בשם ben במחשב planet יכול לקבל מכתבים לכתובת: ben@macs.biu.ac.il אם הממוען נמצא באותו מחשב בו אתה נמצא אפשר לוותר על כתובת המחשב ולציין רק את שם המשתמש.

שליחת דואר mail

על מנת לשלוח מכתב למשתמש אחר, יש להקיש את הפקודה: כתובת-הנמען mail

דוגמא:

```
mail sam@vm.biu.ac.il
```

באמצעות הפקודה הנ"ל, נשלח מכתב למשתמש sam, במחשב vm.biu.ac.il. במערכות מסוימות תופיע בשלב זה על המסך בקשה למלא את נושא המכתב (subject). כתוב את תוכן המכתב בשיטת ה- editor line. ניתן לתקן שגיאות רק בשורה הנוכחית. עם סיום המכתב הקש את התו ". " בתחילת שורה חדשה. ניתן לכתוב את המכתב בקובץ באמצעות עורך טקסט כלשהו (כגון emacs) ולאחר מכן לשלוח את הקובץ בדואר.

דוגמא:

```
mail sam@vm.biu.ac.il < f1
```

בדוגמא זו נשלח ל-sam קובץ בשם f1 הנמצא במדריך הנוכחי. על משמעות התו "<" ראה פרק שירותי ה- Shell.

קריאת דואר

במערכת קיימות שתי תיבות דואר: תיבת דואר כללית: mailbox system - המאחסנת את כל המכתבים החדשים המגיעים למערכת וכן תיבת דואר פרטית בשם mbox - הנמצאת בד"כ במדריך הראשי של כל משתמש.

כדי לקרוא מכתבים חדשים מקישים את הפקודה: mail

במידה וקיים דואר ב-mailbox system, אזי תוצג בפני המשתמש רשימת המכתבים ופרטים הכוללים: מספר סידורי של המכתב, שם השולח, תאריך ושעת המשלוח.

ניתן לעיין במכתבים בזה אחר זה ע"י הקשת Enter/Return או להקיש את מספר המכתב כדי לקרוא את תוכנו.

יציאה מהדואר

ישנן שתי דרכים לצאת מתוכנית הדואר: יציאה עם שמירת השינויים ויציאה ללא שמירת השינויים. באפשרות הראשונה, כל שינוי שנעשה בתיבת הדואר, כגון מחיקת קבצים, יתבצע באופן סופי. כמו כן, כל

מכתב שנקרא ולא נמחק יועבר לתיבת הדואר הפרטית שלך - mailbox. באפשרות השניה, כל השינויים שנעשו לא מתבצעים והמכתבים שנקראו נשארים ב-mailbox system. יציאה תוך שמירת השינויים נעשית ע"י הפקודה "q". יציאה ללא שמירת השינויים נעשית ע"י "x". קריאת דואר מתיבת הדואר הפרטית mailbox מתבצעת ע"י הפקודה: mail -f

elm

קריאת דואר

elm מאפשר טיפול בדואר אלקטרוני בצורה פשוטה, מובנת וקלה לשימוש. מפעילים את תוכנית הדואר האלקטרוני elm ע"י הפקודה elm מתקבל מסך מכתבים, המציג את רשימת המכתבים שנשלחו אליך ונמצאים בתיבת הדואר הכללית.

pine

יתרונות pine

גם pine מאפשר טיפול בדואר אלקטרוני בצורה פשוטה, מובנת וקלה לשימוש. יתרון בולט של pine הוא האפשרות לשלוח קבצים בינאריים, כגון קבצי תמונה, קבצי קול, יחד עם המכתב. זאת באמצעות קישור (attachment) של הקובץ למכתב. כמו כן ניתן בצורה פשוטה לשלוח מכתב בעברית, במידה והמסוף שלך תומך בפונטים עבריים.

צורת העבודה עם pine

הכנס ל-pine באמצעות הפקודה: pine בתפריט הראשי שיתקבל תוכל לבצע מספר פונקציות. בחירת הפונקציה נעשית באמצעות מקשי החיצים והקשה על Enter/Return או ע"י הקשת התו המתאים לאותה פונקציה. למשל:

C	לכתיבת מכתב באנגלית ושליחתו
B+ctrl	לכתיבת מכתב בעברית ושליחתו
	לעיון במכתבים בתיבת דואר
Q	ליציאה מ-pine

בעת כתיבת מכתב, pine משתמש בתוכנית עריכה משלו שנקראת pico

פקודות עריכה עיקריות:

ctrl + D	מחיקת תו נוכחי
ctrl + K	מחיקת שורה נוכחית
ctrl + U	ביטול מחיקת שורה אחרונה
ctrl + J	הכנסת קובץ חיצוני לטקסט המכתב.
ctrl + T	בדיקת איות

לאחר סיום כתיבת המכתב ניתן לשלוח אותו ע"י ctrl+x או לבטל אותו ע"י ctrl + C.

בכל שלב ילווה אותך תפריט בתחתית המסך שיתן לך את האופציות האפשריות.

ממשקים נוספים

ישנו ממשק נוסף פנימי לעורך emacs לשליחת דואר ולקבלת דואר.
לקרוא דואר השתמש ב- `rmail M-x`
וכדי לשלוח דואר השתמש ב- `m x + ctrl`
ממשק זה אינו מומלץ לשימוש.

שאלה: מה יתרונתיו של "קורא הדואר" `less $MAIL` מתי כדאי להשתמש בו?.

7. שירותי ה-Shell

חלק גדול מעוצמת UNIX נזקף לזכות תוכנית ה-Shell. זוהי תוכנית שרצה עם כניסתך למערכת ועד ליציאתך ממנה. תוכנית ה-Shell היא זו המדפיסה את ה- prompt, קולטת את הפקודות אותן אתה מקיש ומעבירה אותן למערכת לביצוע. משום כך, היא משמשת מימשק בין המשתמש לבין המערכת.

קיימות מספר תוכניות Shell כגון:

Bourne shell	(sh)
Korn shell	(ksh)
C shell	(csh)
Tcsh	(tcsh)

אם ברצונך לבדוק מהי תוכנית ה-Shell הרצה אצלך הפעל את הפקודה **ps** ה-Shell מופיע בטור ה- COMMAND או COMD. יתכן שגם הפקודה `echo $SHELL` תעזור לך גם-כן. הסבר על הפקודה ps תוכל למצוא ב-"נספח ב' - תוספת למתקדמים" בפרק על עבודות (JOBS). נספח זה דן בהרחבה גם ב- Shells העיקריים וההבדלים ביניהם. אנו נניח שהינך עובד עם csh או tcsh, אך באותם מקומות בחוברת בהם יש שינויים לעומת Shells אחרים נציין זאת במפורש.

בפרק זה נדון בשירותים שונים שתוכנית ה-Shell מספקת:

- הכוונת קלט-פלט
- צינורות pipes
- שמות קיצור לפקודות aliasing
- אחסון וחזרה על פקודות history

אם ברצונך להעמיק יותר בנושא ה- Shells, תוכל לעשות זאת באמצעות "נספח ב' - תוספת למתקדמים" הדן ב:

- משתני ה-Shell
- עבודות (JOBS)
- Scripts Shell
- השלמת שמות קבצים
- תקון טעויות כתיב.

הכוונת קלט-פלט

כל תוכנית הרצה תחת UNIX, והפקודות שלמדת הן בסך הכל תוכניות, על פי רוב מקבלת, אם לא בקשת בפירוש אחרת, קלט תקני מהמקלדת ושולחת פלט תקני למסך. (מלבד קבצים אחרים שהתוכנית יכולה להשתמש בהם, כמו קבצי ביניים, קבצי קונפיגורציה וכדומה) את ברירות המחדל האלו ניתן לשנות! ראינו שהפקודה `finger` מציגה את רשימת המשתמשים המחוברים למערכת. נניח שאנו מעונינים לשמור את הרשימה בקובץ. UNIX מאפשר בפשטות לנתב את הפלט התקני של פקודה לקובץ ע"י שימוש בתו ">". דוגמא:

```
Finger > users.list
```

אם הקובץ `users.list` לא היה קיים הוא ייוצר, אך אם הקובץ היה קיים תוכנו הישן עשוי להמחק. התוים ">" מאפשרים שרשור פלט של פקודה לסוף קובץ קיים בלי למחוק את תוכנו הקודם. כמו כן, קיים התו "<" לניתוב קלט תקני של פקודה.

דוגמא: אם למשל `sort` היא תוכנית הקוראת נתונים מהמסך, הפקודה הבאה תקרא את הנתונים מהקובץ `users.list` במקום מהמסך:

```
cat users.list < sort
```

צינורות (pipes)

נניח שאנו מעונינים למיין את הפלט של הפקודה `finger` לפי השדה הראשון, שם משתמש.

הפקודה `sort` תבצע זאת ע"י `cat users.list < sort` הרשימה הממויינת תוצג על המסך. לא יחול שום שינוי בקובץ `users.list`. למעשה ניתן לקצר את התהליך ולקבל רשימה ממויינת של משתמשים ללא שימוש בקובץ ביניים כלל ע"י שימוש ב-`pipe`.

