

מסמך זה נכתב על ידי newbie בשביל newbies מתוך עניין, אמונה ואהבה ללינוקס וכמובן עקב מחסור רב באינפורמציה כתובה עיברית ברורה פשוטה ועיניינית לגבי מערכת ההפעלה לינוקס. זהו תרגום חלקי של Inag. את המקור השלם ניתן להשיג ב:

**<http://sunsite.dk/linux-newbie>**

התרגום נעשה בהסכמתם של המחברים המקוריים:

Copyright (c) <1999-2002> by Peter and Stan Klimas.

ותחת ה- Open Publication License v1.0, 8 עם השינויים שמצוינים ב Inag\_licence

([http://sunsite.dk/linux-newbie/Inag\\_licence.html](http://sunsite.dk/linux-newbie/Inag_licence.html))

תיקונים ביקורת וכו' ייתקבלו בברכה ב [etay218@bezeqint.net](mailto:etay218@bezeqint.net)

כל עוד זה תלוי במתרגם, קוראי מסמך זה יכולים לעשות בו ככל העולה ברוחם.

אלעד איתי

תודה מיוחדת לאמיר טל מ [www.whatsup.org.il](http://www.whatsup.org.il) , לכל חברי [www.penguin.org.il](http://www.penguin.org.il) , ולרן צחור-מנהל פורום לינוקס בynet- שאם לא היה עונה לי על שאלה ששאלתי, ספק אם הייתי כותב שורות אלו.

*"software is like sex its better when its free"*

תוכן;

## חלק 0. לאלה שעדיין לא החליטו

### תוכן החלק הזה:

- 0.1 ועכשיו ברצינות, למה לינוקס? [עמ' 2]
- 0.2 האם לינוקס בשבילי? [עמ' 2]
- 0.3 לינוקס קשה למתחילים [עמ' 2]
- 0.4 מה הוא הערך המוסף של לינוקס? [עמ' 3]
- 0.5 מה הם ההבדלים בין לינוקס ליוניקס? [עמ' 5]
- 0.6 מה הם ההבדלים בין לינוקס ומייקרוסופט ווינדוס? [עמ' 5]
- 0.7 אני לא מאמין בתוכנה חינם וכו' [עמ' 6]
- 0.8 אין דבר כזה ארוחות חינם [עמ' 6]
- 0.9 אני זקוק לאבטחה ברמה גבוהה, עם תוכנות מסחריות, אני יכול לתבוע אותם אם משהו מתפקשש [עמ' 7]
- 0.10 אני צריך סטנדרטים. תאגידי תוכנה גדולים (מייקרוסופט) מספקים סטנדרטים [עמ' 7]
- 0.11 הפופולריות של מייקרוסופט ווינדוס מבטיחה שהיא "כאן כדי להישאר" [עמ' 9]
- 0.12 אבל לינוקס מסתעפת לכמה מערכות שונות [עמ' 9]
- 0.13 לינוקס זה קאלט [עמ' 9]
- 0.14 העלות הכוללת הסופית של הבעלות על לינוקס גבוהה [עמ' 10]
- 0.15 לינוקס חולמת בהקיץ; עסקים זה מה ששולט היום בעולם [עמ' 11]
- 0.16 לינוקס זה חרטא וכו' [עמ' 11]

## חלק 2. מקורות לקריאה, עזרה וכמה לינקים

### תוכן החלק הזה:

- 2.1 חומר קריאה לגבי לינוקס? [עמ' 36]
- 2.2 האם ישנה פקודת עזרה? [עמ' 37]
- 2.3 איזה שהוא מילון לגבי מונחים? [עמ' 38]

2.4 חיפוש ברשת [עמ' 38]

2.5 קבוצות דיון [עמ' 39]

2.6 לינקים ללינוקס באינטרנט? [עמ' 40]

2.7 קוד מקור--המקור האולטימטיבי [עמ' 43]

### **חלק 3.1 יסודות**

#### **תוכן החלק הזה:**

3.1.1 שמות קבצים. [עמ' 46]

3.1.2 למה משמשות המחיצות השונות? [עמ' 49]

3.1.3 איך אני מריץ תוכנית? [עמ' 52]

3.1.4 כיצד אני יכול לשנות את התיב? [עמ' 54]

3.1.5 איך אני יכול לכבות את המחשב? [עמ' 55]

3.1.6 איך אני מתמודד עם תוכנה תקוע? [עמ' 57]

3.1.7 אפשרויות פקודה. [עמ' 61]

### **3.2 משתמשים, סיסמאות, הרשאות קבצים, ואבטחה**

3.2.1 ספריות בית, root הוספת משתמש [עמ' 62]

3.2.2 קצת על אבטחת סיסמא [עמ' 63]

3.2.3 שכחתי את הסיסמא של root [עמ' 65]

3.2.4 שכחתי את סיסמת המשתמש שלי [עמ' 68]

3.2.5 ביטול או הסרה של חשבון משתמש [עמ' 68]

3.2.6 יש לי בעייה עם הרשאות קבצים, הרשאות קבצים ובעלות על קבצים. איך זה עובד?  
[עמ' 69]

3.2.7 נגן mp3 שלי נחנק. הצליל לא נשמע טוב (איך להגדיר suid) [עמ' 73]

### **3.3 תזמון משימות עם "&", "at", "batch", ו-"cron"**

3.3.1 איך אני מבצע פקודה ברקע? [עמ' 75]

3.3.2 איך אני מבצע פקודה בזמן מסויים (על ידי שימוש ב"at" או "batch")? [עמ' 75]

3.3.3 איך אני מגדיר את cron? [עמ' 77]

## shell 3.4

- 3.4.1 מה זה shell (מעטפת) והאם אני רוצה להשתמש באחרת? [עמ' 79]
- 3.4.2 כיצד אני מבצע התאמה אישית למעטפת שלי? [עמ' 80]
- 3.4.3 צבע בטרמינל טקסט [עמ' 81]
- 3.4.4 כיצד אני מדפיס סימנים בקונסול, באפליקציות מצב טקסט, וב-X? [עמ' 82]
- 3.4.5 איך אני כותב סקריפט מעטפת פשוט? [עמ' 83]
- 3.4.6 המשמעות של ציטוטים [עמ' 85]
- 3.4.7 הפניה מחודשת של פלט/קלט [עמ' 85]
- 3.4.8, תוי מעטפת מיוחדים (metachacters) [עמ' 88]

## 3.5 התקנת חבילה ומנהל החבילות rpm?

- 3.5.1 איך אני מתקין תוכנית שהורדתי מאינטרנט? [עמ' 90]

### חלק 5 : קיצורי דרך ופקודות

#### תוכן החלק הזה:

- 5.1 קיצורי דרך חיוניים ופקודות שפיות [עמ' 95]
- 5.2 פקודות עזרה [עמ' 99]
- 5.3 מידע מערכת [עמ' 100]
- 5.4 פעולות בסיסיות [עמ' 104]
- 5.5 ניהול קבצים [עמ' 105]
- 5.6 צפייה ועריכת קבצים [עמ' 107]
- 5.7 מציאת קבצים [עמ' 108]
- 5.8 היסודות של X-windows [עמ' 109]
- 5.9 אפליקציות רשת [עמ' 110]
- 5.10 דחיסה ופריסה של קבצים [עמ' 113]
- 5.11 שליטה על תהליכים [עמ' 114]

5.12 פקודות ניהול בסיסיות [עמ' 116]

5.13 כלי ניהול דיסק [עמ' 122]

5.14 ניהול חשבונות משתמשים והרשאות קבצים [עמ' 124]

5.15 התקנת תוכנה [עמ' 127]

5.16 גישה לכווננים/מחיצות [עמ' 129]

5.17 כלי ניהול רשת [עמ' 129]

5.18 פקודות שקשורות למוזיקה [עמ' 131]

5.19 פקודות שקשורות לגרפיקה [עמ' 133]

5.20 משחקים קטנים [עמ' 135]

## חלק 0. לאלה שעדיין לא החליטו

### תוכן החלק הזה:

- 0.1 ועכשיו ברצינות, למה לינוקס? [עמ' 2]
- 0.2 האם לינוקס בשבילי? [עמ' 2]
- 0.3 לינוקס קשה למתחילים [עמ' 2]
- 0.4 מה הוא הערך המוסף של לינוקס? [עמ' 3]
- 0.5 מה הם ההבדלים בין לינוקס ליוניקס? [עמ' 5]
- 0.6 מה הם ההבדלים בין לינוקס ומייקרוסופט ווינדוס? [עמ' 5]
- 0.7 אני לא מאמין בתוכנה חינם וכו' [עמ' 6]
- 0.8 אין דבר כזה ארוחות חינם [עמ' 6]
- 0.9 אני זקוק לאבטחה ברמה גבוהה, עם תוכנות מסחריות, אני יכול לתבוע אותם אם משהו מתפקשש [עמ' 7]
- 0.10 אני צריך סטנדרטים. תאגידי תוכנה גדולים (מייקרוסופט) מספקים סטנדרטים [עמ' 7]
- 0.11 הפופולריות של מייקרוסופט ווינדוס מבטיחה שהיא "כאן כדי להישאר" [עמ' 9]
- 0.12 אבל לינוקס מסתעפת לכמה מערכות שונות [עמ' 9]
- 0.13 לינוקס זה קאלט [עמ' 9]
- 0.14 העלות הכוללת הסופית של הבעלות על לינוקס גבוהה [עמ' 10]
- 0.15 לינוקס חולמת בהקיץ; עסקים זה מה ששולט היום בעולם [עמ' 11]
- 0.16 לינוקס זה חרטא וכו' [עמ' 11]

## 0.1 ועכשיו ברצינות, למה לינוקס?

אם אתה באמת נהנה לעבוד עם מחשבים, לינוקס היא מערכת ההפעלה של חלומותיך. היא מהנה הרבה יותר מכל מערכת הפעלה אחרת בסביבה. בכל מקרה, הסיבה שבגללה לינוקס באמת מהפכנית היא בגלל שהיא תוכנה פתוחה. המדע והטכנולוגיה שלנו עובדים הודות לזמינות החופשית של המידע והאפשרות לצפות האחד בעבודתו של השני. האם היית טס במטוס שנבנה על בסיס סוד מדעי ומידע מוסתר של איש פרט למתכנן המטוס אסור להסתכל בקרבו? ובכן מדוע שתבטח בתוכנת מחשב שמכילה חלקים ואלגוריתמים סודיים? מערכת ההפעלה לינוקס בעלת הקוד הפתוח אידיאלית לאפליקציות חשובות וקריטיות--יכולות האבטחה והעוצמה שלה מבוססות על פתרונות חזקים שכל אחד יכול להסתכל בהם, לבקר או לשפר אותם. זוהי ההטמעה של הגישה המדעית אל תוך המיחשוב.

ייצור פרסות סוסים, זכוכית טובה, או מדידת זמן היו פעם סודות מסחריים שמורים היטב. המדע והטכנולוגיה התפוצצו לפני 500 שנה הודות לשיתוף הידע על ידי הדפוס. בימים הראשונים של הדפוס רבים מאלו שהעזו לשתף נרדפו בגין חשיפת סודות מסחריים. לינוקס לעידן המחשב היא מה שגוטנברג היה לכתיבה. בתקווה שאיש לא יירדף הפעם. לינוקס אכן מתנגשת עם אלו שטוענים לבעלות של מידע, שמנסים להחזיר את הזמן 500 שנים אחורה.

## 0.2 האם לינוקס בשבילי?

רק אתה יכול לענות על השאלה הזו. לינוקס היא מערכת הפעלה דמויית יוניקס, בוגרת, עוצמתית מאובטחת ובעל יכולות רבות. העוצמה וריבוי היכולות שלה באות יחד עם מחיר--ייתכן ותצטרך לדעת קצת קורא וכתוב במחשבים כדי להקים את לינוקס ולתחזק אותה. באופן יחסי לינוקס קלה לשימוש באופן כזה שברגע שהתקנת את האפליקציות ואת מערכת ההפעלה כמו שצריך גם אמא שלך תוכל להשתמש בה, במידה והכנת בשבילה חשבון גרפי והנחת את האייקונים/תפריטים המתאימים על גבי שולחן העבודה שלה. לינוקס מאובטחת, ככה שאמא שלך לא תוכל לפגוע במערכת לא משנה מה היא תנסה לעשות, אלא אם כן היא תשתמש בפטיש.

לינוקס די שונה ממייקרוסופט ווינדוס, אז אל תצפה שהיא תהיה ברורה וקלה לך אם אתה מכיר קצת ווינדוס. ייתכן ותצטרך ללמוד קצת. מצד שני אם אתה מגיע מיוניקס, לינוקס תהיה קלה בשבילך. אם אינך יודע הרבה על מחשבים או שאתה לא נהנה מהם, רוב הסיכויים שאדמינסטריית לינוקס אינה בשבילך.

אם אתה לא מכיר את החומרה שלך התקנת לינוקס יכולה להיות אתגר.

## 0.3 לינוקס קשה למתחילים

יש בזה משהו. אבל השאלה האמיתית היא: האם אתה רוצה ללמוד אותה? לאיש מהמחברים (או מהמתרגמים) אין רקע במדעי המחשב, ועדיין, אנחנו משתמשים בלינוקס כל יום ואנחנו אוהבים את זה.

## 0.4 מהו הערך המוסף של לינוקס?

לינוקס יכולה לתת לך:

\*סביבת משתמש מודרנית, יציבה מאוד, בעלת יכולת של ריבוי משימות על גבי חומרת מחשב לא יקרה, ללא (או כמעט ללא) תשלום תמורת התוכנה. לינוקס היא פלטפורמה עשירה ועוצמתית--אל תחשוב שהיא מערכת הפעלה לעניים. ללינוקס מהקופסה יש יכולות כמו למייקרוסופט ווינדוס NT עם \$5000 תוספות תוכנה. היא מאוד יציבה, ודורשת חומרה פחות חזקה מאשר ווינדוס למשימות זהות.

\*עוצמת עיבוד אולטימטיבית, ניידות, וגמישות. אשכול מחשבים של לינוקס הביס (אפריל 1999) מחשב על בבנצ' מרק סטנדרטי.

\*לינוקס מאוד "מסטונדרטת" -- היא POSIX, תואמת UNIX. כן, לינוקס היא צאצא משובח של יוניקס. המילה יוניקס לא נאמרת יחד עם לינוקס מכיוון ש"UNIX", זה סימן רשום.