`pipe`, מיוצג ע"י התו `|`, מאפשר להשתמש בפלט התקני של פקודה אחת כקלט תקני לפקודה הבאה אחריה. דוגמא:

```
sort | finger
```

כאן הפלט של הפקודה `finger` מופנה כקלט לפקודה `sort` הממיינת אותו ומציגה את התוצאה על המסך. אם נרצה לשמור את הרשימה הממויינת בקובץ, נוכל לשלב הכוונת פלט יחד עם `pipe`:

```
finger | sort > users.sorted
```

דוגמא שימושית ל-`pipe` היא כאשר מספר שורות הפלט של פקודה עולה על מספר השורות במסך ואז משתמשים ב-`pipe` עם הפקודה `less`. דוגמא:

```
ls -C /bin | less
```

שמות קיצור לפקודות (Aliases)

כאשר עובדים עם ה-`C-shell` או עם ה-`Tcsh` ניתן להגדיר שמות קיצור לפקודות באמצעות הפקודה: `alias <שם-קיצור> <"פקודה">`

ב-`Ksh` מפרט הפקודה מעט שונה. ראה טבלה 1. השוואה בין ה-`Shells` ב-`UNIX`.

דוגמאות:

```
alias up "cd .."
alias = "ls -a"
alias ls "ls -l"
```

פקודה זו מגדירה שם קיצור `up` אשר מפעיל את הפקודה `cd ..`.
 פקודה זו מגדירה שם קיצור `=` לפקודה `ls -a`.
 פקודה זו מגדירה מחדש את הפקודה `ls`, הקשת `ls` תפעיל את אותה הפקודה עם האופציה `-l`. אם רוצים להפעיל את הפקודה `ls` המקורית, יש לרשום את המסלול המלא שלה: `/bin/ls` או להקדים לה `ls` backslash.

על מנת לקבל את רשימת הגדרות שמות הקיצור ב-`shell` רושמים: `alias`

על מנת להסיר הגדרה מסויימת רושמים: שם-קיצור `unalias`

אחסון וחזרה על פקודות (history)

בד"כ מעונינים להשתמש שוב בפקודה קודמת או בשנוי קל שלה. לשם כך ה-`C-shell` וה-`Tcsh` מנהלים מאגר `buffer history`, אשר שומר את הפקודות האחרונות שהוכנסו ע"י המשתמש.

מספר הפקודות הנשמרות נקבע ע"י הפקודה: `set history = <מספר>`

דוגמא:

```
set history = 50
```

הפקודה הנ"ל מאפשרת לשמור את הפקודות אותן אתה מקיש לתוך מאגר.

כדי לעיין בפקודות המאוחסנות רשום את הפקודה: `history`

קלט הפקודה יראה בערך כך:

```
John mail    31
project.cc more 32
ps man      33
who         34
history     35
```

ניתן לחזור על פקודות מאוחסנות ע"י:

!! חזור על הפקודה האחרונה.

!ma חזור על הפקודה האחרונה שמתחילה בתוים 'ma', במקרה זה `ps man`

!31 חזור על פקודה מספר 31, במקרה זה `john mail`.

תיעוד מורחב על פקודות חוזרות ועריכתן ניתן לקבל ע"י: `man csh`

הערה: ה-`Tcsh` מאפשר דפדוף ועריכת פקודות מאוחסנות בצורה נוחה יותר לשימוש ראה "נספח ב' - תוספת למתקדמים" בפרק ה-`Shells` בהרחבה.

8. שימוש בקומפיילרים - "מהדרים"

פרק זה מיועד לכותבי תוכניות בשפות התכנות אסמבלי, C ו- C++ אם אינך משתמש ב-UNIX בשפות הנ"ל תוכל לדלג על פרק זה.
 כל תוכנית עוברת שלושה שלבים עיקריים:

1. כתיבת התוכנית ועריכתה.
2. הידור, קומפילציה - תרגום התוכנית ע"י המהדר לשפה המובנת למחשב.
3. הרצת התוכנית.

המהדר הוא תוכנית המקבלת כקלט את התוכנית שכתבת ונותנת כפלט קובץ מוכן לריצה בשפת מכונה. עבור כל שפת תכנות קיים מהדר נפרד. כשלושן בהידור התוכנית כפי שקורה בד"כ בפעמים הראשונות מצריך חזרה לעורך כדי לתקן את השגיאות.

שפת אסמבלי

הפקודה להפעלת המהדר לשפת אסמבלי היא: שם-קובץ as
 כאשר "שם-קובץ" הוא שם תוכנית אסמבלי בעל סיומת ".s", למשל: targil.s .

שפת C

מהו C?

C, noun: A programming language that is sort of like Pascal except more like assembly except that it isn't very much like either one, or anything else. (Ray Simard)

C היא שפת האם של מערכת ה-UNIX.

הפקודה להפעלת המהדר לשפת C היא: שם-קובץ cc או שם-קובץ gcc , כאשר "שם קובץ" הוא שם תוכנית C ב על סיומת ".c", למשל: targil.c .

זכור ! תן שם אחר לקובץ הפלט !!!

שפת C++

הפקודה להפעלת המהדר לשפת C++
 כאשר "שם-קובץ" הוא שם תוכנית ++ C בעלת סיומת ".cpp", למשל targil.cpp .
 הערה: המהדרים יכולים לעבד גם קבצי object, סיומת ".o", וקבצי assembler, סיומת ".s".
 אם לא צוינו אופציות בפקודת ההידור, המהדר מבצע גם את שלב ה- "linkage" של התוכנית ע"י הפעלת ה- (editor linkage ld) ויוצר קובץ בשם a.out המוכן לריצה. מריצים אותו ע"י הקשת a.out .

אופציות של מהדרים

הפעל שם-מהדר man לקבלת מידע נרחב על האופציות של מהדר מסויים.

להלן תקציר האופציות העקריות המשותפות למהדרים השונים:

-c הידור התוכנית ללא הפעלת ה- (editor linkage ld).

נוצר קובץ עם סיומת ".o"

שם-קובץ -o הקובץ המוכן להרצה שנוצר ע"י המהדר יקרא "שם-קובץ" ולא a.out .

-O קריאה לאופטימיזציה במהלך ההידור.

האופטימיזציה יוצרת code יותר יעיל יותר ולכן מקטינה את זמן ריצת התוכנית, עם זאת - זמן ההידור גדל. השיפור בתוכנית כולל: חישוב חד-פעמי של ביטויים זהים שיש בהם שמוש חוזר במשך התוכנית, משתנים המחושבים בתוך לולאה והם בלתי תלויים במשתני הלולאה, מחושבים מחוץ ללולאה ועוד.

-g מייצר טבלה סימבולית מיוחדת הדרושה להפעלת ה- debugger, ראה להלן.

שם-ספריה -l "קישור" לספריה המצויינת בזמן ה- linkage. ספריות רבות נמצאות במדריך lib-usr-

ויש להתייחס אליהן בשמות מקוצרים.

למשל: ספריה libX11.a תצוין ע"י: X11 -l ספריה המתמטית libm.a

תצוין ע"י: -lm, כלומר, מצינים את התוים שאחרי "lib" ולפני הסיומת ".a" .

שם-מדריך L - אם מיקום הספרייה המצויינת באופציה l- אינו במדריך -usr/lib
אלא במדריך אחר, יש לציין את שם המדריך אחרי האופציה L.

דוגמא:

```
cc -c main.c sub1.c sub2.c
cc -o mainx main.o sub1.o sub2.o
```

mainx

בפקודה הראשונה קימפלנו תוכנית C בשם main.c הקוראת לשתי סברוטיות חיצוניות, sub1.c ו-sub2.c. בשל האופציה c- לא התבצע קישור וקיבלנו שלושה קבצי sub1.o (object, main.o) ו-sub2.o. בפקודה השנייה בצענו קישור של הקבצים. כיוון שהסיומת שלהם היא ".o" - הם מועברים ישירות ל-editor linkage. לקובץ המוכן לביצוע קראנו mainx, ע"י שימוש באופציה o- והפעלנו את התוכנית ע"י mainx.

הודעות שגיאה של המהדר

אזהרות והודעות שגיאה נשלחות בעת הידור תוכנית לפלט שגיאות הסטנדרטי, בד"כ למסך. הודעות אלו מכילות את מספר השורה השגויה בתוכנית והסבר מקוצר של הודעת השגיאה. ניתן להפנות את הודעות השגיאה לקובץ ע"י התוים ">&" , נכון עבור csh ו-tcsh. עבור shells אחרים ראה "טבלה 1.1. השוואה בין Shells ב-Unix".

דוגמא:

```
gcc -o prog prog.c -lm >& prog.error
```

התוכנית prog.c עוברת הידור ובשלב ה-linkage חל קישור עם הספרייה המתמטית lib/libm.a-/usr. הקובץ שנוצר ומוכן להרצה ייקרא prog וכל הודעות השגיאה בזמן ההידור ייכתבו בקובץ prog.error.

הרצת תוכנית

לאחר הידור מוצלח של התוכנית נוצר, כאמור, קובץ מוכן לריצה בד"כ a.out. אותו מריצים ע"י הקשת שמו.

בעיות שכיחות

1. בעיות לוגיות: קיבלת משהו מאד לא דומה למה שציפית לקבל. שבור את הראש, רק לוגית. יש לך bug ורק אתה, בעזרת debugger והדפסות ביניים או מחשבה נוספת - יכול לפתור את בעיותיך. אין שום טעם לשאל חברת מתרגל ו/או איש-תמיכה לפני שחקרת רבות אודות מקור ה-bug!
ב 100% מהמקרים - המחשב לא התקלקל. ב - 100% מהמקרים הוא לא שונא אותך...

2. קיבלת הודעה: a.out: denied permission

הודעה זו אומרת שאין לך הרשאה להריץ את הקובץ a.out עקב תקלה בהידור.

3. faults למיניהם: ישנן הודעות מגוונות ביותר שהמערכת מספקת כאשר היא אינה יכולה להמשיך את ההרצה הנוכחית. לרוב - הבעיה היא בעיה לוגית בתוכניתך. חזור לשלב 1. ודבג!
לדוגמא: הודעה dumper core error: bus הקובץ core מכיל מידע, לא בשבילך, ל-debugger בלבד! על מצב תוכניתך בזמן הנפילה. פרטים בקורס "מבוא למחשבים".
חכה לתרגיל ההרצה הראשון ותבין למה לא תמיד מגיעים בנוחות כשנוסעים באוטובוס.

dbx Debugger

Debugger הוא כלי לניפוי שגיאות, המאפשר מעקב אחרי ביצוע של קובץ שעבר קומפילציה, קובץ מסוג object. בכדי לעבוד עם ה-debugger יש לציין את האופציה g- בפקודת ההידור. דוגמא:

```
cc -g -o prog prog.c
```

בדוגמא זו ביצענו קומפילציה של התוכנית prog.c. נוצר קובץ בשם prog מוכן לריצה. האופציה g מבקשת מהקומפיילר ליצור מידע הדרוש להפעלת ה-Debugger.

ב-UNIX קיים Debugger בשם dbx. מפעילים אותו ע"י:

```
object < dbx -שם-קובץ
```

פקודות dbx

רשימה מקוצרת של פקודות dbx	help
יציאה מ-dbx	quit
רשימת נקודות עצירה ומעקב, trace	status
רשימת המשתנים וערכם	set
קבע נקודת עצירה בשורה n	n at stop
עצור בפרוצדורה הנתונה	in stop
התחל את ביצוע התוכנית ועצור בנקודות הציון	run
הדפס ערכו של "שם-משתנה" בכל פעם שערכו משתנה	trace שם-משתנה
התקדם צעד אחד בריצת התוכנית	step
התקדם צעד אחד בריצת התוכנית דלג על פרוצדורות.	next
המשך את ביצוע התוכנית עד לנקודת העצירה הבאה	cont
הדפס את ערך המשתנה	שם-משתנה print
כתוב את שמות כל הפונקציות הפעילות שבמחשנית, "stack".	where

gdb

הוא debugger נוסף דומה ל-dbx. הכנס אליו באמצעות **<שם-קובץ-object> gdb** .
 אנו משאירים לקורא את לימוד ה-gdb באמצעות הפקודה הפנימית שלו: help.
 במערכת קיים מימשק חלונות לתוכנית gdb.
 תחילה יש להפעיל את מערכת החלונות, ראה נספח א. X-WINDOWS מערכת חלונות.
 בתוך חלון יש להקיש: **<שם-קובץ-object> mxgdb**
 מתקבלת סדרה של חלונות: אחד עבור ה-source של התוכנית, אחד עבור עיון בערכי המשתנים, חלון אחר מאפשר הקשת פקודות dbx ועוד.

ספריות

הפקודות המבצעות קומפילציה, gcc cc, g++, linkage עם הספריות הסטנדרטיות. קישור עם ספריות נעשה ע"י שימוש באופציה -l. למשל קישור לספריה המתמטית libm.a ולספריה הגרפית libX11.a ייעשה ע"י:

```
cc file.o -lm -lX11
```

ספריות ar

ספריות אלו מכילות מספר רב של מודולים, כאשר משתמשים בחלק מהם לבניית תוכנית. למשל, הספריה המתמטית המכילה פונקציות מתמטיות שונות היא ספריית ar. הפקודה לטיפול בספריית ar היא: **פרמטרים אופציות ar** יש לציין את האופציות ללא רווחים ביניהן.

האופציות העיקריות:

הצגת תוכן הספריה	(table)	t
מחיקת מודולים מהספריה	(delete)	D
הוספת/החלפת מודולים בספריה	(replace)	r
הצגת פרטים על המודולים בזמן הטיפול בספריה	(verbose)	v
עדכון טבלת התוכן של הספריה.	(symbol)	s

דוגמאות

יצירת ספריה בשם mylib שתכיל את המודל f1.o.	f1.o mylib.a rsv ar
הוספת מודול f2.o לספריה.	f2.o mylib.a rsv ar
הצגת תוכן הספריה.	mylib.a t ar
ביטול מודול מהספריה.	f1.o mylib.a dvs ar
קומפילציה וקישור תוכנית לספריה.	-mylib.a main.o cc

אם משתמשים ביותר מספריית ar אחת יש חשיבות לסדר בו הן מופיעות בפקודת הקישור.

10. עבודה ברשת (Networking)

המחלקה מרושתת ברשת Ethernet ומחוברת לרשת הכלל-עולמית הידועה בשם Internet. אחד מהשרותים שרשת Internet מאפשרת הוא משלוח מכתבים בדואר אלקטרוני בין מחשבים הקשורים ברשת. ראה פרק 6. דואר אלקטרוני. בפרק זה נמנה שרותים נוספים.

פקודות הפעלה מרחוק (Remote) בין מחשבי UNIX

פקודות Remote בין מחשבי UNIX מאפשרות לך לעבוד על מחשב מקומי ולבצע פקודות במחשב אחר, בד"כ מרוחק, ללא התחברות, login למחשב זה. לשם כך, אתה צריך שם משתמש במחשב המרוחק. בפרק זה נדון בפקודות rcp, rlogin, rsh.

הפקודה rsh

הפקודה rsh מאפשרת לך לבצע פקודות במחשב UNIX אחר בו יש לך שם משתמש (login) או להתחבר למחשב זה. מפרטי הפקודה: (remote shell) **<פקודה>** **<כתובת-מחשב-מרוחק>** rsh בדוגמאות הבאות נשתמש בשמות מקוצרים לכתובת המלאה של המחשבים בהנחה ששמות אילו מוגדרים במערכת.

דוגמא: משתמש העובד במחשב planet ויש לו שם משתמש על מחשב urim יכול לראות מי עובד על מחשב urim באמצעות: `who urim rsh`
אם לא מציינים פקודה ב-rsh - מתבצע login למחשב המצויין, בדומה לפקודה rlogin (ראה בהמשך), ואז ניתן לציין את שם המשתמש ע"י: **<שם משתמש>** -1 **<כתובת-מחשב-מרוחק>** rsh

הפקודה rlogin

הפקודה rlogin מאפשרת למשתמש העובד במחשב UNIX מקומי להתחבר למחשב UNIX מרוחק ברשת. המשתמש יתבקש להכניס סיסמא אם המחשב המרוחק אינו בוטח במחשב שבו אתה עובד.

מפרטי הפקודה: **<שם משתמש>** -1 **<כתובת-מחשב-מרוחק>** rlogin **דוגמא:** משתמש dcohen במחשב planet שיש לו שם משתמש cohend במחשב urim.ns.biu.ac.il יכול להתחבר למחשב זה ע"י:

```
rlogin urim.ns -l cohend
```

הפקודה rcp

הפקודה rcp מאפשרת העתקת קבצים בין מחשבי UNIX הקשורים ברשת. לשם כך יש לעדכן את הקובץ rhosts במחשב המרוחק, ראה להלן.

ההתייחסות לשם קובץ במחשב המרוחק היא ע"י הצמדת שם המשתמש ושם המחשב לשם הקובץ. אם שם המשתמש זהה בשני המחשבים אין חובה לציין.

דוגמאות: נניח שהמשתמש ben מחובר למחשב planet ויש לו שם משתמש זהה במחשב nova. הפקודה: `rcp nova:/tmp/test .`

תעתיק את הקובץ test הנמצא במדריך tmp/ במחשב nova לקובץ בשם זהה במחשב planet במדריך הנוכחי.

הפקודה: `rcp file nova:/tmp/new`

תעתיק את הקובץ file הנמצא במדריך הנוכחי, לקובץ new במדריך tmp/ במחשב nova. אם new הוא מדריך - יועתק הקובץ file למדריך זה בשמו הנוכחי.

הערה: בד"כ אין צורך להעתיק קבצים ממחשב אחד למשניהו מפני שקבצים הנמצאים ב-home directory שלך נגישים מכל מחשב UNIX כאילו היו מקומיים.

הפקודה telnet

תוכנית שירות כללית אחרת לביצוע login בין מחשבים מרוחקים התומכים בשירות זה היא: telnet יש לרשום: **<כתובת-מחשב> telnet** כאשר כתובת-מחשב היא כתובת-המחשב המרוחק אליו רוצים להתחבר. ניתן להתחבר למספר מחשבים במקביל, להשהות את חלקם ולבצע פעולות שונות המפורטות בדפי ה-man. הערה: הפקודה telnet מאפשרת התקשרות בין מחשבים מסוגים שונים לא רק בין מחשבי UNIX.

הפקודה ftp

הפקודה ftp מאפשרת העברה של קבצים ממחשב אחד לשני.

יש לרשום: **<כתובת-מחשב> ftp** להלן נכנה את המחשב בו בוצעה הפקודה ftp - 'המחשב המקומי' והמחשב אליו התקשרת - 'המחשב המרוחק'. לאחר הקשת כתובת-מחשב ftp תתבקש להכניס את שם המשתמש והסיסמא שלך במחשב המרוחק. יתקבל סימן מנחה >ftp המסמן לך שהינך בתוך ftp.