\*לינוקס פופולרית מאוד על גבי מחשבים מבוססי אינטל (מחיר), אבל רצה היטב גם על מספר רב של פלטפורמות חומרה, ממחשבים דמויי צעצוע עד למחשבי מיינפריים. הפצה אחת (דבין-DEBIAN) מבטאת את הרעיון בצורה הבאה:

"לינוקס, מערכת ההפעלה האוניברסלית"

\*פלטפורמת לימוד גדולה באמת. אם אתה הורה, אתה אמור לשמוח אם בנך/בתך משתמשים בלינוקס. היא/הוא באמת יילמדו משהו בעל ערך מוסף שילווח אותם לאורך זמן. אם אתה מורה, כדאי שתשקול התקנת לינוקס בבית הספר שלך. "זהו אכן עולם מוזר בו מחנכים צריכים להשתכנע לגבי העובדה ששיתוף מידע, בניגוד לחסימת גישה אל מידע הוא אכן דבר חיובי"

<http://edge-op.org/grouch/schools.html>

אתה בוחר בלינוקס אם אתה מעוניין לספק השכלה ולא אימון או הכשרה. כמה שהאוניברסיטה טובה יותר, ככה הסיכוי שמחלקת המחשבים שלה משתמשת בלינוקס ללימוד. תחת לינוקס אתה יכול למשל, להתחיל לשנות לערוך ולקמפל בעצמך אפליקציית גיליון נתונים מורכבת בדיוק כמו מייקרוסופט אקסל. לינוקס מניחה אותך ישר בחזית בטכנולוגיה, בניהול פרוייקטים, ומתודולוגיית מדעים (מורים רבים לא ישתמשו בלינוקס בבית הספר מכיוון שההשכלה שלהם בתחום המחשבים לוקה בחסר. זה לפחות מה שאני רואה)

\*יכולות רשת מעולות בנויות אל תוך מערכת ההפעלה שלך. אתה חושב שאין לך צורך ברשת?

ברגע שתתנסה ברשת ביתית לא תוכל לחיות בלעדיה. תוכל לשתף כוננים בין מחשבים כרטיסי קול מודמים, מדפסות וכו'. תוכל לגלוש באינטרנט דרך שני מחשבים באותו הזמן תוך שימוש בחיבור יחיד לאינטרנט. תוכל לשחק משחקים עם הבן שלך. ואפילו ה-386 הישן שלך עם ווינדוס 3.11 יקום לתחיה, כשתחבר אותו לשרת לינוקס פנטיום שלך. כל תוכנות הרשת הדרושות מגיעות עם לינוקס כסטנדרט באופן חופשי, רק התקנה והקמה דרושות, וזו אינה תוכנה סוג ב'--זו בדיוק אותה תוכנה שמריצה את רוב האינטרנט (שרת הווב אפאצ'י - APACHE מריץ יותר מ-50% מכל שרתי הווב באינטרנט וסנדמייל SENDMAIL נוגע בכ-70% מכל הדואר האלקטרוני).

את ההנאה שברשת ביתית יכולתי לגלות רק על ידי שימוש בלינוקס.



\*קישורים לרשתות מייקרוסופט, נובל, ואפל. קריאה/כתיבה אל תוך פורמטי דיסק של מייקרוסופט ופורמטים אחרים. זה כולל שימוש שקוף במידע שמאוחסן על מחיצות ווינדוס בכונן שלך.

\*מאות אפליקציות, חופשיות, במגוון תחומים בחינם.

\*מאות רבות של אפליקציות מיוחדות/מתמחות שנבנו על ידי חוקרים ברחבי העולם (אסטרונומיה, טכנולוגיית מידע, כימיה, פיסיקה, הנדסה, בלשנות, ביולוגיה...) בתחומים רבים לינוקס נראית כמערכת ההפעלה היחידה שקיימת (נסה לברר מה חברך האסטרונום מריץ על המחשב שלו). התוכנה בקטגורייה זו אינה כל כך קלה לשימוש, אבל אם אתה מעוניין בעוצמה, זוהי התוכנה הטובה ביותר שיש לאנושות בתחומים אלו. יש לך ספקות? תן מבט ב: <http://SAL.KachinaTech.COM/Z/2/index.html> אם אתה מעוניין בדוגמאות.

\*אלפי כלים, יישומים, ותוכניות קטנות. "קטן זה יפה" הולך טוב עם הפלוסופיה של לינוקס. הכלים והתוכניות הקטנות של לינוקס עובדות לעיתים אחת מאחורי השניה כדי לבצע משימות מורכבות.

\*גירסאות של כל התוכניות המסחריות הטובות ביותר כולל כל מסדי הנתונים הגדולים (אורקל וסיבייס אבל לא מייקרוסופט). רבות מתוכניות אלו (כולן?) מוצעות באופן חופשי למפתחים ולשימוש אישי.

\*פלטפורמת פיתוח מלאכת מחשבת עם שפות תיכנות הטובות מסוגן וכלים שמגיעים חופשי עם מערכת ההפעלה. גישה אל קוד המקור של מערכת ההפעלה, במידה ואתה זקוק לו, גם הוא מגיע חינם. הקומפילר של שפת C שמגיעה כסטנדרט עם לינוקס יכול לקמפל קוד לרוח הפלטפורמות כנראה, יותר מכל קומפילר אחר עליי האדמות. PERL, PYTHON, GUILLE, TCL, RUBY scripting "SHELL" עוצמתי ואפילו אסמבלר מגיע כסטנדרט עם לינוקס.

\*חופש יחסי מוורוסים, סוסים טרויינים, "פיצ'רים" של יצרני תוכנה, חדירה אל הפרטיות, שידרוגים כפויים, פורמטי קבצים סגורים, סכמות רישוי ושיווק, רישום מוצר, מחירים גבוהים, ופירטיות. איך זה נשמע לך?

\*פלטפורמת ההפעלה שמובטח שהיא "כאן כדי להישאר". מכיון שלינוקס אינה בבעלות של איש, היא אינה יכולה לצאת אל מחוץ לעסקים. הרשיון הציבורי הכללי ה-GPL= GENERAL PUBLIC LICENSE-מבטיח שפיתוח/תחזוקה ייסופקו כל עוד ישנם משתמשי לינוקס. ישנו מספר גדול של משתמשי לינוקס משכילים מאוד ועשרות אלפים של פרויקטים שנמצאים בפיתוח.

\*פלטפורמה שתפתח טכנולוגית בקצב מהיר. זה מובטח על ידי מודל פיתוח מודרני מבוסס קוד פתוח שלינוקס מיישמת: "נבנה על גב המפתח הקודם" וגם "עמית מסתכל בקוד שלך" בניגוד לגישה האנכרוניסטית הסגורה "תמיד תתחיל מכלום" ו"אף אחד לא יראה את הקוד שלי". אפילו אם האופנה האחרונה הלכה לעולמה לינוקס תמשיך להתפתח בדיוק כמו לפני שהאופנה הזו התחילה. לפיתוח בקוד פתוח יש את המוזריות שלו: הפיתוח נראה איטי למדי (במאונך) אבל הוא חל על גבי חזית רחבה מאוד, באגים חמורים באבטחה מתוקנים כמעט עם זמן הגילוי שלהם, ישנן בדרך כלל כמה אלטרנטיבות לתוכנית או לפונקצייה מסויימת.

אם אתה מעוניין ללמוד ממקור ראשון על הרשיון הציבורי הכללי, בדוק את מסמכי GNU האלה:

<http://www.gnu.org/copyleft/gpl.html>

<http://www.gnu.org/gnu/linux-and-gnu.html>

<http://www.gnu.org/philosophy/categories.html#thegnusystem>

סביר להניח שהרשיון תחתיו מופצת לינוקס הינו אחד החלקים היותר חשובים במסמך זה. הרשיון נועד לקדם את חופש המידע. פרויקטים נוספים של הקוד הפתוח כוללים מדע וחוק (בקושי בדיחה) השיטה של לינוקס אינה משהו חדש--היא פשוט ההחלה של הגישה המדעית על תוכנה: אתה מקבל מידע באופן חופשי, מוסיף את הרעיונות שלך, מתפרנס, ולבסוף, משאיר את המידע חופשי. אבל נראה שכמה תאגידיים גדולים ועורכי הדין שלהם מתאמצים מאוד כדי לשנות את זה, הם מנסים לדחוף את הזמן אחורה אל ימי הביניים, אל הזמנים בהם מידע היה בבעלות מישהו. לכן, אתה יכול לראות כמה אנשים שקשורים ללינוקס מעורבים בכל מיני מאבקים.

למילה "חופשי" יש שתי משמעויות די שונות בשפה האנגלית (והעיברית-א.א.), מה שמוביל לתפיסה לא נכונה את הטבע החופשי של לינוקס. שתי משמעויות אלו נובעות משם התואר הלטיני "liber" ומתואר הפועל "gratis" והן בדרך כלל מתקשרות אל המינוחים "free speech" ו-"free (of charge) beer". רוב התוכנה בלינוקס היא free בשני המובנים, אבל רק הראשון הכרחי ללינוקס.

## 0.5 מה הם ההבדלים בין לינוקס ליוניקס?

מבחינת שימוש בשורת הפקודה, כמעט כלום, למרות שזה משתנה (לטובה או לרעה). ללינוקס יש הרבה יותר כח משיכה שיווקי. מבחינת ממשק גרפי, ההבדלים אינם מז'וריים--לינוקס, כמו רוב היוניקסים משתמשת במערכת x-windows.

ההבדלים העיקריים:

לינוקס חופשית, בעוד יוניקסים רבים עולים הרבה. כנ"ל לגבי אפליקציות--אפליקציות טובות רבות ניתנות להשגה באופן חופשי ללינוקס. אפילו אותה אפליקציה מסחרית (אם אתה מעוניין לקנות אחת) עולה הרבה יותר ליוניקס מסחרית מאשר ללינוקס.

לינוקס רצה על פלטפורמות חומרה רבות, בעוד יוניקס טיפוסית קשורה אל חומרה מסויימת בדרך כלל-וחומרה זו נוטה להיות יקרה הרבה יותר מאשר תואם פיסי טיפוס. עם לינוקס, אתה בשליטה על המחשב שלך, בעוד ברוב היוניקסים אתה בדרך כלל מוגדר כ"i-user" (כמה אדמיניסטרטורים מבטאים זאת "loser")--לינוקס מרגישה קצת כמו דוס ווינדוס בשנות ה-80/90 המאוחרות, אבל היא הרבה יותר יציבה ועשירה. בעוד מערכת יוניקס טיפוסית מרגישה כמו מיינפריים משנות ה-60/70. ייתכן שיוניקסים רבים בוגרים יותר באזורים מסויימים (אבטחה למשל, כמה אפליקציות שקשורות בהנדסה, תמיכה טובה יותר בחומרה "כבדה"). לינוקס היא יותר בשביל ג'ו שרוצה להריץ את שרת משל עצמו או תחנת עבודה הנדסית.

## 0.6 מה הם ההבדלים בין לינוקס למייקרוסופט ווינדוס?

מבחינת הקלקה על העכבר, כמעט אין, מרגע שלינוקס מותקנת כראוי.

ההבדלים העיקריים:  
לינוקס חופשית, בעוד ווינדוס עולה כסף. אותו הדבר לגבי אפליקציות, פורמטי הקבצים של לינוקס חופשיים, כך שאתה יכול לגשת אליהם במיגוון דרכים. בווינדוס, העיקרון הנפוץ הוא לגרום לך לנעול את המידע שלך בפורמטים סודיים שניתן לגשת אליהם רק בעזרת כלים שהושכרו לך על ידי היצרן במחיר שלו. עד כמה מושחתים (או לא מתאימים לתפקיד?) צריכים להיות הפוליטיקאים שנועלים את המידע הציבורי שלנו בפורמטים כאלו! מה שנקבל עם מייקרוסופט זה חוזה שכירות לשלוש שנים על רשומות בריאות שאנחנו צריכים לשמור "למשך מאה שנים"

[http://news.bbc.co.uk/hi/english/health/newsid\\_1694000/1694372.stm](http://news.bbc.co.uk/hi/english/health/newsid_1694000/1694372.stm)

עם לינוקס, הסבירות שתפר איזה שהוא הסכם רישוי, נמוכה. כל התוכנה, שלך היא. עם מייקרוסופט ווינדוס סביר להניח שכבר הפרת כל מיני הסכמי רישוי וניתן להכריז עליך כעל פיראט מחשבים אם רק איזה עורך דין חכם ירצה בכך. מייקרוסופט ווינדוס מנסה להיות מערכת ההפעלה בעלת המכנה המשותף הנמוך ביותר (לטובה או לרעה) בעוד לינוקס בנויה למשתמשי מחשבים מתוחכמים יותר (לטובה או לרעה) מייקרוסופט ווינדוס מבוססת על דוס. לינוקס מבוססת על יוניקס. הממשק הגרפי של ווינדוס מבוסס על הנחיות של מייקרוסופט עצמה בעוד הממשק הגרפי של לינוקס מבוסס על סטנדרט התעשייה X-WINDOWS. לינוקס מרפה את ידי ווינדוס בכל מה שקשור לאפשרויות רשת, כפלטפורמת פיתוח, יכולות עיבוד נתונים, וכתחנת עבודה מדעית. לשולחן העבודה של ווינדוס יש מראה מבריק יותר, אפליקציות עסקיות חלקות יותר, ויותר משחקים לילדים (אלו לא משחקים טובים יותר, המשחקים של לינוקס נוטים להיות מתוחכמים יותר). לינוקס הרבה יותר עשירה באפשרויות מאשר תוכל לדמיין.

נשמע פעם באינטרנט:  
"שני מוצרים גדולים הגיעו מהאוניברסיטה של קליפורניה: UNIX ו-LSD.  
ואני לא חושב שמדובר בצירוף מקרים."

## 0.7 אני לא מאמין בתוכנה חינם וכו'

והאם אתה מאמין באינטרנט? האינטרנט ולינוקס חולקים רעיונות משותפים רבים ויש להם שורשים משותפים. האם אתה זוכר את חוסר האמונה ששר לגבי האינטרנט לפני כמה שנים? הוויכוח הבלתי נגמר והבלתי מנוצח לגבי העובדה שאינטרנט חופשי לא יכול להתקיים. המציאות פשוטה. שיתוף פעולה ורצון טוב יכולים להועיל לרבים באותו זמן: המאמץ שלך אינו ההפסד שלי. האינטרנט עובד בסדר גמור ומתפתח בקצב מהיר. כך גם לינוקס.

לפניך דעה של מנהל ב-IBM:  
"הסיבה שבגללה אנחנו כל כך מתרגשים לגבי לינוקס היא שאנחנו מאמינים שלינוקס יכולה לעשות לאפליקציות מה שהאינטרנט עשה לרשתות"  
IBM הוציאה זה עתה (מאי 2002) ביליון דולר כדי לגרום ללינוקס לרוץ על כל פלטפורמות החומרה שלה (מיינפריים, תחנות עבודה, פייסי, ניידים).

## 0.8 אין דבר כזה ארוחות חינם?

ברט בזנט -Bret Bazant<bbazant@shaw.wave.ca> כתב ב-linix today:

המבנה העיסקי שהופך אמירה זו לנכונה תלוי במלאי המשאבים. משאבי תוכנה הינם דלים כי כולנו שומרים עליהם בבעלות ובסודיות. אבל לא לינוקס! כשאני נותן לך את התוכנה שלי, ייתכן מאוד שאני יוצר לעצמי הזדמנות, בכל מקרה אני מצליח לשמור עליה אפילו לאחר שנתתי לך אותה. זוהי ארוחת חנינם שנחשפת בהיסטוריה רק על ידי ככרות לחם ודגים. (מדובר כאן על נס מהנצרות שמזכיר קצת את נס חנוכה מהיהדות-בו קבוצת אנשים שברשותה אוכל מחלקת אותו לאחרים והאוכל לא רק שאינו נגמר אלא מספיק לעוד ועוד אנשים ואינו מתכלה, -א.א-תודה לגלעד בן יוסף).