הפקודות העיקריות שתוכל לבצע:

שינוי מדריך במחשב המרוחק.	cd <שם-מדריך>
הצגת שם המדריך הנוכחי במחשב המרוחק.	pwd
שינוי שם מדריך במחשב המקומי.	lcd <שם-מדריך>
הצגת רשימת הקבצים במחשב המרוחק.	dir
העתקת קובץ מהמחשב המרוחק למחשב המקומי. הקובץ יהיה בעל אותו שם. ניתן לציין שם אחר כפרמטר שני של הפקודה.	get <שם-קובץ>
העתקת קובץ מהמחשב המקומי למחשב המרוחק. הקובץ יהיה בעל אותו שם. ניתן לציין שם אחר כפרמטר שני של הפקודה.	put <שם-קובץ>
קביעת סוג ההעברה ל-ascii, עבור קבצי טקסט.	ascii
קביעת סוג ההעברה לבינארית, עבור קבצים שאינם טקסט.	bin
דומה ל-get ו-put, אך עבור מספר קבצים. ניתן להשתמש בתוים מיוחדים בשמות קבצים. הקבצים המועתקים יהיו בעלי שמות זהים למקוריים.	mput / mget
מציג רשימת פקודות אפשריות. ניתן לציין שם של פקודה כפרמטר של help לקבלת תיאור מקוצר של הפקודה.	help
יציאה מ-ftp.	quit

יש מחשבים המאפשרים התקשרות ב-ftp אליהם גם אם אין לך שם משתמש במחשב זה. לשם כך קיים שם משתמש כללי בשם anonymous. מפעילים את הפקודה ftp כרגיל ומכניסים את שם המשתמש anonymous. הסיסמא היא כתובת ה-e-mail שלך. בד"כ הקבצים שתהיה מעוניין בהם במחשבים אלו יימצאו במדריך pub. הערה: הפקודה ftp מאפשרת העברת קבצים בין מחשבים מסוגים שונים לא רק בין מחשבי UNIX.

נספח א' X-WINDOWS מערכת חלונות

מערכת X-windows הינה ממשק גרפי למשתמש, המאפשר חלוקה של המסך למספר חלונות. חלון הוא שטח על המסך המציג יישום מסוים. יישומים שונים יכולים לרוץ בו-זמנית בחלונות שונים. למשל, בחלון אחד ניתן לערוך קובץ, בחלון שני להריץ תוכנית וחלון שלישי יכול לדמות מסוף של מחשב הקשור ברשת. המשתמש יכול לעבור בקלות מחלון אחד למשניהו. הממשק תומך בעבודה דרך הרשת וכך מאפשר לצג אחד להציג חלונות של תוכניות שרצות על מחשבים שונים המחוברים ברשת. תוכנה זו פותחה ב-MIT ומופצת על ידה באופן חופשי. ארגון ה-Software Open (OSF Foundation) שמטרתו ליצור תקנים בתיכנון מערכות תוכנה וחומרה הפך את X-windows לסטנדרט בתחום ממשקי המשתמש ל-UNIX.

מרכיבים של מערכת חלונות:

שרת X - (X server)
 לקוח X - (X client)
 פרוטוקול X
 "מנהל חלונות" (Windows Manager)

שרת X

השרת היא אפליקציה גרפית מיוחדת של מערכת X-windows המנהלת את התצוגה המקומית על המסך. השרת רץ על חומרה הקשורה לצג, bitmap לוח מקשים ועכבר ואחראי על הצגת החלונות, הפונטים, הצבעים, הקלט מלוח המקשים ומהעכבר ועוד. כל זאת לפי בקשות מהלקוח.

לקוח X

הלקוח היא תוכנית יישומית מבוססת על מערכת X-windows הרצה על המחשב ומנהלת דו-שיח עם שרת X - התוכנית יכולה לרוץ על מחשב השרת עצמו או על מחשב אחר. השרת מעביר את הקלט, מלוח המקשים, העכבר, ללקוח. הלקוח מבצע את החישובים ומחזיר לשרת את הנתונים להצגה על הצג.

דוגמאות לאפליקציות X -

xterm - פותח חלון המדמה מסוף.
 xclock - מציג שעון על המסך.
 xcalc - מציג מחשבון ומאפשר חישובים והצגת תוצאות.

פרוטוקול X

הקשר בין הלקוח לשרת הוא באמצעות פרוטוקול X - . זוהי מערכת מוגדרת של הודעות וכללים הפועלת ברשת או במחשב בודד. הקשר הוא דו-כיווני: הקלט של המשתמש מועבר מהשרת ללקוח ובקשות לציור על המסך מועברות מהלקוח לשרת.

מסוף X

מסוף X (X terminal) הוא מסוף ששרת X רץ עליו וכל ייעודו הוא להציג גרפיקה לפי בקשת לקוחות - הרצים במחשבים מרוחקים.

מנהל חלונות

מנהל חלונות הוא לקוח X מיוחד המפקח על כל החלונות המוצגים ע"י שרת X .
תפקידיו כוללים:

- פתיחה וסגירה של חלונות.
- הזזה, הגדלה והקטנה של חלון.
- בניית מסגרות סביב החלונות.
- הקטנת חלון לאיקון והגדלתו לחלון.
- שימוש בתפריטים.

עם כניסתך למערכת UNIX סביר להניח שמערכת החלונות ומנהל החלונות יופעלו אוטומטית בשבילך ותקבל חלון להכנסת פקודות.

שימוש בעכבר

העכבר הוא כלי המאפשר למשתמש אינטרקציה נוחה עם המסך. בעכבר יש בד"כ 3 כפתורים המכונים MB1 - השמאלי, MB2 - האמצעי ו- MB3 - הימני. בעזרת העכבר בוחרים פקודות מתפריטים, משנים גודל ומיקום חלון ועוד. הזזת העכבר גורמת להזזת מצביע על המסך. צורתו הגרפית של המצביע ניתנת לשינוי.

אם במהלך הזזת העכבר הגעת לקצה משטח העכבר, אבל המצביע עדיין לא הגיע למקום הרצוי, יש להרים את העכבר, ולהחזירו על המשטח במקום כלשהו ולהמשיך להניעו. בזמן הרמת העכבר המצביע לא יזוז ממקומו.

מושגים

הצבעה	הבאת המצביע למקום מסויים על המסך.
הקשה, "קליק"	לחיצה והרפיה על כפתור העכבר.
הקשה כפולה	שתי הקשות מהירות על אותו כפתור בעכבר.
לחיצה	החזקת כפתור העכבר לחוץ.
גרירה	הזזת המצביע תוך כדי לחיצה על כפתור העכבר.

חלון פעיל

החלון אליו מופנה הקלט מלוח המקשים הוא הנקרא חלון פעיל. ישנם מספר דרכים להפוך חלון לחלון פעיל בהתאם למנהל החלונות בו אתה משתמש ולתצורה - "קונפיגורציה" - שלו. לפני הקשת הפקודות בתוך חלון יש להפוך אותו לפעיל ע"י הצבעה בשטח החלון ובד"כ גם והקשה על MB1. במערכות מסוימות יש להצביע על שטח הכותרות בראש החלון ולהקיש על MB1. מנהל החלונות ידאג להבליט חלון זה ע"י שינוי צבע המסגרת וכדומה.

העתקת טקסט

הצבע על התו הראשון של הטקסט ולחץ MB1. גרור את המצביע עד סוף הטקסט המבוקש ושחרר MB1. הטקסט שמור במחסנית. הבא את המצביע למקום להעתקת הטקסט ולחץ MB2.

הזזת חלון

הצבע על אזור הכותרת בראש החלון. לחץ על MB1. גרור את החלון למיקומו החדש. הרפה מהלחיצה.

שינוי גודל חלון

הצבע על אחד מגבולות החלון. המצביע ישנה את צורתו. לחץ על MB1 וגרור את הגבול העליון או התחתון לשינוי אורך החלון, ואת הגבול הימני או השמאלי לשינוי רוחבו. לשינוי בו זמנית של אורך ורוחב החלון גרור את אחת מפינות של החלון. הגודל החדש של החלון נקבע עם שחרור הלחיצה על MB1.

איקון ICON

איקון הוא סמל מוקטן המייצג חלון מסויים. כאשר אין מקום על המסך, ניתן להקטין חלון לאיקון ע"י הצבעה על הכפתור השני, הפנימי, בפינה הימנית עליונה של החלון והקשה על MB1. האיקון נשמר ומוצג בפינת המסך. לשחזור החלון הצבע על האיקון והקשה כפולה על MB1.

שימוש בתפריטים

הקש על אחד הכפתורים של העכבר ברקע (הצבע על שם התפריט). לחץ על MB1. ותקבל תפריט. גרור את המצביע לפריט המבוקש, שחרר את MB1. גרירת המצביע מחוץ לתפריט והרפיית הלחיצה מבטלת את התפריט.

פונקציות ההזזה ושנוי גודל של חלונות ניתנות להפעלה גם דרך 'תפריט חלון' אותו מקבלים ע"י הקשה על הכפתור בפינה השמאלית עליונה של החלון.

דפדוף בחלון

הדפדוף נעשה באמצעות עמודת דפדוף בצד החלון, אשר הופעתה נקבעת ע"י תפריט. בחלון מסך לחץ על מקש ה - Ctrl השמאלי יחד עם MB2. בתפריט המתקבל בחר Bar Scroll, עמודת הדפדוף תופיע בצד. כאשר המצביע נמצא בעמודת הדפדוף הוא נהפך לצורת חץ כפול. לחיצה על MB3 בחלק העליון של עמודת הדפדוף תגרום להצגה של החלון שורה אחת קודם. לחיצה על MB1 תקדם את החלון בשורה אחת.