## **0.9 אני זקוק לאבטחה ברמה גבוהה. עם תוכנה מסחרית אני יכול לתבוע אותם אם משהו מתפקשש**

אל תסתמך על תביעה. מדי יום ביומו, דברים מתפקשים על מכונות NT. קרא את הסכם הרשיון שיש לך ממייקרוסופט כדי לגלות שאין כל ערבות לך שמהו יעבוד ואין נזקים שמוכרים בבתי משפט. בכלל, נסיון לתבוע יהיה ביזבוז של כסף. גם לינוקס לא מספקת כל ערבות, למרות שהיא הרבה יותר מאובטחת מכל גרסה של ווינדוס. אם אתה באמת רגיש לגבי אבטחה, אתה יכול להשתמש בכלי האבטחה החזקים שנבנים על ידי חברות שנסמכות על הזמינות של קוד המקור כדי לעצב את כלי האבטחה שלהן (קריפטקום Kriptkom מגרמניה- מספקת פיירוול חזק).

האבטחה העמומה שמיושמת במייקרוסופט ווינדוס הוכחה פעמים רבות כגישה נאיבית.

לפניך ציטוט מפי דויד קרטרופ, מהנדס מחקר, בוכם, גרמניה:  
"העובדה הפשוטה שמכתיבה את השימוש בלינוקס ובמוצרים פתוחים אחרים ולא במייקרוסופט ווינדוס NT היא הימנעות ובריחה מסיכון. הסיכונים שאתה נוטל על ידי שימוש ב-NT-הינם לחלוטין מחוץ לכלל שליטתך, ואין אף אחד שתוכל לתבוע במידה ומשהו לא מסתדר כשורה. העובדה שאנשים עדיין מאמינים במיתוס שווינדוס מציעה יותר אמינות מאשר לינוקס עדיין נשאר בגדר חידה בשבילי".

## **0.10 אני צריך סטנדרטים. תאגידים גדולים (מייקרוסופט) מספקים סטנדרטים**

אולי זה מה שאנשים מצפים מתאגידים גדולים, אבל המציאות שונה. בעבר חברות גדולות אהבו להמציא אגוזים שיכלו להתפצח רק בחנויות השרות שלהן. האם אגוזים אלו הפכו להיות הסטנדרט? בקושי. הם לא הפכו כי לא הייתה מעורבת בכך טובת הציבור, והם לא יכלו להפוך לכאלה כי הם היו פטנט. למזלנו, היום יש לנו סטנדרטים פתוחים וחופשיים לאגוזים. "סטנדרט שנמצא בבעלות" הוא כזה אוקסימורון מגוחך שממש קשה להאמין שאנשים משכילים יכולים להאמין בו. (נכון לעכשיו, מנגנוני השיווק מכתיבים לנו את המושג "הדה פקטו סטנדרט" או "סטנדרט התעשייה" כדי לכסות על הכיעור של חוסר הסטנדרטים)

דוגמא מעולם המחשבים. הפורמט ה"סטנדרטי" של מייקרוסופט וורד שונה פעמים רבות במשך השנים האחרונות. וקרוב לוודאי שהוא ממשיך להשתנות מסיבה משתלמת כלכלית: ברגע שחברות אחרות. מבצעות מה שניקרא "הינדוס הפוך" על פורמט הוורד הנוכחי, מייקרוסופט משנה אותו. ישנם אפילו תתי פורמטים (שמירה מהירה של מייקרוסופט אומר למישהו משהו?). מה גם שכל התחום הזה הוא סגור לגמרי. מייקרוסופט לא מפרסמת כל הנחיות. איך משתמש יכול להרוויח מזה בטווח הרחוק. מה היא הערבות של מייקרוסופט שהפורמט של וורד 6 יהיה תקף גם ב-2020?

קסביר בסורה:

"הסטנדרטים של מיקרוסופט הינם עריציים-החמקמקות של חוסר התאימות בין פורמט הקבצים של אופיס 97 עם גרסאות ישנות יותר או תת הגרסאות של סטנדרטים פתוחים כמו X.M.L עם תוספות בלעדיות שדורשות אינטרנט אקספלורר 5, האקטיב סרבר של מייקרוסופט, וכיצא באלו, הינם תזכורות פיכחות למה שהחברה הזו עושה לשוק..."

המונופול של מייקרוסופט לא מבטיח לך שגירסת האופיס הנוכחית שלך תעבוד עם כל גירסא קודמת או עתידית של אופיס. וזה למרות שמספר מתנצלי מייקרוסופט טוענים שההטבה מהמונופול של מייקרוסופט היא קביעת הסטנדרט בייצור אפליקציות.

כדי להוסיף על הביבלול, חברות בדרך כלל לא "מסנדרות" את פורמטי הקבצים שלהם אלא רק לאפליקציות שאמורות לתמוך בהן. זה כמו לקבוע סטנדרט לייצרן אגוזים במקום על האגוזים עצמם. איך זה אמור לעבוד אם ייצור הקבצים ממשיך להשתנות במטרה מפורשת כדי להעצים את המכירות.

טד קלרק (קומפק):

"אנחנו זקוקים לפורמטי קבצים פתוחים בעלי סטנדרט, כדי שמשמשים יוכלו להחליף מסמכים בין פלטפורמות. תוכנת עיבוד התמלילים ששימשה לייצרת הקבצים הללו בכלל לא אמורה להיות אישו"

ישנם כמה פורמטי קבצים נוספים מקושרי טקסט או מסמך שהינם יותר סטנדרט מפורמט הקבצים של מייקרוסופט וורד:  
...ascii, xml, html, sgml, latex, tex, postscript, pdf, dvi  
ולכולם יש תמיכה מצויינת תחת לינוקס. גם פורמט הקבצים של word יכול להיכתב ולהיקרא היטב תחת לינוקס.

לינוקס מטיבעה מבוססת על סטנדרטים אמיתיים, מופצים וחופשיים בגלל שהקוד הפתוח גורם לספסיפיקציות להיות זמינות לכולם (מתחרים ולא מתחרים). אני חושב שהצורך הדחוף בסטנדרט פתוח הוא אחד מהכוחות המניעים שמאחורי לינוקס. חלק מהאנשים מרגישים שהם אינם יכולים להרשות לאלגוריתמים ולנתונים שלהם להיות יישות מסחרית, ועוזבים לבד את מי שבאופן קבוע מפגין חוסר אמינות. הסתכל בטיטה הזו של חוק ארגנטינאי כדי לטעום מן העתיד. זה נשמע כאילו הארגנטינאים יהיו הראשונים שיחליטו שהרשומות הציבוריות שלהם לא יכולות להיות מוחזקות כבנות ערובה על ידי יישות מסחרית.

"אירגונים ציבוריים לאומיים שמוזכרים במאמר 1 של חוק זה, לא יורשו להשתמש בתוכניות שמאחסנות מידע בפורמט לא ציבורי"

דעה עמוקה בקהילת המשתמשים של לינוקס טוענת שאכן יש בעייה רצינית עם סטנדרטי מיחשוב שמנוצחים על ידי יצרני תוכנה גדולים. זה כולל את הנתונים ה"סטטיים" שלנו, כמו גם את המידע המוטמע בקוד על גבי המחשב שלנו. האם אנחנו יכולים להרשות לעצמנו שיהיה מישהו אחר שיחליט בשבילנו מתי, איך, ובאיזה מחיר אנחנו יכולים לגשת לנתונים אלו? הבעייה זוכה להתעלמות ואפילו זוכה לעידוד מהאנשים שמקבלים שכר בשביל לדאוג לנושא.

אם אתה כותב תוכנת מחשב שאמורה לאחסן כל סוג של מידע לגבי החברה שלך, כדאי שתיתק את העצות הבאות בחשבון:

1. בחר בסטנדרט מבוסס קוד פתוח.

2. שמור על פשטות. בצורה זו אתה מבטיח הגנה אמיתית להשקעה שלך בתוכנה. אותו הדבר נכון לגבי בחירת הפורמטים שאתה בוחר לאיחסון המידע היקר שלך.

### 0.11 הפופולריות של מייקרוסופט מבטיחה שהיא כאן כדי להישאר

קרוב לוודאי שזה נכון. נניטנדו גם היא כאן כדי להישאר. בכל מקרה אני אוהב מיחשוב אז אני בוחר במערכת הפעלה עוצמתית, לא את המערכת בעלת המכנה המשותף הנמוך ביותר שמעוצבת ל"כולם".

לינוקס היא די כאן להישאר בגלל טבע הקוד הפתוח שלה (לינוקס פשוט לא יכולה להיות מחוץ לעסקים) היא הבחירה הסטנדרטית לאין ספור פרוייקטים שלא הולכים להיעלם, וחלקם אף די קריטיים. בדוק את תחנת החלל הבינלאומית, שבה לינוקס היא מערכת ההפעלה -

<http://www.linuxjournal.com/article.php?sid=3024>

ובנוסף, לעולם אל תפחית בערך כוחה של קהילת הלינוקס. לינוקס היא "כאן כדי להישאר" לפחות בשביל אוונגרד המיחשוב. לינוקסאים רבים לא אפילו לא מעוניינים שלינוקס תהיה פופולרית מאוד מכיוון שהם מפחדים שזה יכול פגוע במעמד העלית של לינוקס.

### 0.12 אבל לינוקס מסתעפת לכמה מערכות שונות...

טיעון נפוץ שמופץ על ידי אלו שמתמחים בשיטות שיווק הידועות כ"פחד, חוסר וודאות וספק".

מסתעפת בהקשר זה פירושה חלוקה לענפים של תוכנית מחשב. כמו גם יצירת תת גירסאות של התוכנית. ולבסוף לעשות פרגמנטציה ללינוקס.

ישנה עדות דלה (אם בכלל) לגבי סיעוף מזיק של כל תוכנה שכלולה בהפצת לינוקס אופיינית. היכן שכן התרחש סיעוף, הוא תמיד הפך למועיל. די אפשרי שזה התרחש, על אף שאין מחסומים מלאכותיים לסיעוף תוכנה תחת לינוקס, בנוסף גם אין כל מחסומים מלאכותיים שמונעים שילוב של החלקים הטובים ביותר חזרה.

הרקע התיאורתי על איך סיעוף תוכנה יכול להועיל להתפתחות שלה ניתן למעשה לפני זמן רב על ידי הפילוסוף הגרמני גורג וויליאם פרדריך הגל (1770-1831), שעם קונספט ההתפתחות הדיאלקטית שלו בתופעת הרוח אמר "כשקיים פילוג בקבוצה, נראה כי הוא מבטא את דלותה, אך למעשה דווקא את עושרה הוא מבטא"

### 0.13 לינוקס זה קאלט

לקהילת הלינוקס הודבקה פעמים רבות תזית של "קנאים דתיים", במיוחד על ידי עיתונאים שמגזיני המחשב המאוד מבוססים בהם הם עובדים קיבלו משוב מסיבי לגבי כתבות לא צודקות על לינוקס. אכן כן, קהילת הלינוקס היא רבת מספרים, מלומדת, ומוכנה לבטא את דעותיה. הרבה מגזיני/עיתוני מחשבים יודעים שלינוקס שווה פחות כסף בשבילם (משתמשים שמשלמים פחות על כל מה שמסביב למחשבים ולציוד נלווה ודורשים יותר) האם זה מסביר את המושג קנאה דתית?

תודו בזה, אנשי המכירות שלכם מעמידים פני עיתונאים. בקושי נשארה איזו שהיא כנות בעיתונות המחשבים. כמה מילים על לינוקס פירסם הפי. סי מגזין (או מה שלא יהיה) בשנים 02-01-1999? האם לינוקס לא הייתה לפחות עניין טכנולוגי בזמן הזה? בוודאי שהיא הייתה, אבל בכל זאת זממתם להשאיר את קהל הקוראים שלכם באפלה, מוכרים את היושר העיתונאי שלכם בעבור כסף. ועכשיו, לאחר שלינוקס צפה אל פני הזרם המרכזי בתקשורת (לא קשורת המחשבים), אתם ממשיכים לכתוב כתבות שאומרות "כן, אבל לינוקס לא תוכל /לא יכולה / אולי תוכל ("...מנסים להשתמש בטקטיקת ה"פחד, חוסר הוודאות והספק" כדי להרוג את התופעה הזו) ובנוסף כותבים " מייקרוסופט כבר ... ממשיכים לכתוב על הפלטפורמה החמקמקה, הלא יציבה היקרה וחסרת הארכיטקטורה, שההישג הגדול ביותר שלה היה רדיפת בצע שלא נשמעה כמותה על ידי מניעת תאימות כוללת, והרג של כל סטנדרט מוצע (על ידי אימוצו וכינוסו לבעלותכם על יד תוספות). את מי אתם משרתים? בטוח שלא את הקוראים שלכם.

אמרתי את זה בצורה החזקה ביותר שיכולתי. האם אני קנאי? או שאני רק מנסה להביע את חוסר ההסכמה שלי לגבי מעשיכם שלא משרתים אדם מלבד את עצמכם?

אתם חושבים שסיפוק עצמי זה בסדר בעסקים? עד כמה מעורר רחמים העסק שלכם צריך להיות! אני תמיד חשבתי שעסקים זה חוזה חברתי שבו אנחנו מחליפים סחורות בעלות ערך, ככה שהתועלת תהיה הדדית. לאורך ההיסטוריה, חברות נהגו לתלות, להגלות או לערוף את ראשי אלו שבאמת התמקדו בעסקים האנוכיים שלהם בלבד. ובכן הזמנים השתנו. לטובה או לרעה, קצת לרעה :)))

#### 0.14 העלות הכוללת הסופית של הבעלות על לינוקס גבוהה

אף אחד לא יודע באמת איך לחשב "עלות כללית סופית" של חתיכת תוכנה. לכן עורך דין טוב +רואה חשבון טוב יכולים להוכיח כל דבר שהם מקבלים כסף תמורתו, והם בדרך כלל עושים את זה. תנו לי לבחון הערכה קטנה לגבי העלות הכללית הסופית שנובעת מבעלות על מייקרוסופט ווינדוס בוא ונחבר את העושר המצטבר של כל יוצרי ווינדוס, נוסף את כל המשכורות של כל מתכנתי ווינדוס, היועצים, אנשי תמיכה והוראה, ניהול וכו' ועכשיו נוסף את ההפסדים שסביר להניח שלקוחות נחלו בזמן שתאגידי תוכנה הציגו להם פיצ'רים רק כדי להגיע למונופול הנוכחי שלהם. נחלק את זה במספר השנים - כל מסגרת זמן שהיא - ואת מספר משתמשי ווינדוס. איך שלא תחשב את זה, זה תמיד יצא אלפים רבים של דולרים אמריקאים לשנת עבודה ממוצעת. לא שילמת כל כך הרבה כסף? ובכן, בטח שכן, זה רק נחבא ממך. כן, מדינות מפותחות מבזבזות ביליונים בכל שנה על תוכנה.