אם מורידים את החץ על עמודת הדפדוף, הלחיצות יגרמו לדפדוף של 2 או יותר שורות עד שבתחתית העמודה יגרמו לדפדוף של מסך מלא. דפדוף של מסך מלא ניתן לביצוע גם על ידי לחיצת מקשי prev ו- next או PageDown ו- PageUp במתאם למקלדת שלך.

פתיחה וסגירה של חלונות

לפתיחת חלון הצבע מחוץ לחלון ולחץ על MB1. מתקבל תפריט Menu Root בחר בפריט Window New. ניתן לפתוח חלון חדש גם עם הפקודה aixterm. לסגירת חלון בחלון המדמה מסוף, היציאה היא ע"י exit או logout. כמו כן, ניתן לבחור בפריט close של 'תפריט החלון'. בחלונות אחרים, סגירת החלון נעשית בד"כ באמצעות תפריט, תלוי ביישום.

סיום ה-session

בתפריט ה-Menu Root בחר בפריט Quit או exit.

שינוי סביבת עבודה של X

הקובץ Xdefaults

מערכת X-windows מאפשרת למשתמש שליטה על התצוגה כגון: בחירת פונטים, צבעים, גודל חלון ועוד. דרך אחת לעשות זאת היא ע"י אופציות בשורת הפקודה. דוגמא:

```
xclock -digital -bg white
```

דרך נוספת היא לרשום בקובץ בשם Xdefaults. במדריך הראשי את מאפייני האלמנטים

(resources) של אפליקציות X-.

דוגמא לקובץ Xdefaults

```
*Font : 12x24
xclock*analog: false
xclock*background: white
xclock*foreground: red
xterm*ScrollBar: true
```

קובץ זה קובע כי כל אפליקציות X תשתמש בפונט 12x24 (לא מומלץ).

כמו כן השעון יוצג בצורה אנלוגית, ולא דיגיטלית.

רקע השעון יהיה בצבע לבן.

המחשבון יוצג עם כיתוב בצבע אדום.

לחלון המדמה מסוף תהיה עמודת דפדוף.

בד"כ ב-man של כל אפליקציות X- מופיעה רשימת ה-resources האפשריים של אפליקציה זו.

הגדרת ה- Display

כאמור ב- X-windows ניתן להריץ תוכנית במחשב אחד ולקבל את התצוגה במחשב אחר. במקרה זה יש להגדיר את מסוף התצוגה באמצעות המשתנה הסביבתי DISPLAY. דוגמא: משתמש העובד על מסוף X- שכתובתו: 132.70.1.222, מבצע rlogin ל-planet. לפני הפעלת אפליקצית X- כלשהי חייב המשתמש לרשום תחילה:

```
setenv DISPLAY 132.70.1.222:0.0
```

הפקודה setenv נכונה עבור C-shell ו-Tcsh. עבור Shells אחרים ראה טבלה 1 השוואה בין Shells ב-UNIX. המערכת בד"כ מגדירה את ה-DISPLAY עבורך אך כדאי לוודא זאת ע'י

```
echo $DISPLAY
```

נספח ב' - תוספות למתקדמים

מסלול חיפוש (Path Search):

מרבית פקודות UNIX הן תוכניות הממוקמות במדריכים מסויימים במערכת הקבצים. כאשר אנו רושמים שם של פקודה, UNIX מחפש אותה, טוען אותה לזכרון ומבצע אותה. לכל משתמש מוגדרת רשימה של מדריכים, מסלול חיפוש, בהם יחול חיפוש של כל פקודה שתוקש על ידו. ניתן לעיין בהגדרת מסלול החיפוש באמצעות הפקודה:

```
printenv PATH
```

פלט הפקודה יראה בערך כך:

```
/bin:/usr/ucb:/usr/local/bin:.
```

חיפוש כל פקודה יתבצע תחילה במדריך /bin, לאחר מכן במדריך /usr/ucb, אח"כ במדריך /usr/local/bin ולבסוף במדריך הנוכחי.

ברגע שהפקודה אותה, החיפוש נפסק והפקודה מבוצעת. מכאן, שסדר המדריכים הרשומים ב-PATH הוא חשוב.

כדי לבצע פקודה שאינה נמצאת במסלול החיפוש, יש לציין את המסלול המלא שלה.

דוגמא: כדי לבצע פקודה בשם targil הנמצאת בתת מדריך בשם test במדריך הראשי שלך שאינו נמצא במסלול החיפוש יש לרשום:

```
~/test/targil
```

באופן דומה, אם קיימות שתי פקודות בעלות שם זהה, הנמצאות במדריכים שונים ורוצים לבצע אחת מהן, יש לרשום את המסלול המלא של הפקודה. כדי לברר איזה קובץ מתבצע כאשר מפעילים פקודה מסויימת

```
which <שם-פקודה>
```

רושמים:

```
rlogin which
```

דוגמא:

יציג את המסלול המלא של הפקודה rlogin, דהיינו: /usr/ucb/rlogin

כדי לקבוע מיקום של פקודה במערכת הקבצים רושמים: whereis גם את מיקום התיעוד של הפקודה.

```
whereis <שם-פקודה>
```

משתנים סביבתיים (Variables Environment)

בתוכניות ה-shells השונות מוגדרים משתנים סביבתיים, אשר תוכניות מסויימות משתמשות בהם, למשל הפקודה lp משתמשת במשתנה הסביבתי LPDEST שערכו הוא שם המדפסת.

להלן רשימה חלקית של משתנים סביבתיים ומשמעותם: שים לב: שמות המשתנים הם ב-UPPERCASE.

שם משתנה	משמעות ערכו
HOME	המדריך הראשי של המשתמש
PATH	רשימת מדריכים לחיפוש פקודות
SHELL	תוכנית ה-shell איתה עובד המשתמש
DISPLAY	הגדרת מסך להצגת גרפיקה מבוססת X. ראה "נספח א' X-WINDOWS מערכת חלונות"
TERM	הגדרת סוג מסוף עבור תוכניות שונות, כגון: vi
LPDEST	הגדרת מדפסת ברירת המחדל

הצגת רשימת כל המשתנים הסביבתיים המוגדרים מתאפשרת באמצעות הפקודה: `less | printenv` השתמשו בפילטר less היות והרשימה יכולה להיות ארוכה ממספר שורות החלון.

מתן ערך למשתנה סביבתי מתאפשר ע"י הפקודה: `setenv <שם-משתנה> <ערך>`

נכון עבור C-shell ו-Tcsh, עבור shells אחרים ראה טבלה 1 "השוואה בין Shells ב-UNIX". **דוגמא:**

```
setenv DISPLAY hds22
```

קבצים מיוחדים

הקובץ login.

הקובץ login. הוא קובץ המכיל אוסף של פקודות המתבצעות בתחילת כל כניסה למערכת. במרבית המערכות מוגדר לכל משתמש קובץ login. וניתן לראות אותו ברשימת הקבצים ע"י `ls -a`.
דוגמא לקובץ login.

```
# file to be executed at login
source /usr/skel/.login.src
msg n
who
```

השורה הראשונה בקובץ היא שורת הערה, ומסומנת ב-#. פקודת `source` מורה ל-csh, או ל-tsch לבצע את הפקודות הנמצאות בקובץ `/usr/skel/.login.src/` מומלץ מאוד לשאיר שורה זו על כנה. הפקודה `msg n` מציינת שהמשתמש מבקש למונע ממשתמשים אחרים לשלוח אליו הודעות באמצעות פקודת `write` או `talk`. הפקודה `who` תדפיס את פרטי המשתמשים המחוברים למערכת.

הקובץ cshrc.

הקובץ `cshrc`. מתבצע עם ההתחברות למערכת, לפני קובץ ה-`login` ועם פתיחת כל C-shell או `tcsch`. פקודות רבות פותחות shell והמשתמש אינו תמיד מודע לכך. ראה "Shells בהרחבה".
דוגמא לקובץ cshrc

```
# file to be executed at the start
# of every C-shell
source /usr/skel/.cshrc.src
alias up "cd" ..
alias = "ls -a"
alias ll "ls -l"
alias rm "rm -i"
set history=50
set prompt="ו"whoami"%ו"
```

בקובץ זה שתי השורות הראשונות הן הערות. בהמשך מוגדרים שמות קיצור לפקודות (`aliases`). הפקודה `set history` מציינת את מספר הפקודות לאיחסון. הפקודה `set prompt` קובעת את סימן המנחה. במקרה זה הסימן נקבע כשם המשתמש ובעקבותיו התו `%`. ה- " " סביב פקודה פרושו שהפקודה תתבצע והפלט שלה יוצב במקומה במחרוזת. זהו אחד משרותי ה-C-shell הרבים.

הערה: רצוי שלא לשים פקודות המדפיסות פלט על המסך בקובץ `cshrc`.

קבצים נוספים

קובץ המכיל פקודות לביצוע בתחילת כל כניסה ל-Bourne shell <code>.,sh</code>	<code>.profile</code>
קובץ המכיל פקודות לבצוע בתחילת כל כניסה ל-(shell Korn ksh)	<code>.kshrc</code>
קובץ להגדרת כינויים לנמענים עבור תוכנית ה-mail.	<code>.mailrc</code>
קובץ המכיל כתובת לניתוב מכתבים. קרא את הפלט של הפקודה <code>email news</code> .	<code>.forward</code>
קובץ המכיל הגדרות לשוני סביבת X כגון פונט הכיתוב, צבע חלון וכד'.	<code>.Xdefaults</code>
ראה נספח א' "X-WINDOWS מערכת חלונות".	