כמה עלתה לינוקס? בקושי כלום. מספר המשתמשים נמוך בהרבה, אבל תצטרך להתאמץ כדי לקבל תוצאה של \$10 בשנה.

בכל זאת לפי דעתי, העלות הכללית אינה משנה במיוחד. מה שמשנה זה מה שאני מקבל בתמורה לכסף שלי. צריך לחשב את הערך הכולל של הבעלות על המוצר, ואז לחסר אותו מהעלות הכוללת הסופית של הבעלות עליו בכדי לחשב את ההטבה שמגיעה כתוצאה מהבעלות על המוצר.

אני מניח שרואי חשבון מדברים רק על עלות כוללת סופית של בעלות על התוכנה כערך "חיוני לעשיית עסקים" ולכן מדלגים על על המשמעות של הערך הסופי וההטבה שמגיע כתוצאה מהשימוש בה. אין כל ערך בתוכנה המסחרית הנורמלית, היא חיונית רק לעשיית עסקים בימים אלו. ובכן, לינוקס עונה על הצרכי המיחשוב שלי בעלות אפס, ובנוסף ההנאה האישית והערך הלימודי הם רבים.

## 0.16 לינוקס חולמת בהקיץ; עסקים זה מה ששולט בעולם היום

חשוב על לינוקס כעל קונסורטיום. עסקים/יחידים נאספים ביחד כדי להשיג צרכי מיחשוב דומים או כדי לפתור בעייה משותפת. ייתכן ואין להם כסף, כח אדם או מנהיג טכני אבל הם יכולים לשתף פעולה כדי לגרום ללינוקס לענות על צרכיהם. הפתרון הוא לגמרי שלהם, והוא לא צריך לעלות הרבה--הם יכולים להשתמש מחדש בחלקי התוכנה הקיימים. הם יכולים לשתף פעולה כדי לגבור על הייתרון שיש ל"מנהיג תעשייה" גדול ושיכול לשמש כנגד האינטרסים שלהם בסופו של דבר. לינוקס זה התוצר הסופי של פעילויות שונות והרבה קונסוטוריום כאלו ש"גירדו את צרכיהם". אז לינוקס זה עסקים, אבל זה לא בהכרח עוסק במכירת תוכנה--זה יותר נוגע בגישה לתוכנה במחיר הגיוני שמתאימה לצרכיך, פותרת את הבעייה שלך, מוכרת את החומרה שלך או את השירות שלך, ושלך באופן טוטאלי ( הרשיון לעולם לא פג, ולעולם לא תנושל או תורחק מקוד המקור).

## 0.16 לינוקס זה חרטא וכו'

אז אל תשתמש בלינוקס. אתה לא עושה לאף אחד טובה על ידי שימוש בלינוקס. GNU/Linux זה תוכנה חופשית ועוצמתית, אבל רק בשביל אלו שצריכים אותה. ישנן מערכות הפעלה חלופיות שאתה יכול לבחור ושיכולות לענות טוב יותר על צרכיך. למרות שרוב הלינוקסאים נהנים מההתרחבות ומקבלים בברכה משתמשים חדשים, חלק לא שמחים כל כך מהעניין בגלל ש"הקהל והמסחור יכולים לקלקל את גן העדן להאקרים שיצרנו" לכן אינך עושה טובה לאיש על ידי שימוש בלינוקס. בהקשר זה, זה יכול להועיל אם נסכם את העוצמות והחולשות של לינוקס:

לינוקס נמצאת בבעלות האוהדים שלה (החלק שלך בבעלות מגיע חינם ביחד עם ההצטרפות החופשית שלך למועדון האוהדים) היא בהחלט עוצמתית ועשירה בפיצ'רים, בעלת יכולות קינפוג גבוהות, גמישה עד כמה שתמצא שתהיה, באה ביחד עם מורכבות, הוצאות פחותות על חומרה, מגיעה עם כל יכולות הרשת המוכרות לאדם, דורשת מנהל רשת שידע קרוא וכתוב במחשבים, חלק מאפליקציות שולחן העבודה עדיין בפיגור אחרי המקבילות המסחריות בפלטפורמות אחרות, מספר רחב של אפליקציות משתמש קצה באות כסטנדרט וחינם עם מערכת ההפעלה, הממשק הגרפי נחמד מאוד אבל עדיין לא משויף כמו של אפל או מייקרוסופט. לינוקס עובדת בסטנדרט גבוה (יוניקס, פוסיקס), פורמטי קבצים פתוחים נמצאים בשימוש כל הזמן, אלפי תוכניות זמינות להורדה חינם (למרות שקלות השימוש והאיכות שלהן משתנות מאוד), ומעל לכל, לינוקס מהנה!

הבא: לפני התקנת לינוקס.



## חלק 1. לפני התקנת לינוקס

### תוכן החלק הזה:

- 1.1 באיזו הפצת לינוקס כדאי לי להשתמש? [עמ' 13]
- 1.2 מה הן דרישות החומרה של לינוקס? [עמ' 16]
- 1.3 האם החומרה שלי תעבוד תחת לינוקס? [עמ' 18]
- 1.4 איך אני מוריד את לינוקס מהאינטרנט? [עמ' 20]
- 1.5 איך אני משיג דיסק של לינוקס? [עמ' 20]
- 1.6 יש לי דיסק של רד האט אבל אין לי את הפלופי של ההתקנה [עמ' 20]
- 1.7 מה אני צריך לקרוא לפני ההתקנה? [עמ' 21]
- 1.8 האם אני יכול להתקין מייקרוסופט ווינדוס ולינוקס על אותו המחשב? [עמ' 22]
- 1.9 איך אני מחלק למחיצות את הדיסק הקשיח שלי? [עמ' 22]
- 1.10 המחיצה של מייקרוסופט ווינדוס תופסת את כל הדיסק הקשיח שלי. האם אני יכול לכווץ/לחלק את הדיסק בלי להתקין מחדש? [עמ' 28]
- 1.11 איך אני מתחיל את ההתקנה? [עמ' 28]
- 1.12 האם ההתקנה של לינוקס קשה? [עמ' 29]
- 1.13 אילו חבילות עליי להתקין? [עמ' 29]
- 1.14 איזה ממשק גרפי להתקין, קיי.די. אי או גנום? [עמ' 30]
- 1.15 סיימתי את ההתקנה, איך אני נכנס למערכת בפעם הראשונה? [עמ' 31]
- 1.16 איך אני גורם ללינוקס לקרוס? [עמ' 31]
- 1.17 האם אני יכול להשתמש בממשק הגרפי כל הזמן? [עמ' 32]
- 1.18 איך אני משדרג הפצת לינוקס? [עמ' 33]

## 1.1? באיזו הפצת לינוקס עליי להשתמש

הפצת לינוקס היא אוסף קוהרנטי של תוכנות חופשיות כשהליבה של לינוקס (מערכת ההפעלה) משמשת כמרכזן.

ההבדלים בין הפצות לינוקס השונות, מינוריים: תוכנית ההתקנה, מבחר הכלים/תוכנות שמחוברים אל המערכת, אירגון כמה דברים על הדיסק הקשיח (רובו של המיקום הוא די זהה, מיקום סטנדרטי בדיסק הקשיח בכל הפצות). בכל הפצה שלא תבחר להתקין בסוף תישאר עם אותה לינוקס .  
אנחנו משתמשים בעיקר ב "MANDRAKE"- (MDK) וב "RED HAT" (RH) מהסיבות הבאות:

1. שתיהן מאוד פופולריות (עובדה שגם עוזרת למשתמש חדש וגם מהווה מדד לאיכות שלהן).

2. שתיהן הפצות שמתאימות למטרות כלליות.

3. שתיהן מגיעות עם תוכניות התקנה קלות יחסית.

4. גם מנדרייק וגם רד האט הינן "תוכנות פתוחות" זה אומר שכל התוכנה שנכתבה על ידי הייצרנים ושכלולה בתקליטורי ההפצה נמצאת באופן חוקי תחת הרשיון הכללי הציבורי (ה-GPL), כך שאפשר להעתיק את התקליטורים שלהן, אפשר לתת אותם, להשתמש מחדש וכו'.

5. את מנדרייק ואת רד האט ניתן להשיג בזול מאוד או בחינם אם אין לך עניין בתמיכה מסחרית. וזה כתוצאה מ-4.

6. מנדרייק מבוססת על רד האט, כך ששתיהן דומות. חבילות תוכנה לרד האט בדרך כלל עובדות על מנדרייק ללא בעיות ולהיפך. בכל אופן, מנדרייק קצת יותר אוטומטית, בעלת שולחן עבודה נחמד יותר ודורשת פנטיום כברירת מחדל, רד האט תעבוד גם על i386 .

בקצרה, כמתחיל, אתה יכול להמר בביטחה על מנדרייק או רד האט אלא אם כן אתה מעדיף משהו אחר או שיש לך דרישות מיוחדות, או שהסביבה שלך מציעה לך להשתמש במשהו אחר (אם יש לך גורו מנוסה באזור, או כמה חברים שמתמשים בלינוקס, יכול להיות שתרצה להשתמש באותה הפצה כמוהם, זה הופך את השגת העזרה למשהו קל בהרבה).

ההפצות האחרונות שאנחנו ממליצים עליהן (יוני 2000) הן רד האט 7.3 או מנדרייק 8.2. אלו הן הפצות מצויינות, הקפד לציין את ההפצה הכי עדכנית כשאתה מזמין את התוכנה שלך מסוחר-- סוחרים רבים אוהבים לנקות את המלאי שלהם על ידי שליחת הפצות ישנות יותר (וזה ישים לא רק על לינוקס). באופן כללי, הפיתוח תחת לינוקס הוא מהיר, ולא תרצה לבזבז את זמנך על הפצות ישנות, . לכותבים של מדריך זה אין כל קשר לרד האט, מנדרייק (ואו לכל הפצה אחרת) בכל אופן שהוא.

ההמלצה שלנו על מנדרייק ורד האט למתחילים לא אומרת שהפצות אחרות לא מציעות ייתרונות או תכונות מיוחדות שעולות על רד האט או מנדרייק בתחומים מסויימים. אנחנו מאמינים שאנחנו מרוויחים מחשיפה להפצות אחרות כי זה עוזר לנו להבין את לינוקס באופן טוב יותר.

ניסינו את דביאן (DEBIAN) ואהבנו אותה מאוד. היא הייתה קלה כמו רד האט אבל נראה שהיא פחות נפוצה (לכן, בהיותנו מתחילים בחרנו ברד האט). מה שיפה בדביאן זה שהיא 100% לא מסחרית (היא מורכבת על ידי האקרים מתנדבים, בהתאם לאופי האמיתי של לינוקס) היא גם הדבקה ביותר בסטנדרטים של לינוקס וכנראה גם קובעת את הסטנדרטים של המערכת.

עוד דבר יפה בדביאן זה שתקליטורי ההפצה שלה מפוצצים בכלים ואפליקציות--הרבה יותר מכל הפצה אחרת. כל האפליקציות/כלים האלה ארוזים באופן נח (כדי להקל על ההתקנה) ונבחרים לתאימות. אז כן, לא תהיה לנו כל בעיה להמליץ על דביאן, הפצת לינוקס גדולה יותר למטרה כללית. דביאן קוראת לעצמה "מערכת ההפעלה האוניבסלית" בגלל סיבה טובה.

בכל פעם דביאן מוציאה שלוש גרסאות :

1. הגירסא ה"יציבה" (לפעמים נקראת -POTATO) ולא היינו ממליצים עליה, אלא אם כן אתה באמת פאראנוייד לגבי יציבות ולא אכפת לך מחבילות די מיושנות.

2. גירסת המבחן של דביאן (לעיתים נקראת -WOODY) והיא יציבה בטח כמו הגירסה היציבה האחרונה של רד האט, ויותר יציבה מהמנדרייק הנוכחית שלך. היא הרבה יותר מעודכנת מדביאן היציבה. דביאן וודי היא הגירסא שאנחנו אוהבים.

3. אם לא אכפת לך להיתקל קצת בצרות, אתה יכול לנסות גם את הוורסיה השלישית שנקראת "לא יציבה" (unstable) אבל קרוב לוודאי גם הכי מעודכנת.

קורל (COREL) עבדו פעם על גירסאות לינוקס משלהם. כנראה בכדי ליצור פלטפורמה נוחה עליה יוכלו להריץ את חבילת יישומי האופיס שלהם: וורדפרפקט, וורדפרוסור, גליון הנתונים קוואנטום פרו, את תוכנת המצגות של קורל, מסד הנתונים פרדוקס, חבילת האומן של קורל דרו... קורל לינוקס התבססה על דביאן, היא דווקא נראתה די מבטיחה, אבל לא ברור לי מה קורל עשתה איתה (קיבלה כסף ממיקרוסופט כדי לרדת מהנושא?).

בקיצור, קורל לינוקס מתה עכשיו, ולעולם לא אמליץ עליה לאף אחד מכיוון שהיא דרך ללא מוצא. הסיבה היחידה להזכיר אותה כאן היא בגלל שקורל לינוקס זכתה פעם לפרסום רב, אז יכול להיות שעוד תשמע עליה פה ושם.

(http://www.corel.com/learning/wbt/linuxos1/-/א.א)

סלאקוור (SLACKWAR)

חביבה בדרך כלל על האקרים מהשורה הראשונה שאוהבים להיות קרוב אל מערכת ההפעלה ואל החומרה שלהם--לעולם לא השתמשנו בה אבל זה מה ששמענו.

יהיה לנו קשה להמליץ על סלאקוור למשתמשי לינוקס חדשים  
הכתב שלנו ביל סטילס מדווח: "הסיבה האמיתית שבגללה משתמש חדש צריך להמנע  
מסלאקוור היא בגלל שהיא דורשת שימוש רב יותר בשורת הפקודה ובגלל שהיא חסרה את  
הכלים החלקלקים והנוטפים שיש להפצות אחרות." בכל מקרה קיבלנו משוב ממשתמשי  
לינוקס חדשים שמשתמשים בסלאקוור והיא עובדת טוב מאוד אצלם. לכן עם קצת מאמץ,  
אדמיניסטרטור שמבין במחשבים יכול להבין באמת מה קורה במערכת ההפעלה שלו ( זה  
משהו שאני לא תמיד יכול לומר על מנדרייק, או על מייקרוסופט ווינדוס ).  
אולי סלאקוור היא בשביל לינוקס מה שדוס עבור ווינדוס :

S.u.S.E-זזה

היא הפצה מאוד פופולרית באירופה. היא בהחלט נראית מגרמניה--הפצה מוצקה למטרות  
כלליות עם תוכנית התקנה קלה ומוניטין מצויין. משתמשים רבים נשבים בסוזה. לא יכולנו  
למצוא תקליטורי סוזה זולים למרות שנראה שאפשר להוריד אותם מהרשת.

קלדרה (caldera) הינה עוד הפצה ידוע ומכובדת מאוד. אומרים שהיא מכוונת אל המשתמש  
הארגוני, יש לה את תוכנית ההתקנה המפוארת ביותר, אוסף של כלים מתקדמים (ויקרים)  
כלים לקיפוג מרחוק, ועוד כלים טובים שמתאימים לארגון.

קלדרה רכשה את היוניקס המקורית-SCO-, באוגוסט 2000, (מה שמשווה לה בעיניי מראה  
עוד יותר "אירגוני" ומקצועי) וגם את הסמל המסחרי-UNIX- נראה שקלדרה לא משקיעה  
חלק גדול מהעבודה שלה בקהילת הלינוקס, וגם לא נראה שאכפת לה ממשתמשי לינוקס  
הביתיים, אז נראה לי שלא אשקול שימוש ביתי במערכת הזו.

ישנן גם גרסאות מקומיות ללינוקס למדינות מסויימות או שפות (קוראנית, סינית, יפנית ...)  
והן בדרך כלל מכילות (כברירת מחדל) את כל התיעוד שמשתמשים במדינות האלה רוצים  
לראות.

ביל סטילס מספר: "ייתכן שתמצאו להזכיר את ההפצה הברזילאית קונקטיבה, באופן כללי  
היא מבוססת על רד האט והיא זמינה בפורטוגזית ובספרדית".

בדוק ב <http://www.conectiva.com.br/>

שמעתי כמה דברים טובים על קונקטיבה, אז בטח הייתי משתמש בה אם פורטוגזית או  
ספרדית היו השפות שלי.

ישנן גם הפצות לינוקס בעלות "מטרות מיוחדות", מהדורות real time, (יכול להועיל אם  
אתה בעניין של אוטומציה, רובוטיקה, עיבוד מידע מהיר וכו').

הפצות לינוקס קטנות מאוד (אם מוצא חן בעיניך הרעיון של הרצת לינוקס מתוך פלופי יחיד  
שיכול להועיל לאבטחת המערכת או להתאוששות מערכת).

לינוקס למערכות מוטמעות (אם אתה רוצה לעצב את לינוקס למכשיר קטן ובעל מטרה  
"מיוחדת" או לטלפון סלולרי מרשים).

מיחשוב מקבילי ומערכות של אשכולות מחשוב (יכול להיות נהדר אם אתה מתכנן מערך חיזוי מזג אוויר משל עצמך, או מדמה פיצוצים גרעיניים למשל...).

במקרים האלו ההבדלים יהיו גדולים יותר, אבל הפצות אלו לא נבנו ל"מטרות כלליות" אני מאמין שכמשתמש חדש, תעדיף שלא להשתמש באחת מאלו, למרות שיכול להיות שתיתפתה לכך. הן ללא ספק מראות את העוצמה ואת השימושיות של לינוקס--לינוקס רצה על צעצועים, אפילו על שעון יד, כמו גם על אשכולות מיחשוב שבונים את המערכות הכי מהירות בעולם, נכון להיום.

ההפצה שאתה צריך חייבת להיות מותאמת לפלטפורמת החומרה שיש לך. זה אומר שאם יש לך חומרה שמכילה מעבד אינטל 386, או אינטל 486, או אינטל פנטיום, או אינטל 586, או אינטל 686, או סירקס, או קיי6, איי.אמ.די. או משהו דומה, אתה צריך את ההפצה הבינארית שנקראת "אינטל" או "386" "x86", אלא אם כן אתה מתכנן להכין אוסף הפצות מקור של לינוקס, (מה שבדרך כלל לא טיפוסית למשתמש חדש).

זה קורה כי ישנן הפצות בינריות לפלטפורמות חומרה אחרות, פאוורפיסי, אלפה, אפל, מחשבי מיינפריים של איי.בי.אם, אינטל סטרוג ארם, טרנסמטה, וכנראה גם לעוד תריסר-- אתה לא תרצה להשיג את הבינארים האלו לתואם פיסי שלך: הם בטוח לא יעבדו על פיסי עם מעבד אינטל או איי.אמ.די בפנים.

אם יש לך חומרה שהיא לא אינטל, יכול להיות שתצטרך לחפש באינטרנט כדי למצוא מי תומך בה.

רוב הסיכויים שדביין תומכת בה, נראה שהיא תומכת בחומרה האקזוטית ביותר, (אתה תצטרך להזמין, דבייןחזק או דביין מוטורולה 686\*0, או דביין פאוור פיסי, או דביין ספארק....).

בקיזור, אף על פי שמשתמשים חדשים מתבלבלים מכל ההפצות הרבות של לינוקס, ישנן מספר סיבות להחזקת הפצות שונות. צריך להסתכל עליהן כעל העוצמה של לינוקס ולא כעל חולשתה. לינוקס פשוט ממלאת את כל הנישות של האפליקציה והחומרה.

המדריך הזה מתרכז ברד האט ומנדרייק לפלפורמת הפיסי (אינטל). רבות מהתשובות יישומות גם לגבי הפצות אחרות או פלטפורמות אחרות, אבל לא ניסינו אותן.

באילו הפצות לבחור בשביל החומרה הישנה שלי? תשובה מהירה: דביין, סלייקוואר, או אולי בייסיק לינוקס (הגירסה העדכנית), או אולי גירסה ישנה של רד האט, מנדרייק או סוזה. ההצדקה: רד האט, מנדרייק, סוזה, קלדרה וטורבו לינוקס מותאמות להוציא את המקסימום מהחומרה העדכנית ביותר ליום הוצאתן. יכול להיות שזה יהיה קשה או בלתי אפשרי להתקין אותן על מכונות ישנות לרוב בגלל מגבלות של זכרון ומהירות. דביין וסלייקוואר מתאימות לרוב החומרה הישנה.

## 1.2 מה הן דרישות החומרה של לינוקס?

לינוקס מהקופסה אמורה לרוץ על פיסי מבוסס 386 עם 8 מגה זיכרון, אבל כזה מחשב מתאים רק לאפליקציות מבוססות טקסט בלבד (בלי X-windows).

486 עם 16 מגה זיכרון ו-600 מגה פנויים (לא מחולקים למחיצות) על הדיסק הקשיח יעבוד תחת איקס ווינדוס אבל אל תצפה ממנו שיטוס.

ה-586 שלי עם 133 מגה הרץ ו-64 מגה זיכרון רץ בצורה סבירה תחת לינוקס עם X. האתלון 1.3 ג'יגה שלי עם 256 מגה זיכרון הוא תענוג אמיתי, עם תגובה מיידית אפילו כשרצות עליו כמה אפליקציות בבת אחת. היום לא אקנה מחשב עם פחות מ-256 מגה זיכרון (דצמבר 2001).

ל-486 עם 33 מגה הרץ, 8 מגה זיכרון ודיסק של ג'יגה אחד יש מעט מדי זיכרון כדי לתפקד כראוי תחת משק גרפי, אבל הוא עדיין יעיל ברשת הפנימית שלי בתור טרמינל של איקס.

מכונה עם 486 מתפקדת כמו שצריך ובזכות עצמה רק אם יש לה לפחות 16 מגה זיכרון (אבל לפעמים קשה להשיג זכרון למחשבים ישנים במחיר שאתה חושב שהוא שווה). הנייד הישן שלי, טושיבה SX 386 עם מעבד 20 מגה הרץ, 9 מגה זיכרון ו-120 מגה דיסק קשיח מריץ "אפליקציות legacy" תחת ווינדוס 3.11, מתחבר לרשת הביתית שלנו ולכן עדיין יעיל. ניסיונו על הטושיבה הישן הזה גם לינוקס דביין, והוא רץ מצויין מצב טקסט. (בבקשה שים לב שמנדרייק דורשת מעבד פנטיום).

אם אתה מעוניין לקפוץ כמה משוחות, אתה אמור להצליח להתקין ולהריץ לינוקס על מחשב עם 4 מגה זיכרון, אבל זה בטח לא שווה את המאמץ בשביל מכונת לינוקס ביתית. לפי דעתי: 32 מגה זכרון, ואם זה אפשרי 128 או 256 מגה. יותר זכרון יכול לגרום להבדל בביצועים כשמריצים כמה אפליקציות ממשק גרפי גדולות ביחד. הזכרונות זולים היום. שים לב בבקשה שלהפצות רבות יש בעיות להריץ את תוכניות ההתקנה שלהן על מחשבים ישנים עם מעט זכרון (למרות שברגע שהיא הותקנה, לינוקס בדרך כלל תרוץ בסדר).

אם אתה זקוק לעוד עזרה לגבי התקנת לינוקס על מחשבים עם מעט זכרון, נסה את:

<http://7thguard.net/files/debianHOWTO.TXT>

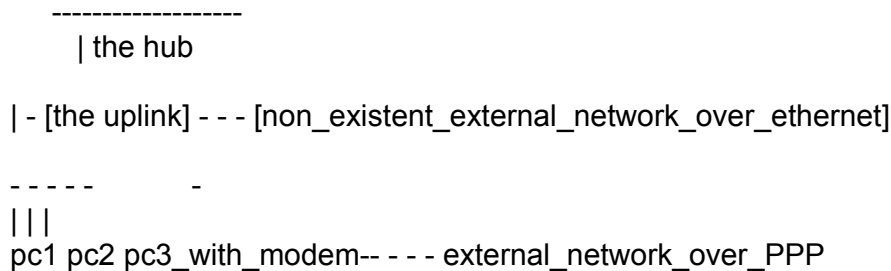
רשתות זה התחום שבו לינוקס באמת זורחת, אז תתחיל לחשוב על להשיג 10 כרטיסי רשת--הם לא יקרם מדי ויהיו מושלמים כדי לחבר ביחד את המחשבים שיש לך בבית. בנוסף נסה להשיג כרטיסי רשת ישנים שנחשבים למיושנים מדי תחת מייקרוסופט ווינדוס. ניתן לקנות אותם במחיר מאוד נמוך הם יעבדו טוב מאוד תחת לינוקס.

כדי לחבר שני מחשבים בלבד, כבל רשת מוצלב לחיבור ישיר יכול להספיק ("רשת לעניים"). כדי לחבר יותר משני מחשבים ביחד, תזדקק להאב (בין 30 ל-80 \$) וכבלים רגילים (לא מוצלבים). אם יש לך כרטיסי רשת מיותרים, אתה יכול להתקין יותר מכרטיס אחד על מחשב, השתמש בכלים מוצלבים, ותחסוך את ההוצאה על ההאב, אבל זה דורש דורש קינפוג מורכב יותר של המערכת.

מערכת-T-BASE 10 משתמשת במחברי רשת מסוג טלפון ענק (RJ45).

כל המכונות מחוברות לקופסה אחת שנקראת האב HUB.  
להאב יש חיבור נוסף (שנקרא אפלינק - UPLINK) שיכנס לשימוש אם אי פעם יהיה לי  
חיבור קבוע אחרי הרשת הפנימית לעולם החיצון.

הנה סידור סכמטי פשוט של רשת ביתית:



כאן אני מראה רשת מקומית פרטית שציגה את פיסה 1,2 ופיסה 3, מחוברים דרך האב. מכיון  
שלא חובר אצלי לאפלינק חיבור לרשת חיצונית, פיסה 3 מספק את התקשורת שלי לעולם  
החיצון (דרך מודם). לכן פיסה 3 נקרא- שער-GATEWAY- לכל המחשבים ברשת המקומית  
הפרטית שלי(פרט לפיסה 3 בעצמו). איפשרתי את תוכנת הפייר וול ב PPP של התקן הרשת  
בפיסה 3, וגרמתי לפיסה 3 לדעת איך לחייג החוצה. העולם החיצון יכול לראות רק את פיסה 3  
מבחינתם פיסה 1 ו 2 לא קיימים. הרשת המקומית שלי בטוחה כי רק לאנשים מורשים יש  
גישה פיסית לפיסה 1 ו 2 ופיסה 3.

PPP מתפרש כ POINT TO POINT PROTOCOL, והוא משמש לתקשורת על גבי קווי  
טלפון.

כמובן שאתה יכול לבנות רשת הרבה יותר מורכבת עם לינוקס. למחשב יכולים להיות שני  
כרטיסי רשת או יותר, והוא יכול לשמש כגשר בין שתי רשתות או יותר. המחשב ישמש  
כשער לכל התנועה בין שתי הרשתות. הרשתות לא חייבות להיות מוכרות לעולם החיצון  
("רשת מקומית פרטית") ולשבת מאחורי פייר וול שמאפשר על המחשב שמשמש כשער.  
העולם החיצון יידע רק על מחשב אחד, השער לרשת החיצונית. מחשבים אחרים עדיין  
יכולו לתקשר עם העולם החיצון, אבל כל התנועה תראה כאילו היא מגיעה ממחשב עמוס  
אחד-השער.

### 1.3 האם החומרה שלי תעבוד תחת לינוקס?

לא כל החומרה עובדת תחת לינוקס, למרות שרוב הציווד כן יעבוד תחת לינוקס, במיוחד  
חומרה, סטנדרטית, פופולרית או ישנה. זה נכון גם לגבי מתאמי סקאזי, סידי רום, צורבים,  
כרטיסי ווידאו, מיקרופונים, מדפסות, מודמים, כרטיסי רשת, סורקים, כונני זיפ וכו'.

יוצאי הדופן הם מודמי התוכנה ה WINMODEMS הימנע מהם כמו מאש, הם קצת יותר  
זולים מודמים מלאים, אבל הם גם קצת נכים (חלק מעבודת העיבוד מתבצעת על ידי המעבד  
של המחשב במקום על ידי המודם עצמו), ויש סיכוי נמוך שתצליח להפעיל מודם כזה עם  
לינוקס מיד עם ההתקנה.  
הסתכל ב: <http://www.idir.net/~gromtikc/winmodem.html> למידע נוסף.

מודמים חיצוניים אינם ווינמודמס, הם קצת יותר יקרים אבל לא גוזלים מהמחשב שלך חשמל ומשאבים הם מתניידים בקלות בין מחשבים ומראים את פעילות המודם .  
נקודות נוספות בנוגע למודמים:  
"מודמים חיצוניים ישנים שמשתמשים בפרורטוקול של רוקוול לא עובדים כל כך טוב. בנוסף , מודמים בעלי חיבור USB החדשים יותר, אינם נתמכים כרגע (מרץ 2001) בצורה טובה." .  
מקור-B. Staelhe .

תחום נוסף עם בעיות פוטנציאליות הוא כרטיס הווידאו. אם יש לך כרטיס חדיש מאוד עם תלת מימד או כרטיס לא קונבנציונלי, אתה יכול לבדוק את התאימות שלו ב-  
<http://www.xfree86.org>.

כונני זיפ מכל הסוגים נתמכים היטב.