עבודות (JOBS)

תהליך (process)

תוכנית או פקודה הרצה בפועל במחשב UNIX נקראת תהליך (process). למשל, כאשר מריצים את הפקודה `finger`, זהו תהליך אחד. אם מריצים אותה שוב, זהו תהליך חדש. במערכת UNIX יכולים, בזמן נתון, להיות פעילים מספר תהליכים. לכל תהליך מוצמד מספר זיהוי (PID) יחודי. בכל ריצה של הפקודה `finger`, נוצר תהליך חדש עם מספר זיהוי שונה.

הפקודה בעזרתה מקבלים פרטים על התהליכים במערכת היא: `ps (status process)`. אם תפעיל את הפקודה, תקבל מידע על לפחות שני תהליכים. אחד מהם היא תוכנית ה-shell והשני היא פקודת ה-`ps` עצמה. הפלט יראה בערך כך:

```
PID      TTY  TIME CMD
15445    pts/1 0:12 tcsh
15463    pts/1 0:01 ps
```

PID הוא מספר זיהוי התהליך, TTY מגדיר את המסוף, TIME הוא זמן הביצוע של התהליך עד כה ו-CMD היא הפקודה שהפעילה את התהליך.

עבודה (JOB)

משימה המבוצעת ע"י המחשב ומורכבת מתהליך אחד או יותר, נקראת עבודה (job). למשל, `finger | sort`, מכיל תהליך אחד עבור `finger` ותהליך אחד עבור `sort`.

הרצת עבודות ב - background

נבחין בין בצוע תוכנית/פקודה תוך כדי המתנה לסיום בצועה (עד להופעת ה - prompt) המכונה בצוע ב-foreground (בצוע אינטראקטיבי), לבין בצוע תוכנית-פקודה ב-background (המכונה בצוע batch). לאחר שליחת תוכנית לבצוע ב-background (ברקע) יודפס מיד ה-prompt על המסך ותוכל להקיש פקודות נוספות בלי להמתין לסיום התוכנית. יתר על כן, תוכל לצאת מהמערכת (ע"י `logout`) התוכנית תמשיך לרוץ עד לסיומה, שמושי לתוכניות הדורשות זמן עיבוד ארוך.

כל תוכנית הנשלחת לבצוע ברקע מקבלת מספר job בנוסף למספר זיהוי תהליך (pid) שניהם מודפסים על המסך. הרצת תוכנית ברקע נעשית ע"י הוספת התו " & " בסוף הפקודה. דוגמא:

```
cc f1.c >& f1.error &
```

בדוגמא זו, הפקודה לבצוע קומפילציה של הקובץ `f1.c` נשלחה לבצוע ברקע. שים לב לניתוב פלט השגיאות לקובץ, במקום למסך. רצוי לעשות זאת כאשר מעוניינים לשמור את הודעות השגיאה של המהדר, למשל אחרי `logout`. עם הפעלת הפקודה נקבל על המסך הודעה בסגנון: `[1] 12312` פרושה שעבודה מספר 1 עם מספר זיהוי תהליך 14241 רצה ברקע.

סיום עבודה

מפסיקים תוכנית בעת ביצועה ב - foreground ע"י הקשת `c + ctrl`

```
kill -KILL %<job>
```

תוכנית הרצה ב-background מפסיקים ע"י:

```
kill -KILL <pid>
```

או

האופציה `-KILL` מציינת שיש להפסיק את העבודה באופן סופי ללא אפשרות מילוט ע"י `handler interrupt` שיתכן שתוכנת לתוכנית. בדרך כלל רצוי להשמיט את האופציה `-KILL` כדי לתת אפשרות לתהליך לסיים באופן נקי, ע"י סגירת קבצים, ערוצי תקשורת, מחיקת קבצים זמניים וכדומה, לפני מיתתו.

כאשר תגדל ותלמד את הקורס "מערכות הפעלה" יוארו עיניך בנושא זה. ואם כבר החלנו לדבר בשבחי קורס זה הרי שרצוי מאוד שתהיה מצוייד באחת מן הפקודות:

```
kill -TERM -1
```

```
kill -TERM 0
```

קודם שתתחיל לתכנת את התרגיל המסכם את הקורס, שאם לא כן עלול אתה לקבל את ההודעה "processes more no" המצינת את העובדה שעברת את מכסת התהליכים המותרת לך עקב bug

רציני בתוכנית אותה אתה כותב. צמד פקודות אלו ינפנף כלאחר יד את כל תהליך כולל את סביבת העבודה הנוכחית שלך.

השהיית עבודה

עבודה ב-foreground מושהית ע"י הקשה על `z + ctrl`
 עבודה ב-background משהים ע"י:
`stop %<job>`
`stop <pid>` או :

בצוע עבודה מושהית

ניתן להחזיר עבודה מושהית לבצוע ב- foreground או ב-background.
 מחזירים עבודה לבצוע ב-foreground ע"י: `% <job>` או `fg %<job>`
 מחזירים עבודה לבצוע ב-background ע"י: `& <job>` או `bg %<job>`
 טעות נפוצה, בפרט אצל משתמשי VMS, היא להפסיק את הרצת התוכנית ע"י `z + ctrl`, ולאחר מכן להתחיל אותה מחדש ע"י הקשה שמה וכן חוזר חלילה מכיוון ש- `z + ctrl`, רק משעה את התוכנית ולא מבטל אותה, הרי עשויים להיות לך עשרות תהליכים מושעים שאינך צריך, התופסים משאבי מחשב יקרים. בפרט לא נעים לעשות זאת עם `emacs`, `netcape`, `lisp` או `prolog` שהם זוללי זיכרון בצורה משוועת.

זכור !

א. `z + ctrl` לא מסיים את התהליך אלא רק משעה אותו.
 ב. החזר תהליך מושעה ע"י `%` או `fg`.

דו"ח על העבודות

כל עוד לא בצענו `exit` או `logout`, ניתן להשתמש בפקודה `jobs` למעקב אחרי תוכניות הרצות ברקע או מושעות.
 הפלט של הפקודה `jobs` מכיל את מספר ה-`job`, שם הפקודה שהפעילה את ה-`job` ומצב ה-`job` (running או stopped).
 במקרה שיצאת החוצה, כאשר היו עדיין עבודות ברקע, תוכל לעקוב אחריהן בכניסתך הבאה למערכת באמצעות הפקודה: `ps`
 בשונה מהפקודה `jobs`, הפקודה `ps` נותנת מידע גם על תהליכים שאינם שייכים ל-shell הנוכחי.
 השתמש בפקודה הר"מ להדפסת כל התהליכים הרצים במערכת. השתמשו בפילטר `less` היות ומספר התהליכים בד"כ עולה על מספר השורות במסך.

```
ps -ef | less
```

הערה: הפקודות שהוזכרו לעיל הן חלק מפקודות ה-shell ולכן כדי לקבל הסברים עליהן עליך להקיש:
`man <shell-הפקודת>`

שימוש בדיסקטים של DOS תחת סביבת UNIX

נפוץ היום להעביר מידע מהמחשב האישי לתחנת העבודה. מומלץ לבצע זאת באמצעות `ftp` מאחד המחשבים האישיים במחלקה לדוגמה: `ftp planet` ישלח או יקבל קבצים מתחנת העבודה `planet`.
 דרך נוספת היא להשתמש בכונן הדיסקטים של תחנת העבודה אם יש לך גישה פיזית אליה.
 על הדיסקט להיות מפורמט ורצוי שתהיה לו תוית (label).

ההנחיות הן כדלהלן:

1. בצע `login` לתחנת העבודה.
2. הכנס את הדיסקט לכונן.
3. הקש `volcheck`.
4. הדיסקט שלך עכשיו נגיש במדריך `/floppy/<label>`, כאשר `<label>` מציין את התוית של הדיסקט. אם התוית בלתי ידועה לך השתמש ב- `floppy` במקום התוית של הדיסקט.

5. תוכל להעתיק אליו קבצים ע"י :
`cp <שם-קובץ> /floppy/<label>`
 וכן להעתיק ממנו למדריך הנוכחי ע"י :
`cp /floppy/<label>/<שם-קובץ> .`
6. רשימת קבצים תקבל ע"י :
`ls -l /floppy/<label>`
 כדי להוציא את הדיסקט הקש:
`floppy eject`
7. סט פקודות חלופי המכונה mtools מכיל פקודות דומות המזכירות את הפקודות המקבילות

```
mcopy a:file .
mcopy file a:
mdir a:
```

הפקודה tar

פקודה זו יצרת ארכיב מרשימת הקבצים שתתן לה. הארכיב יכול להיות סרט מגנטי, דיסקט או, כפי שהוא בד"כ, סתם קובץ במדריך שלך. תחביר הפקודה:

```
tar <פרמטרים> <פקודות>
```

הפקודות העיקריות:

c	(create)	יצירת ארכיב חדש בסרט.
t	(table)	הצגת תוכן ארכיב על המסך.
f	(file)	הגדרת כוונן הסרט.
r		הוספה לארכיב קיים.
x	(extract)	שליפת קבצים מהארכיב לדיסק.
כ	(verbose)	הדפסת פרטים על הקבצים תוך כדי הקריאה-כתיבה.