לא הייתי סומך על לינוקס שתתמוך בסורק בעל חיבור מקבילי, ולא משנה אם הייצרן טוען לתאימות (TWIN (=Technology Without An Interesting Name - no joke here)).

התשובה הקצרה היא כן, מרוב הבחינות הפיסי הסטנדרטי שלך יריץ לינוקס בלי בעיות. אתה לא משקיע הרבה כשאתה מנסה את לינוקס, לכן הדרך הקלה ביותר הלבדוק היא להתקין לינוקס על החומרה הנוכחית שלך. ישנן רשימות תאימות חומרה ב:

<http://hardware.redhat.com/hcl/genpage2/cgi>  
<http://metalab.unc.edu/ldp/howto/harsware-howto.html>

אם אתה רוצה לבדוק את החומרה שלך.

עם רכישת חומרה חדשה, אני תמיד בודק את התאימות שלה ללינוקס ברשימות המזכרות למעלה. אתה תמיד יכול לשאול את ספק החומרה שלך אם החומרה נתמכת תחת לינוקס, אבל הייתי לוקח את התשובה בערבון מוגבל, לחברות רבות יש תמיכה טכנית/אנשי מכירות שלא מוכשרים לתפקיד.

הייתי שוקל לקנות מחשב שלינוקס כבר מותקנת עליו.

אם אתה אדם הרפתקני, כמוני, הייתי מתעלם מן ההערות האלו, הסיכויים הם 10-90 שהחומרה תעבוד.

אם חומרה שבבעלותך לא נתמכת תחת הפצת לינוקס הנוכחית, אל תרים ידיים. הסיכויים הם:

1. היא כן נתמכת, אבל אתה לא יודע איך להתקין אותה (פתרון: השאר עם לינוקס לכמה שבועות, אל תבזבז את הזמן, כשתהיה לך קצת הבנה לגבי איך המערכת שלך עובדת, אולי תוכל להתקין אותה).

2. אתה צריך לעבור תהליך התקנה מורכב יותר כדי לתמוך בחומרה (לתת איזו פקודה שאתה לא מכיר או לקמפל מחדש את הקרנל, מה שבדרך כלל לא קשה כמו שזה נראה).

3. הפצה מעודכת יותר (אחרת ??) תומכת בחומרה מהקופסה.

4. כבר קיים שידרוג איפה שהוא באינטרנט, תצטרך למצוא אותו, להוריד אותו, ולהבין איך להתקין אותו.



## 1.4 איך אני מוריד לינוקס?

עשה לעצמך טובה ואל תוריד את לינוקס מהרשת. קנה במקום זה דיסק התקנה. ניתן להוריד את לינוקס בשלמותה מהאינטרנט, אבל זוהי מערכת הפעלה גדולה מאוד ומתוחכמת מאוד. ההורדה יכולה לקחת שעות או ימים, ואתה יכול להיתקל בבעיות ותיסכולים בגלל טעויות בקבצים שאתה מוריד אם יש לך חיבור מהיר לאינטרנט (בהחלט לא מודם 56 אולי מודם כבלים) ואינך מתשתמש חדש מאוד הורדת לינוקס יכולה להיות אופציה.

נסה ב: <http://www.linuxiso.org/> לאימג'ים מוכנים לצריבה של הפצת לינוקס המועדפת עליך.

## 1.5 איך אני משיג דיסק של לינוקס?

(חפש קצת בפורומים הישראליים של לינוקס ובאינטרנט-אם יהיה לך רצון, אתה תמצא-א.א.)

## 1.6 יש לי את הדיסקים להתקנת לינוקס אבל אין לי את הפלופי להתקנה. מה אני עושה?

אם המחשב שלך תומך באיתחול מכונן התקליטורים (רוב המחשבים הישנים אינם תומכים) אינך זקוק לפלופי לאיתחול ההתקנה של לינוקס. בדוק את הביוס שלך: ניתן לקבוע את סדר האיתחול משם בדרך כלל. במחשב שלי כונן התקליטורים הוא מכשיר האיתחול הראשון בביוס ועדיין איני יכול לאתחל מהכונן הזה. ככה שהגדרת הביוס לא בהכרח משקפת את היכולות של המחשב שלך, אם אתה יכול לאתחל מכונן התקליטורים, כל מה שעליך לעשות בכדי להתחיל את תוכנית ההתקנה זה להכניס את הדיסק של רד האט אל הכונן ולאתחל את המחשב. במידה והמחשב שלך לא מאתחל מכונן התקליטורים, הכן דיסק התקנה תחת dos.

חשוב מאוד שיהיה לך פלופי תקין (ללא סקטור פגום אחד). התוכנית שמכינה את הפלופי לא בודקת אם הוא נכתב כראוי. בנוסף, אל תסמוך על תוכנת הפורמט של דוס שתמצא פלופי לא שמיש--סביר להניח שהיא לא. אם אני הייתי אתה, הייתי מניח שניים או שלושה פלופים בבת אחת--תופתע לדעת לכמה דיסקטים יש בעיות. אצלי הפלופי השלישי עבד! אם דיסקט ההתקנה שלך לא מאתחל, הכן אחד נוסף--הוא אמור לעבוד.

הנה הפקודות. כדי להריץ את הפלופי:

```
F:\dosutils\rawrite.exe -f F:\images\supp.img -d a: -n
```

כדי להריץ את הדיסק הנוסף (אופציונלי):

```
F:\dosutils\rawrite.exe -f F:\images\supp.img -d a: -n
```

בהנחה שכונן התקליטורים שלך תחת דוס הוא: F, והפלופי הוא: A, התאם את הפקודות במידה והכוננים מופיעים אחרת במערכת שלך.

2 הפקודות למעלה מריצות את היוטיליטי שנקרא "rawrite"-ומציינות את מיקום קובץ הקלט, תעתיק הדיסק תחת האות f, את הכונן המיועד אחרי האופציה d, ועל ידי האופציה n מורה להכניס פלופי כיעד.

יכול להיות שתמצא שקל יותר להריץ את rawrite- ללא כל ארגומנטים--אתה תתבקש לקבוע את האימאג' ואת אות הכונן המיועד.

תחת לינוקס, אתה יכול להכין פלופי על ידי המרת הדיסק של רד האט והקלדת הפקודות הבאות (כ root):

```
cd /mnt/cdrom/images/
```

```
dd if=boot.img of=/dev/fd0
```

הפקודה למעלה מציינת ש: קובץ הקלט if הוא boot.img. ושקובץ הפלט "of" הוא /dev/fd0 שהוא למעשה הפלופי. הפקודה "dd" מעתיקה קבצים.

אם יש לך כונן פלופי נוסף ואתה רוצה לכתוב אליו השתמש ב1 בסוף ולא ב0.

## 1.7 מה עליי לקרוא לפני ההתקנה?

עוזר מאוד לקבל קצת אוריינטציית יוניקס אם אין לך בכלל. קנה מדריך לינוקס טוב או שתסיר את האבק ממדריך היוניקס הישן שלך. כמעט כל העקרונות של יוניקס מוחלים על לינוקס, וכמעט כל הפקודות של יוניקס ירצו טוב תחת לינוקס. אני מוצא את המדריכים למייקרוסופט ווינדוס כחסרי תועלת (תקליק כאן, תקליק שם, הסתכל בצילומי המסך), לעומת זאת מדריכי לינוקס/יוניקס בדרך כלל טובים (מקנים לך הבנה לגבי המערכת, וזה ילווה אותך לאורך זמן).

יכול מאוד להיות שתמצא ללמוד על החומרה שלך: כמה ומה גודל הדיסק הקשיח שלך, הסוג, המספר סדר וגודל כל המחיצות בכל כונן, היכן ממקומות מחיצות הדוס/ווינדוס שלך, מי מהן היא מחיצת האיתחול של דוס/ווינדוס (אם אתה מתכנן איתחול כפול) מהו סוג העכבר שלך, איזה כרטיס מסך יש לך וכמה זיכרון יש לו, איזה מסך (תדירויות סינכרון מקסימליות) וכו'.

גש אל הביוס ובדוק את המספר והגאומטריה של הדיסק הקשיח שלך. הרץ את fdisk תחת דוס כדי להציג את טבלת המחיצות שעל הדיסק, והדפס אותה. התבונן באיתחול המערכת שלך כדי ללמוד על סוג כרטיס המסך שלך ועל כמות הזיכרון שלו. אתחל את מייקרוסופט ווינדוס, גש אל מנהל ההתקנים בלוח הבקרה ורשום לעצמך איזה כרטיס קול, מודם, כרטיס רשת יש לך ומה ההגדרות שלהם (שם, סוג, irq, כתובות קלט/פלט, ערוץ dma, קרא מה רשום על התיות מתחת לעכבר שלך כדי לבדוק מה סוג העכבר שלך (בפעם הבאה שתקנה עכבר, קנה לוג'יטק עם 3 כפתורים או משהו דומה -- לינוקס עושה שימוש טוב בכל שלושת הכפתורים) בדוק בחוברת שהגיעה עם המסך שלך מה הן תדירויות הסינכרון המקסימליות (אנכי ואופקי) שבהן המסך שלך תומך.

לעולם אל תשתמש בתדירויות שחורגות מעבר ליכולות המסך שלך -- זה עלול לפגוע בו.

ייתכן שתמצא לדפדף קצת בחבורות של רד האט או של מנדרייק. אם אין ברשותך עותק מודפס, גירסת html נמצאת על הדיסק שלך אז אתה יכול לנסות לקרוא אותה על יד דפדפן.

### 1.8 האם אני יכול להתקין מייקרוסופט ווינדוס ולינוקס על אותו מחשב?

כן אתה יכול. לינוקסאים רבים משתמשים באיתחול כפול. ניתן לעשות את זה על ידי התקנת ווינדוס על מחיצה אחת והתקנת לינוקס על אחרת.

לינוקס מגיעה עם מנהל איתחול פשוט בשם LILO או אחד מתחומם יותר בשם GRUB שיאפשרו לך לבחור, בזמן האיתחול את מערכת ההפעלה שאתה מעוניין לאתחל.

קודם כל התקן את מייקרוסופט ווינדוס ואחר כך את לינוקס, אחרת ההתקנה של ווינדוס תבטל את הגישה שלך ללינוקס. תכין לעצמך דיסקט איתחול למקרה שתצטרך להתקין את ווינדוס מחדש--ווינדוס בטוח תבטל את הגישה שלך ללינוקס, אתה תצטרך לאתחל את לינוקס מהדיסקט ולהריץ שוב את הפקודה lilo כדי לחזור לעינינים.

תחת לינוקס, אתה תוכל לקרוא ולכתוב למחיצות של ווינדוס כך שהעברת הנתונים בין ווינדוס ללינוקס הינה שקופה. בנוסף תוכל גם להשתמש במשאבים מבוססי ווינדוס: קבצי קול, רקעים, תמונות, פונטים וכו' (רצוי שתבדוק לפני כן שאינך מפר את הסכם הרישוי, עבור מוצרים שאתה שוכר ממיקרוסופט סביר להניח שאתה כן. עם לינוקס, אני גאה בעובדה שאין שום תוכנה פיראטית במערכת שלי).

### 1.9 איך אני מחלק למחיצות את הדיסק הקשיח שלי?

לפני התקנת לינוקס, תרצה לדעת מה זה בכלל מחיצה על דיסק קשיח. הדאגה היא שתמחק את מחיצת הווינדוס שלך כשאתה למעשה לא חפץ בכך, אתה מעוניין בשתי מחיצות שונות כדי שתוכל ליישם איתחול כפול. זה אומר: מייקרוסופט ווינדוס על מחיצה אחת, לינוקס על השניה. בדרך כלל לא מתקינים לינוקס על שטח פנוי במחיצה של ווינדוס, למרות שזה אפשרי, אנחנו לא ממליצים על זה.

אם אתה מתכנן איתחול כפול ראשית השתמש בfdisk תחת דוס כדי ליצור את המחיצות של ווינדוס, השאר חלק מהדיסק לא מחולק למחיצות בשביל לינוקס.

אתה תכין ותפרמט את המחיצות בזמן התקנת רד האט (או מנדרייק או מה שלא יהיה). לינוקס תזהה את השטח הפנוי על הדיסק הקשיח.

הפוך את מחיצת הווינדוס שלך למחיצת האיתחול ולראשית. התקן, קנפג ובחן את הווינדוס שלך לפני התקנת לינוקס. אם אתה מתכנן להתקין לינוקס בלבד, כל מה שאתה צריך זה דיסק קשיח נקי, ללא מחיצות כדי להתחיל.

ניתן ליצור רק מחיצת לינוקס אחת אם אתה מעוניין בכך ( בנוסף למחיצת הווינדוס אם אתה משתמש באיתחול כפול ), אבל כדאי שיהיו לך עוד כמה מחיצות כדי שתוכל לשמור נתונים של משתמשים בנפרד משאר מערכת ההפעלה. ככה, אם משהו מסתבך, או אם פירמטת מחדש והתקנת מחדש את מערכת ההפעלה, אינך מאבד נתונים של משתמשים אתה יכול לבצע התקנה מחודשת של לינוקס בלי לאבד את התוכן של /home, הסיפרייה שמכילה את כל הנתונים של המשתמשים, במידה ואתה מדלג על אופציית הפירמוט מחדש. אבל כדי להשיג את זה ספריית /home צריכה להיות על מחיצה משלה.

בזמן התקנת לינוקס, אתה תתבקש לחלק למחיצות את השטח הפנוי שעל הדיסק הקשיח שלך. ישנן דרכים רבות לחלק למחיצות את הדיסק, תלוי בגודל של הדיסק הקשיח שלך, בדרישות ובהעדפה שלך, אני אוהב מחיצות לינוקס כאלו: (למודל צנוע של דיסק עם שני ג'יגה שנתתי ללינוקס בדוגמא הזו):

mount point	type	size
/	ext2	300MB
/usr	ext2	1200MB
/home	ext2	380MB
swap	swap	120MB

בדוגמה זו, ייעדתי 300 מגה למחיצה שמחזיקה את הבסיס למערכת ההפעלה לינוקס, המחיצה הראשית the root partition, בנוסף הקדשתי 1200 מגה לנקודת המרה שתיראה במערכת שלי כספריית ה /usr אשר תכיל אך ורק את תוכניות ששיכות למשתמש (תוכניות שלא מגיעות עם המערכת הבסיסית ושאינן אתקין בשלב מאוחר יותר, כמו start office).

הקדשתי גם 380 מגה למחיצה שתיראה כ /home אשר תכיל את ההגדרות והנתונים לגבי כל המשתמשים במערכת. ולבסוף נתתי עוד 120 מגה שישמשו את המערכת לזיכרון וורטואלי (ההארכה של זיכרון הסיליקון הפיזי, על גבי הדיסק הקשיח, שנקרא גם swap).