דוגמאות:

יצירת ארכיב בשם myfiles.tar בדיסקט של DOS הנושא את התווית mydiskette שיכיל את כל הקבצים מהמדריך dir ומטה:

```
tar cvf /floppy/mydiskette/myfiles.tar dir
```

הצגת תוכן הארכיב על המסך:

```
tar tvf /floppy/mydiskette/myfiles.tar
```

להוסיף לארכיב קיים עץ המתחיל מהמדריך dir2:

```
tar rvf /floppy/mydiskette/myfiles.tar dir2
```

העתקה מהארכיב שבדיסקט לדיסק:

```
tar xvf /floppy/mysdiskette/myfiles.tar
```

שים לב שביצירת הארכיב סופקו השמות dir ו-dir2 יחסית למקום הנוכחי בעץ. אם בכתיבת הארכיב צויין מסלול מלא, name path full, הרי שבהעתקה חזרה לדיסק הקבצים יועתקו למסלול זה. אחרת הם יועתקו למסלול המצויין בארכיב, יחסית למדריך הנוכחי. אשר על כך רצוי ליצור ארכיבים ללא שמות מלאים, מדוע?

פתיחת ארכיבים דחופים

```
zcat file.tar.Z | tar xvpf -
gzip -d -c file.tar.gz | tar xvpf -
```

Shells בהרחבה

ה-Shell הינו ממשק בין המשתמש לבין מערכת ההפעלה UNIX. ה-Shell מציג את סימן המנחה קולט פקודה מהמשתמש, מפרש את הפרמטרים של הפקודה, מחפש את הפקודה במסלול החיפוש, מבצע אותה ולאחר ביצועה מדפיס שוב את סימן המנחה לקליטת הפקודה הבאה. קיימות מספר תוכניות: Shell (sh), Korn Shell (ksh), C shell (csh), T C shell (tcsh) ועוד. המשתמש יכול לבחור לעצמו כסביבת עבודה הנוחה לו את אחד ה-Shell הקיימים, להחליף את ה-Shell באחר בכל עת, או ליצור תחתיו Shell אחר הנקרא subshell. יצירת subshell תחת ה-Shell הנוכחי נעשית פשוט ע"י כתיבת שם ה-Shell, למשל ksh. יתכן מאוד שהמסך לא השתנה ויתכן שקבלת prompt שונה. בכל מקרה נפתח Shell חדש ותוכל לאשר זאת ע"י

הפקודה ps. באופן זה תוכל להמשיך וליצור Shell בתוך Shell. הפקודה לביטול Shell וחזרה Shell אחד אחורה היא exit. למה ליצור subshell? בדרך כלל, לא תצטרך לפתוח subshell. ואם באמת תצטרך אז כבר תידע למה ...

ה-Shell הוא גם שפת תכנות interpreter language command המאפשרת למשתמש לפתח כלים חדשים וליצור פקודות משלו על בסיס הפקודות הקיימות.

קבצים המכילים תוכניות הכתובות ב-Shell נקראים Scripts Shell. הפעלת Scripts Shell יוצרת subshell שהיא הסביבה בה מתבצע ה-Script.

C-Shell

בפרקים הקודמים דנו בתכונות ה-C-Shell:

- ניתוב קלט-פלט.
- הרכבת פקודות כאשר פלט של פקודה אחת משמש כקלט לפקודה הבאה אחריה (piping).
- תמיכה בתוים מיוחדים (cards wild) בשמות קבצים.
- הרצת תוכניות ברקע (background).
- שמירת פקודות ואפשרות לחזור עליהן (history).
- אפשרות להגדיר קיצורים או כינויים נוספים לפקודות (aliasing).

כשפת תכנות, יש ל-C-Shell אוסף של פקודות וכללי תחביר ולהלן עיקרם:

- משתנים
- כל המשתנים מוגדרים כמחרוזות.
- הפקודה set נותנת ערך למשתנה.

דוגמאות:

```
set name=Joe
set year=1993
```

ערך משתנה המכיל רווח או תוים מיוחדים אחרים יירשם בין גרשיים.

```
set fullname="Joe Willis"
```

הפקודה echo

מדפיסה מחרוזת, או ערך של משתנה. בהדפסת מחרוזת יש לבטל משמעות של תוים מיוחדים (כגון !,?) ע"י התו "\ ". בהדפסת ערך משתנה יש להקדים את שם המשתנה בתו \$.

דוגמאות:

```
echo hello, how are you /?
echo $name
echo $year
```

- משתנה יכול לקבל ערך שהוא הפלט של פקודה לאחר ביצועה. זאת ע"י שמוש בגרשים הפוכים (ליד מקש ~).

דוגמא: ערך n יהיה מספר המשתמשים המחוברים למערכת.

```
set n=`who | wc -l`
```

- משתנים המופיעים בין גרשיים כפולים " " יתורגמו לערכם.

דוגמא: תדפיס: 3 users are logged in

```
echo ` $n users are logged in `
```

- משתנים המופיעים בין הגרשיים ' ' לא יתורגמו לערכם.

דוגמא: תדפיס: number of users : \$n

```
echo "number of users:$n"
```

הגדרת רשימה - "מערך":

דוגמא

```
set a=(dogs cats cows)
echo $a[2]
```

ידפיס: cats

פעולות אריתמטיות

אריתמטיקה במספרים שלמים בלבד.
פקודה אריתמטית תתחיל ב - @. יש להקפיד לשים רווח לפני ואחרי כל אופרטור.

דוגמא

```
@ i = 7
@ j = 5 + $i * 16
@ i = $i + 1 @ זהה ל- i ++
@ i = $i + 7 @ זהה ל- i += 7
```

אופרטורים לפי סדר קדימות עולה:

&&		או-לוגי, וגם (and)-לוגי
==	!=	שווה ל... שונה מ...
>	<	<=
>	<	גדול מ- גדול או שווה ל- קטן מ- קטן או שווה ל -
+	-	חיבור חיסור
*	/	כפל, שארית (מודולו), חילוק
!		היפוך (not)
()		סוגריים.

משפטי תנאי - if

ביצוע פקודות אם התנאי מתקיים. אחרת ביצוע פקודות אלטרנטיביות.

פקודה (תנאי) if

או 1

```
if (תנאי) then
    פקודה 1
    פקודה 2
    . . .
endif
```

או 1

```
if (תנאי) then
    פקודה 1
    . . .
else
    פקודה 2
    . . .
endif
```

דוגמא

```
set n=`who | wc -l`
```

```
if ($n > 0) then
echo "$n users are logged in:
else
echo 'no one is logged in'
endif
```

פקודת while

ביצוע פקודות כל עוד התנאי מתקיים.

```
while תנאי
    1 פקודה
    2 פקודה
    . . .
end
```

דוגמא:

```
set i=5
while ($i > 0)
    date
    @ i --
end
```

יבצע הדפסת התאריך 5 פעמים.

פקודת foreach

ביצוע פקודות כמספר הערכים ברשימה . . . , ערך2, ערך1
(כאשר בכל סיבוב המשתנה var מקבל את הערך הבא ברשימה)

```
foreach var (ערך1, ערך2, ...)
    1 פקודה
    2 פקודה
    . . .
end
```

דוגמא:

```
foreach f(*)
mv $f $f.old
endif
```

מוסיף לכל הקבצים "old." לשמם.

Script Shell

את פקודות ה-shell ניתן לרשום מיד לאחר סימן המנחה המודפס ע"י ה-shell. הפקודות יתבצעו בזו אחר זו. אך לפעמים מעוניינים לרשום את הפקודות בקובץ shell script ולבצע אותו לפי הצורך.

יש להתחיל את הקובץ בשורה: `#!/bin/csh/ -f`
שורה זו יוצרת C-Shell חדש אשר יהיה הסביבה בה יתבצעו הפקודות בקובץ. האופציה `-f` מציינת לא להפעיל את קובץ האיתחול `cshrc`. הנמצא במדרוך הראשי שלך.
לפני ביצוע הקובץ יש להגדירו כבר-ביצוע (executable) ע"י: `chmod +x <שם-הקובץ>`
ביצוע הקובץ יעשה ע"י כתיבת שמו ליד הסימן המנחה בדומה לכל פקודת UNIX.

דוגמא:

```
#!/bin/csh/ -f
# Remove all empty files
set hour = `date`+%H ``
```

```

if ( $hour < 12 ) then
echo ` Good Morning \! `
else
echo ` Good Day !\ `
endif

echo ` I will remove all your empty files now. `
foreach f (*)
if (-z $f) then
    echo "-- Removing file $f --"
    rm $f
endif
end
end

```

הסבר :

הפקודה "date +H" מדפיסה את השעה, ראה `date man`. כאן פלט הפקודה הוכנס למשתנה `.hour`. אם השעה קטנה מ-12 התוכנית מברכת ב-`Morning Good` אחרת הברכה היא `Day Good`. התוכנית תמחק את כל הקבצים שגודלם 0. ה-`*` מתורגמת ע"י ה-`C-Shell` לרשימת כל הקבצים במדרוך הנוכחי. כל קובץ יושם במשתנה `f` בזה אחר זה. על כל קובץ תחול הבדיקה: "שם-קובץ -z". הבודקת אם גודל הקובץ 0. אם התנאי מתקיים הקובץ יימחק.

קיימים תנאים נוספים על קבצים כגון:

-f	קובץ רגיל
-d	מדריך
-e	קובץ קיים
-w	קובץ עם רשות כתיבה.

פרמטרים בשורת ההפעלה

Script Shell יכול לקבל פרמטרים במסגרת שורת ההפעלה (arguments command-line). הפרמטרים מוחזקים ב-`C-Shell` ברשימה בשם `argv` ומשתנה בשם `#argv` מכיל את מספר הפרמטרים.

דוגמא: נניח `shell-script` בשם `cline`

```

#!/bin/csh/ -f
echo There are $#argv arguments
echo The first argument is $argv[1]
echo The last argument is $argv[$#argv]
echo The second through fourth arguments are $argv[2-4]

```

הפעלת ה-`shell-script` ע"י:

```
cline the big guys always win
```

הפלט:

```

There are 5 arguments
The first argument is the
The last argument is win
The second through fourth arguments are big guys always

```

Tcsh

- ה- Tcsh הינו הרחבה של ה- C-Shell. השרותים הנוספים שהוא מאפשר כוללים:
- השלמת שמות פקודות-קבצים.
 - תיקון טעויות כתיב (spelling)
 - דפדוף במאגר הפקודות ועריכתן.

השלמת שמות פקודות-קבצים

כאשר מקישים פקודה, אין צורך להקיש את שמה המלא. ניתן להסתפק בהקשה חלקית של שם הפקודה. ה Tcsh ישלים את השם המלא עם הקשה על המקש TAB. אם הקיצור מתאים ליותר מאשר שם אחד, ההשלמה תתבצע עד לנקודת הזוהר המקסימלית. במקרה זה תקבל רשימה של כל השמות המתאימים לשם החלקי שהוקש. שירות זה נכון גם עבור שמות קבצים, פרמטרים של חלק מהפקודות ועוד.

תיקון טעויות כתיב (spelling)

Tcsh מתקן טעויות בכתיב של שם קובץ, פקודה או שם של משתמש. לאחר הקשת המלה לחץ Esc-S לבדיקה ותיקון תקינות הכתיב. תקון שורת פקודה שלמה מתאפשר ע"י הקשה על \$-Esc.

דפדוף במאגר הפקודות ועריכתו

כאמור ה- C-shell וה- Tcsh מנהלים מאגר של פקודות שהוקשו ע"י המשתמש. ב- Tcsh ניתן לדפדף במאגר זה בעזרת מקשי החיצים. ניתן גם לערוך את שורת הפקודה ואף להשתמש בחלק מהפקודות של emacs כמו a + ctrl לתחילת שורה ו- e + ctrl לסוף שורה.

טבלה 1. השוואה בין Shells ב- UNIX

ksh	tosh	osh	sh	תכונה
\$	>	%	\$	סימן מנחה (ברירת מחדל)
VAR = <value> export VAR	setenv VAR <value>	setenv VAR <value>	VAR = <value> export VAR	הגדרת משתנה סביבתי
.profile	.oshrc או .toshrc .login	.oshrc .login	.profile	קובץ אתחול
✓	✓	✓		מנגנון history ב- session נוכחי
✓	✓			חזרה על פקודות ועריכתן
✓	✓	✓		מנגנון history בין sessions
✓	✓	✓		ניהול הרצת תוכניות ברקע
alias <shortcut> = command	alias <shortcut> command	alias <shortcut> command		הגדרת קיצורים לפק' (aliasing)
	✓	✓		השלמת שמות קבצים ופקודות
	✓			תיקון טעות בכתיב שם קובץ או פקודה
2 >	>&	>&	2 >	סימן לניתוב פלט שגיאה

נספח ג - רשימת פקודות

פקודות המסומנות ע"י * אינן מופיעות בחוברת.
 הפעל: <שם-פקודה> man לקבלת עזרה על פקודות אלו.

שם הפקודה	תאור מקוצר
alias	שמות קיצור לפקודות
ar ,ld	טיפול בספריות
bg, fg% , stop, jobs	שליטה בעבודות
cat	עיון בקובץ
cb	מייפה תוכנית ב-C
cc, gcc g++, as	מהדרים
cd	שינוי מדריך
chmod	שינוי הרשאות קריאה - כתיבה לקובץ
clean.trashcan	מחיקה סופית של קבצים "מחוקים"
cmp *	השוואת 2 קבצים לפי בתים
compress, uncompress	דחיסת קובץ ופריסתו
cp	העתקת קבצים
date	הצגת תאריך ושעה
dbx, gdb mxgdb	ניפוי שגיאות עבור תוכניות ב- Fortran,Pascal C,
du *	סיכום שימוש בדיסק
echo	הדפסת הודעה ב-shell
elm	קבלת ומשלוח דואר
find	מציאת קבצים בתוך מערכת הקבצים
finger	הצגת מידע לגבי המשתמש
ftp, xftp	העתקת קבצים בין מחשבים מרוחקים
grep *	ביצוע חיפוש בקבצים עבור ביטויים
(egrep, fgrep)	
groups *	הצגת קבצי הרשאה להם משתייך המשתמש
gzip	דחיסת/פריסת קובץ בפורמט zip GNU
head	הצגת השורות הראשונות בקובץ (BSD)
history	הצגת מאגר פקודות
if, while foreach	משפטי בקרה
kill	סיום עבודה של תהליך
lint	בדיקת תקינות תוכנית C
lpq	הצגת התור למדפסת ומצבו
lpr	הדפסת קובץ במדפסת
lprm	מחיקת קובץ מתור המדפסת
ls	הצגת רשימת קבצים

קבלת ומשלוח דואר	mail	
הצגת עזרה על פקודות	man, news, whatis	
העתקת קבצים מ לדיסקט	mcopy	
הצגת רשימת הקבצים שבדיסקט	mdir	
מתן או אי-מתן הרשאה לקבלת הודעות	mesg	
יצירת מדריך	mkdir	
הצגת קובץ על המסך	more, cat, less	
שינוי שם קובץ - הזזת קובץ	mv	
מנהלי חלונות נוסח Motif	mwm	*
	fvwm	
שינוי קדימות job	nice	*
מנהלי חלונות נוסח olwmm Openlook	olwm	*
החלפת סיסמא	passwd	
תוכנית דואר ב "מצג מלא"	pine	
הצגת ערכי המשתנים הסביבתיים	printenv	
הצגת מצב התהליך	ps	
הצגת המדריך הנוכחי	pwd	
העתקת קבצים בין מחשבים מרוחקים, בין מחשבי UNIX בלבד	rcp	
התחברות למחשב מרוחק בין מחשבי UNIX בלבד	rlogin	
מחיקת קובץ	rm, unrm	
מחיקת מדריך	rmdir	
בצוע פקודות במחשב UNIX מרוחק	rsh	
השמת ערך למשתנים של ה-shell	set, setenv	
תוכניות shell	sh, csh, tcsh, ksh	
מיון ו/או חבור קבצים	sort	
בדיקת כתיב	spell	
השהיית עבודה	stop	
בצוע אופציות עבור צגים	stty	*
הצגת סוף קובץ	tail	*
ניהול דו-שיח עם משתמש אחר	talk	
ניהול ארכיבים	tar	
התחברות למחשב מרוחק	telnet	
processor עבור קבצי TeX	tex	*
תרגום תווים	tr	*
postprocessor עבור קבצי טקסט	troff (nroff)	*
הצגת שם הצג שלך	tty	*
עריכת קובץ	vi, emacs	
הכנסה והוצאת דיסקט	volcheck eject	
הצגת מספר השורות, המילים והבתים בקובץ	wc	*

חיפוש תוכניות במערכת הקבצים	whereis	
הצגת מסלול מלא של פקודה	which	
הצגת המשתמשים במערכת	who	*
שליחת הודעה למשתמש אחר	write	*
מציג שעון בסביבת חלונות	xclock	
מדמה מסוף לסביבת חלונות	xterm	
מעדכן את שרת ה-X ב-X-resources חדשים	xrdb	*

עשה ואל תעשה (DON'TS & DO'S)

כבר הוזכר מראש שה- login שלך הוא אישי ולא ניתן להעברה. עליך לשמור על סיסמתך באופן אישי ולא לאפשר לאחרים לעבוד תחת ה- login שלך. לכן עליך למשל לדאוג ליציאתך המסודרת (logout) מהמערכת קודם שתעזוב את המסוף עליו עבדת.

כמו כן עליך להגן על קבציך, דהיינו לא לתת להם הרשאות גישה מיותרות.

כמו כן אסור לך לקרוא קובץ דואר וכו' של אחרים ללא רשותם גם אם המחשב מאפשר זאת. אין צורך לומר שאסור לגרום נזק כל שהוא במערכת.

בנוסף אסור לך לשמור ו/או להביא מהרשת קבצים בעלי תוכן שאינו מתאים לרוח המוסד בו אתה לומד ("ודי לחכימא ברמיזא").

עליך גם להתחשב במגבלות הקיימות לגבי משאבי מחשב ולא להשתמש ביותר מדי נפח דיסק, זכרון מחשב, זמן CPU וכדומה. במידה והינך זקוק למשאבים למעלה מן הממוצע עליך לפנות בבקשה מיוחדת לצוות התמיכה כדי לתאם זאת מראש.

השימוש במשאבי המחשב כולל השימוש בשירותי הרשת הבין-אוניברסטית מיועד אך ורק לצרכי לימודיך האקדמיים. אין להשתמש במשאבים הנ"ל למטרות אחרות.

אי עמידה בכל הכללים האמורים לעיל יכולה לגרום לשלילת זכותך ל-`login` ואף להעמדה לפני ועדת משמעת של האוניברסיטה במידת הצורך.

מודעה רבתי:

המחלקה אינה יכולה להתחייב לגבי פרטיות המידע במחשביה.

המחלקה, באמצעות האחראים על המערכת, שומרת לעצמה את הזכות לבדוק את המידע הממוחשב שבתחומה כולל דואר, קבצים, מידע הזורם ברשת וכו' לצורך הטיפול במערכת והגנתה וזאת עפ"י שיקול דעתה הבלעדי.