אם הקרנל שלך נמוך מ-2.2 (זוהי המקרה ברד האט 5.2 סטנדרטי וכל מה שלפניו), מחיצת הסוואפ שלך לא יכולה להיות גדולה מ-127 מגה. חוק האצבע הוא שמחיצת הסוואפ צריכה להיות פעמיים הזיכרון הפיזי אם אתה מעוניין ביותר (אם יש לך הרבה זיכרון פיזי, או שאתה מתכוון להריץ תוכניות עם מבני נתונים גדולים) יכול להיות שתרצה ליצור מחיצת סוואפ גדולה יותר בזמן ההתקנה (או כמה מחיצות סוואפ קטנות יותר) או להוסיף קובץ (ים) סוואפ מאוחר יותר.

שני ג'יגה זו כמות מכובדת של שטח דיסק שאמורה להספיק למשתמשים שאוהבים אפליקציות רבות. זה מתאפשר מכיון שאפליקציות של לינוקס נוטות להיות רזות יותר מהמקבילות המייקרוסופטיות שלהן) בכל מקרה, אם תנסה להתקין את כל מה שזמין על גבי הדיסקים של ההפצות המודרניות, ייגמר לך המקום בדיסק הקשיח. הניסיון שלי לימד אותי שלא משנה עד כמה גדול הדיסק, הוא תמיד מתמלא ואני מתחרט על זה שאין ושלא רכשתי דיסק גדול יותר.

במידה והשטח על גבי הדיסק הקשיח שלי מוגבל אני שוקל הקמת שתי מחיצות כמו בדוגמא הזו (למחיצה דלה של 650 מגה שמיועדת ללינוקס):

mount point	type	size
/	ext2	600MB
swap	swap	50MB

כאן הקדשתי 600 מגה לבסיס של מערכת ההפעלה, אפליקציות, ולנתונים/מסמכים של

המשתמש והקצתי 50 מגה למחיצת הזיכרון הווירטואלי, מחיצת הסוואפ. 50 מגה סוואפ אמורים להספיק לפעולות הכרחיות בעלות משקל בינוני. מגבלת ה-600 מגה למערכת ההפעלה, אפליקציות ונתוני משתמש תכריח אותך להיות בררני ביותר בבחירת האפליקציות שתתקין או שתסתכן בכך שייגמר לך המקום על הדיסק.

נסה ללחוץ על <F1> כשאתה מתקין תוכנה אופציונלית שמגיעה על גבי הדיסק של רד האט אתה תקבל תיאור קצר לגבי מה התוכנה עושה וכך תוכל להחליט אם אתה באמת זקוק לה. (אל תדאג יותר מדי אם החמצת משהו שאתה צריך, תמיד תוכל להתקין את החלקים החסרים מאוחר יותר). אתה יכול לסיים את בקלות את ההתקנה של רד האט עם 200 מגה פנויים על הדיסק הקשיח (מתוך 600 בדוגמא הזו) במידה ואתה בוחר בחירות הגיוניות. שים לב, ש"הדבקה" של מחיצת ה-root/ ושל מחיצת ה-home/ ביחד, אולי תחסוך לך מקום, אבל היא אינה הפתרון הבטוח ביותר.

ניתן להתקין לינוקס על גבי שטח דיסק קטן יותר מאשר בדוגמא למעלה, אבל תצטרך להיות בררן מאוד לגבי מה שאתה מתקין.

לדיסק בעל נפח גדול יותר, הייתי שוקל את ההתקנה הבאה (ל-15 מגה מפנקים וייעודיים ללינוקס):

mount point	type	size
/	ext2	800M
/usr	ext2	5000MB
/usr/local	ext2	3000MB
/home	ext2	5200MB
swap	swap	1000MB

אנא שים לב שנקודות ההמרה יכולות לשכון על גבי דיסקים קשיחים שונים. לינוקס אוספת את כל שטח הדיסק אל תוך אשכול אחד.

נקודה נוספת שצריך לקחת בחשבון כשמקימים מחיצות:

ביוסים ישנים רבים מגבילים את מחיצת האתחול ככה שהיא לא יכולה להמשיך מעבר לצילינדר ה-1024 על גבי הדיסק הפיזי הראשון. כדי להתגבר על המגבלה הזו, כל מה שעליך לעשות זה ליצור את מחיצת האיתחול הראשונה שלך כך שהיא תסתיים לפני צילינדר מספר 1023 (זה גורם לכך שהמחיצה תהיה בגודל של 512 מגה בערך שזה המון ל-root/). ברגע שלינוקס תרוץ, למגבלת הביוס כבר לא תהיה כל חשיבות מכיוון שלינוקס תשתלט על ניהול החומרה והיא דווקא כן יכולה לגשת אל מחיצות שמעבר לצילינדר מספר 1023.

כשמתקינים ומשתמשים בלינוקס, הכוננים שלך מופיעים כהתקנים בעלי השמות הבאים:

hda -- כונן ide הראשון (מתפרש כ"hard drive a", כונן המאסטר בממשק ide הראשון).

hdb -- כונן ide השני (הכוונה היא כאן לכונן ה-slave בממשק ה-ide הראשון).

hdc--כונן ide השלישי (כונן המאסטר בממשק ide השני).

hdd--כונן ide הרביעי (slaven בממשק ה ide השני).

המספרים מצביעים על מחיצות בתוך בדיסק הקשיח:

hda1 -- המחיצה הראשונה ב hda.

hda2-- המחיצה השנייה ב hda.

hda3 המחיצה השלישית ב hda.

וכן הלאה, אם יש לך יותר מחיצות בכונן הראשון:

hdb1-- המחיצה הראשונה שבכונן השני או רק hdb אם הסידי רום מותקן כסלייב בממשק ide הראשון.

hdc1- המחיצה הראשונה בכונן השלישי, וכן הלאה.

לכונני סקאזי יש שמות מקבילים אבל הם מתחילים באותיות "sd", "SCSI drive=" כשבהמשך מופיע מספר שמצביע על הממשק ומספר שמצביע על ההתקן.

sda4 למשל מתפרש ככונן הסקאזי הראשון, בעל מספר הזהות 4.

במידה ויש לך כונן זיפ חיצוני שמחובר לחיבור המקבילי של המחשב שלך, הוא יופיע כהתקן סקאזי sda4- כונני זיפ עובדים במצב שמדמה סקאזי.

רשימת המחיצות שמוצגת לפניך בתוכנית ההתקנה של לינוקס תכלול את כל המחיצות של מייקרוסופט ווינדוס שיש לך. לדוגמא, לי יש את מחיצות ווינדוס הבאות:

mount point	type	size	comment
[no mount]	vfat	1200MB	["win C drive, hda1]
/mnt/dos_hdd2	vfat	1600MB	["win D drive' hdd2]

אל תמחק את מחיצות ווינדוס הללו בזמן התקנת לינוקס אם אתה מעוניין באיתחול כפול. אם תמחק אותן לא יהיה לך עם מה ליישם איתחול כפול! במידה ואינך בטוח, גבה את המידע שנמצא כל גבי מחיצות אלו לפני התקנת לינוקס.

ווינדוס NT/98/95/3.x "ntfs" "vfat" "fat" & "msdos" הינן מערכות קבצים שנמצאות בדרך כלל בשימוש תחת דוס

כמבוא מהיר, לפניך סיכום זריז של סוגי מערכות הקבצים ("file systems") בלינוקס מלווה בתיאור קצר.

העתקתי את המידע מתוך חוברת ההסברים של לינוקס ("linux manual pages"):

על ידי שימוש בפקודות `man fs` ו `man mount` (ביחד עם כמה הערות שהוספתי לאחר שהסתכלתי בקבצי קוד המקור ב `/usr/src/linux/fs`).

מערכות הקבצים שמוּדגשות הן אלו שסביר יותר שתשתמש בהן. מערכות הקבצים אחרות (אינן רשומות למטה) זמינות כתוספות (כמו מערכות קבצים דחוסות, מוצפנות וכו').

-minix

הינה מערכת קבצים אשר נמצאת בשימוש תחת מערכת ההפעלה Minix הראשונה שרצה תחת לינוקס. יש לה מספר מגבלות: גודל מחיצה של 64 מגה בלבד, שמות קבצים קצרים, וחותמת זמן (time stamp), אחת וכו'. היא עדיין שימושית בפלופי וב RAM disks.

extended filesystem-ext

זוהי הארכה מפותחת של מערכת הקבצים מיניקס. היא דוכאה לחלוטין על ידי הגירסה השנייה של מערכת הקבצים המוארכת (ext2), ולבסוף תוסר מהקרנל.

ext2-היא מערכת קבצים בעלת ביצועים גבוהים שנמצאת בשימוש על ידי לינוקס לדיסקים קבועים ולמדיה שניתנת להסרה. מערכת הקבצים המורחבת השנייה עוצבה כהרחבה של מערכת הקבצים המוארכת (ext).

ext2 מציעה את הביצועים הגבוהים ביותר (במושגים של מהירות ושימוש במעבד) שמערכת קבצים שנתמכת תחת לינוקס יכולה להציע. בקיצור ext2 היא מערכת הקבצים העיקרית והדיפולטית של לינוקס.

ext3-הרכבה של מערכת הקבצים ext2 עם ג'ורנלינג (journaling). יש לה תאימות אחורה וקדימה עם ext2- ללא כל צורך בפירמוט מחדש או אובדן נתונים. (יש צורך אך ורק בהמרה מחדש של החלק המבוקש).

ext3-יכולה להיות מומרת בקלות ל ext2- גם כן ללא אובדן של נתונים או מידע. אני משתמש ב ext3 החל מאוקטובר 2001 והיא פשוטה ונטולת צרות. היא כלולה כאופציה בהתקנה מאז רד האט 7.2 ו מנדרייק 8.0. מומלץ מאוד שתשתמש במערכת קבצים זו.

xiafs-עוצבה והוטמעה על מנת להיות מערכת קבצים יציבה ובטוחה על ידי הרחבת הקוד של מערכת הקבצים מיניקס. היא מספקת את היכולות המבוקשות ביותר ללא מורכבויות מיותרות. מערכת הקבצים xiafs כבר אינה מפותחת או מתוחזקת. משתמשים בה לעיתים נדירות.

msdos- זוהי מערכת קבצים שנמצאת בשימוש תחת דוס, ווינדוס וכמה מחשבי os/2. שמות הקבצים שלה לא יכולים להיות יותר גדולים מ 8 תווים עם תוספת אופציונלית של רווח 31 תווים.

umdos-היא מערכת קבצי דוס מורחבת שנמצאת בשימוש על ידי לינוקס. היא מוסיפה יכולות של שמות קבצים ארוכים, UID/GID, הרשאות POSIX, וקבצים מיוחדים (התקנים וכו') תחת מערכת הקבצים דוס, ללא הקרבת תאימות לדוס.

vfat-הרחבה של מערכת הקבצים דוס שבשימוש מייקרוסופט מאז ווינדוס 95 ווינדוס .NT

vfat מוסיפה יכולות לשמות קבצים ארוכים תחת מערכת הקבצים MSDOS.

proc-מערכת קבצים פסבדונית שנמצאת בשימוש כממשק למבני נתונים של הקרנל יותר מאשר קריאה ופיענוח של `/dev/kmem`.  
הסתכל ב `man 5 proc`.

iso9660 - מערכת קבצים של CD-ROM שבנויה לפי הסטנדרט ISO 9660.  
שתי הרחבות (רשומות למטה) נתמכות אוטומטית.

high sierra - לינוקס תומכת במקדימה זו של הסטנדרט ISO 9660  
היא מוכרת באופן אוטומטי בתוך תמיכת לינוקס בסטנדרט הנ"ל.

rockridge  
לינוקס תומכת גם בפרוטוקול השיתוף שמונחה על ידי rock ridge interchange protocol  
הוא משמש על מנת לתאר את הקבצים במערכת ISO 9660 למארח יוניקס, ומספק מידע  
כמו שמות קבצים ארוכים, UID/GID, הרשאות POSIX, והתקנים. היא מזהה  
אוטומטית בתוך מערכת הקבצים ISO 9660, אשר נתמכת תחת לינוקס.

high performance filesystem-hpfs  
בשימוש ב `os/2` מערכת קבצים זו נקראת בלבד תחת לינוקס עקב מחסור בתיעוד זמין.

sysv - הטמעה של System V/coherent filesystem ללינוקס. היא מיישמת את כל  
מערכות הקבצים של SystemV/386 & Coherent, xenix.

nfs-מערכת קבצים שמשמשת לגישה אל כוננים אשר ממוקמים על גבי מחשבים מרוחקים.

smb - מערכת קבצים של רשת שתומכת בפרוטוקול SMB.  
פרוטוקול זה נמצא בשימוש על ידי קבוצות עבודה של מייקרוסופט ווינדוס ווינדוס NT,  
ו Lan Manager. כדי להשתמש במערכת קבצים זו, אתה זקוק לתוכנית המרה מיוחדת  
שניתן למצוא ב `ftp://sunsite.unc.edu/pub/Linux/system/Filesystems/smbfs/`.  
הפקודה `smbmount` גם תספיק.

ncpfs - מערכת קבצים של רשת אשר תומכת בפרוטוקול NCP שנמצא בשימוש על ידי  
רשת Nvll NetWare.

devpts--הינה מערכת קבצים פסבדונית, שבאופן מסורתי מומרת ב `/dev/pts`.  
בכדי להשיג פסוודו טרמינל, פרוסס (process) פותח את `dev/ptmx`.  
מספר הפסוודו טרמינלים נהיה זמין לפרוסס, והסלייב (slave) פסוודו טרמינל הופך להיות  
גיש כ `/dev/pts/<number>`.

fat - אינה מערכת קבצים נפרדת, אלא חלק ממערכות הקבצים `msdos`, `umdos`, & `vfat`.

UFS - מערכת קבצים שנמצאת בשימוש רחב במערכות הפעלה שונות.

swap-סוג מיוחד של מחיצה שמשמשת לגריפת נתונים מהזיכרון אל הדיסק הקשיח.

raiseffs - מערכת ג'ורנליזים חדשה לגמרי, זמינה כסטנדרט עם לינוקס מקרנל גירסה  
2.4.1 ומעלה.  
(יונאר 2001)



(hierarchical file system)=hfs מערכת הקבצים של מקינטוש. זוהי גירסת בטא מאוחרת. לא ממולץ לעבודה עם נתונים חשובים, אלא אם כן כקריאה בלבד.

ntfs מערכת הקבצים של מייקרוסופט ווינדוס-NT. עדיין נסיונית תחת לינוקס, לא ממולץ למכונות יצרניות, אלא אם כן כקריאה בלבד. (אוגוסט 2001)

### **1.10 המחיצה של מייקרוסופט ווינדוס תופסת לי את כל הדיסק הקשיח. האם אני יכול לכווץ/לחלק אותה בלי להתקין מחדש?**

ייתכן. ישנו כלי שנקרא `fipes.exe` על גבי הדיסק של רד האט שיש לי שעושה בדיוק את זה. בדוק את הסיפרייה `\dosutils\fipsdocs\` בדיסק של רד האט שיש לך אם אתה מעוניין בתיעוד.  
אם אני הייתי במקומך, הייתי מגבה את כל המידע הקריטי שאני מחזיק לפני שהייתי עושה משהו למחיצות שלי.  
ישנם גם כלים מסחריים שמאפשרים לשנות את גודל המחיצה בלי לפגוע בתוכן שלה. ההעדפה האישית שלי היא לבצע התקנה נקייה של ווינדוס על מחיצה אחת ייעודית.  
אני משאיר קצת מקום לא מחולק למחיצות על הדיסק כדי שאוכל להשתמש בו מאוחר יותר בשביל לינוקס. החלוקה ההוגנת של דיסק קשיח בין ווינדוס לבין לינוקס היא 50/50.  
תוכניות של לינוקס נוטות להיות קטנות יותר אבל הן כוללות (כסטנדרט) רכיבים שמייקרוסופט ווינדוס מציעה רק תמורת דולרים רבים שרתים (לא רק קליינטים) ל: `http`, `ftp`, `telnet`, דואר, כמה מסדי נתונים, שפות תיכנות, תוכניות עיבוד גרפי...

### **1.11 איך אני מתחיל את תהליך ההתקנה?**

הכנס את דיסקט האיתחול אל תוך כונן הפלופי שלך, את התקליטור של רד האט אל תוך כונן התקליטורים שלך, ואתחל את המחשב. אם אתה יכול לאתחל מכונן התקליטורים, הכנס את התקליטור של רד האט אל כונן התקליטורים שלך ואתחל את המחשב.

תוכל להתחיל את ההתקנה גם מדוס (או תחת מצב דוס בווינדוס) בדיסק של רד האט שיש לי הרצתי את הפקודה `ezstart.bat`.

רוב ההתקנות מציעות לך לבחור בין מצב גרפי למצב טקסט. אתה צריך לבחור בטקסט אם המחשב שלך מוגבל בזיכרון.

### **1.12 האם ההתקנה של לינוקס קשה?**

היא לא הייתה קשה בשבילי. נראה שלרוב המשתמשים החדשים, היא די עינינית ונטולת כאבים. זה גם תלוי בחומרה שלך ובבחירות ההתקנה, בדרך כלל היא נמשכת בין 40 דקות לשעתיים [פרט להתקנות ארוכות או בעיות עם מערכות ישנות בעלות זיכרון מוגבל מאוד-- לקח לי לילה שלם להתקין רד האט 6 על ה-33-486 מגה הרץ שלי עם 8 מגה זיכרון `ram`, המערכת עבדה ונראה שהיא לא עשתה כלום אבל בכל זאת ההתקנה הצליחה] שידרוגים מגרסאות קודמות אורכים יותר זמן ונוטים להיות יותר בעייתיים.

כמה משתמשים חדשים דיווחו שההתקנה הייתה סיוט טוטאלי בשבילם (בעיות חומרה? חוסר נסיון?) אם אתה נתקל בבעיות העצה שלי היא להתקין מערכת פשוטה, בלי להילחם על הרזולוציה הגבוהה ביותר לכרטיס המסך המפואר שלך או חומרה מתקדמת מאוד שיש לך. ניתן להוסיף/לקנפג הכל בשלב מאוחר יותר, אחרי שתהיה לך יותר הבנה לגבי איך הדברים עובדים על המערכת שלך. אפילו התקנה מחדש היא אופציה למשתמש חדש (מספרים שגורואים של לינוקס חושבים שזו בושה להתקין מחדש).

הרבה משתמשים חדשים נתקלים בבעיות כי הם קובעים רזולוציה גבוהה מדי (שאינה נתמכת או שנתמכת רק עם כיוון נוסף). שוב, זה לא יהיה חכם לשבור התקנה שלמה בשביל תמיכה בהתקן אחד ניתן לתמוך במכשיר ולכוון אותו מאוחר יותר.

### 1.13 אילו חבילות עליי להתקין?

תוכנה של לינוקס מגיעה בחבילות. לדוגמא, הדיסק של לינוקס מנדרייק שיש לי מכיל 1002 חבילות. מנדרייק 7.2 אורזת שני דיסקים של תוכנה--ההתקנה שלי של מנדרייק 7.2 הניחה 1123 חבילות על גבי הדיסק הקשיח שלי. רק שתדע, זו לא כל התוכנה שזמינה ללינוקס--אלא רק מבחר של חבילות שאנשי מנדרייק שמו בהפצה. מנדרייק נוטה לארוז יותר תוכנה מאשר רד האט.

לא משנה באיזו הפצה או גירסא, הדיסק מכיל חבילות שהופכות את מערכת ההפעלה הבסיסית (קרנל, ספריות, מבחר של כלים של שורת הפקודה לקינפוג ותחזוקה וכו') למבחר עשיר של קליינטי ושרתי רשת ביחד עם כלי הקינפוג והתחזוקה המתאימים, כמה אפליקציות במצב טקסט, מערכת X-WINDOWS בסיסית, לפחות ממשק גרפי אחד (סביר להניח שכמה ממשקים), וכמות גדולה של אפליקציות בעלות ממשק גרפי.

תוכנית ההתקנה (מנדרייק או רד האט) תשאל אותך אילו חבילות להתקין. אם תבחר בהתקנת תחנת עבודה, אז החבילות שבדרך כלל נמצאות בשרתים יישמטו מההתקנה שלך (למשל, שרת הווב אפאצ'י) אם תבחר בהתקנת שרת, אז בדרך כלל אפליקציות שולחן העבודה לא יותקנו (כמו האפליקצייה הגרפית THE GIMP) תוכל גם להתקין את הכל, וזוהי הבחירה המועדפת עליי להתקנה ביתית.

לבסוף, תוכל לבחור את מבחר החבילות שאתה רוצה להתקין. המשך לקרוא.

זה בהחלט רעיון גרוע לבחור התקנת חבילות/תוכניות לפי עד כמה השם שלהן נשמע לך מעניין-לכמה מהחבילות יש שמות לא רגילים בכלל ולעולם לא אצליח לנחש מה הן עושות. תוכל לגרום לנכות של המערכת שלך על ידי השמטת חבילות חיוניות (ספרייה). ייתכן גם שתתאכזב כשתתקש להריץ אפליקציה עם שם מיוחד, שבמקרה כלולה בהפצה. באופן כללי, יכול להיות שתתעצבן מהפונקציונליות (או מאי הפונקציונליות) שהלינוקס ה"מותאמת" שלך מציגה. כמשתמש חדש, זה משתלם לעיתים לבטוח בבחירות הדיפולטיות שאורגנו על ידי יוצרי ההפצה שלך.

לכן, להתקנה הסופית שלי, הייתי נמנע מלהתפתות לבחור באפשרות בחירה ידנית של חבילות אלא אם כן אני מעוניין להתקין הכל. למתחילים, אני מעדיף את "ההתקנה הדיפולטית המקסימלית", או איך שהאופציה הזו לא נקראת בדיסק שלך.

## 1.14 איזה ממשק גרפי עליי להתקין KDE או GNOME?

במידה ויש לך מקום על הדיסק, את שניהם ללא ספק. מאוחר יותר תוכל להחליט אם אתה מעדיף את KDE, GNOME, או שולחן עבודה אחר, מה שלא תהיה הבחירה שלך, בטוח שתמצא את הספריות של שניהם מותקנות. ברגע שהספריות מותקנות, ניתן להריץ תוכניות של KDE תחת GNOME ולהיפך, שזה נהדר מכיוון שישנן אפליקציות נחמדות מאוד לשתי הספריות.

מבחינת שימוש בשטח דיסק, שולחן העבודה הוא רק חלק קטן מהמערכות KDE ו-GNOME ככה שאתה לא חוסך הרבה מקום על ידי השמטת שולחן העבודה מההתקנה ומתקין את הספריות בלבד. גם GNOME וגם KDE מגיעים עם מבחר נחמד של תוכניות וכלים, לכן בהחלט שווה להתקין את שניהם במלואם. לא שמעתי מעולם שהם עבדו בצורה שמפריעה או סותרת אחד את השני. לשימוש יום יומי, אני משתמש בשולחן העבודה של KDE מכיוון שהוא מרגיש יותר מוצק מ-GNOME. אם אתה רוצה ממשק יותר COOL בחר ב-GNOME אבל אל תתלונן אם הדברים לא עובדים כל כך טוב.

בנוסף הייתי מתקין גם "מנהלי חלונות" נוספים. הם בקושי תופסים מקום על הדיסק (חלק ממש זעירים) ועדיין הם יכולים להיות שימושיים בנסיבות מסוימות. אתה יכול להריץ כל אפליקצייה של KDE או של GNOME תחת כל אחד מהם, כל עוד הספריות של KDE ושל GNOME מותקנות.

KDE "רעב" יותר למשאבים.

על גבי חומרה ישנה (פנטיום 133 MHz) אני מעדיף את GNOME על פני KDE. מנהלי חלונות אחרים קלים יותר מ-KDE או מ-GNOME.

לכן, על גבי חומרה צנועה באמת הייתי בוחר במנהלי החלונות ה"אחרים".

## 1.15 סיימתי את ההתקנה. איך אני נכנס למערכת בפעם הראשונה?

כ root :

```
my_machine_name login: root
password: my password
```

בדוגמא למעלה, הקלדתי את המילה root בבקשת ה login. לאחר מכן, הכנסתי את הסיסמא שבחרתי במהלך ההתקנה. הסיסמא לא הופיעה על המסך כשהקלדתי אותה (אבטחה). לאחר שנכנסתי למערכת, אני מוצא את עצמי במסוף טקסט.

אם התקנת את מסך הכניסה הגרפי, תהליך ה login נראה דומה, אבל מתרחש על גבי מסך x-windows (אם אתה נתקל כאן בבעיות הקלדה, זכור שסמן העכבר צריך להיות מעל קופסת הדיאלוג. מסך ה login x מיישם מדיניות של "פוקוס הולך בעקבות העכבר"). לאחר כניסה מוצלחת למערכת, הממשק הגרפי הדיפולטי שלי משוגר.

root הוא חשבון מיוחד עם כח אבסולוטי לגבי המערכת, והוא משמש לניהולה. כדאי שתיצור חשבון נוסף מאוחר יותר (או במהלך ההתקנה) לעבודה רגילה. המשך לקרוא כדי ללמוד איך לעשות את זה.

## 1.16 איך אני גורם ללינוקס לקרוס?

כ root , אתה יכול לעשות ככל העולה על רוחך, וזה כולל קריסה מואצת של המערכת. נסה (אם אתה באמת רוצה לגרום למערכת לקרוס) את:

```
cp /dev/zero /dev/mem
```

כ root אתה אפילו יכול למחוק את כל הקבצים במערכת שלך בפקודה אחת תמימה (אל תעשה את זה):

```
rm -fr /
```

זה לא אומר שקל להפיל את לינוקס, אלא שלמנהל המערכת (root) יש את השליטה המוחלטת על המערכת. אתה יכול לגרום למיקרוסופט ווינדוס להיות לא שמישה על ידי מחיקת קבצים מ c:\windows או מ c:\windows\system.

תחת לינוקס, חשבונות מנהל המערכת ושאר המשתמשים הינם נפרדים. משתמשים רגילים יכולים לגעת אך ורק בקבצים ששייכים להם, ובאופן דומה להריץ תוכניות רק ב"אזורים" שלהם, ולא משנה כמה מבוגגת או עויינת התוכנית תהיה. ההפרדה בין חשבונות מנהל מערכת (administrator) ובין חשבונות רגילים מוסיפה למורכבות המערכת, אבל היא עושה את לינוקס למערכת מרובת משתמשים (multi user) אמיתית. להבדיל מהגישה הישנה של מייקרוסופט. בגירסאות האחרונות של ווינדוס, מייקרוסופט נעה אל לעבר גישה יוניקסית.

אני נזכר באימרה ישנה שאומרת: "אלו שלא מכירים יוניקס מיועדים להמציא אותה מחדש".

מסקנה: אל תשתמש בחשבון root לעבודה שיגרתית. הוסף חשבון משתמש רגיל כמשימה הניהולית הראשונה שלך והשתמש בחשבון זה להתנסות שלך עם לינוקס.

הנה הדרך לבצע זאת (עם root):

```
adduser johnbrown
```

```
passwd johnbrown
```

[הזן סיסמא טובה וארוכה עם מספר באמצע]

[הקלד שוב את אותה הסיסמא בלי טעויות]

```
exit
```

[בצע לוג אין עם johnbrown]

החשבון root נועד לניהול והקמה בלבד. כ root לא הייתי עושה דברים/מריץ תוכניות שאני לא כל כך מבין איך הם עובדים ומה הם עושים. לפחות לא על מחשב שאני מייעד לעבודה אמיתית. באמת.

נתיב הלמידה שלי היה כזה:

1. התקנת לינוקס.

2. משחק עם המערכת, נסיונות (כ root ולא כ root , מקים דברים, מתנסה. אני רוצה להיות מסוגל לחזות את ההתנהגות של המערכת שלי--רק אז אדע שאני מבין אותה).

3. מתקין מחדש.

4. במידה ואין לי מספיק זמן (אף פעם אין לי) חוזר ל 2.

ברגע שלינוקס מותקנת כראוי על חומרה טובה, לינוקס של שורת הפקודה יציבה בצורה אגדית- היא יכולה לרוץ במשך חודשים ואפילו שנים. כמשתמש חדש, אתה יכול להיות בטוח שהתנהגות מוזרה של המערכת נובעת מהמעשים שלך כ root או מחומרה לא תקינה.

### 1.17 האם אני יכול להשתמש בממשק גרפי כל הזמן?

אני מאמין שזה אפשרי תחת לינוקס, אבל אני לא חושב שזה יהיה יעיל לניהול המערכת. הממשק הגרפי תחת לינוקס הוא נחמד מאוד, אבל הוא מחוויר בהשוואה למגוון האפשרויות הזמינות תחת שורת הפקודה. אני משתמש בממשק גרפי בכל יום תחת לינוקס וגם תחת ווינדוס, ואני מוצא אותו נהדר להפעלת תוכניות, ומשימות שיגרתיות אחרות. בכל זאת לניהול מערכת ואוטומציה, הממשק הגרפי אינו גמיש. ממשק שורת הפקודה הוא ממשק עשיר יותר מבחינתי, ובנוסף הוא גם נותן לי קצת הבנה לגבי העבודה של המערכת שלי.

שני ציטוטים כדי לפשט את הנקודה הזאת:

"השימוש בממשק גרפי הוא די קל כי הוא מגביל אותך לכמה פעולות בסיסיות שאתה יכול ללמוד במהירות. אבל, אם הפעולות האלו הן לא מה שאתה צריך, אז יכול להיות שתתקל בבעיות".

" מאפיין אחד של מערכת ידידותית למשתמש הוא שהיא עושה מה שהמשתמש רוצה. במילים אחרות, המערכת הידידותית ביותר אינה בהכרח הפשוטה ביותר".

אני מוצא ששילוב של ממשק גרפי ושורת פקודה הוא מושלם לכיסוי כל הצרכים שלי. אם אתה נחוש בדעתך להשתמש בממשק גרפי בלבד ואינך מוכן ללמוד אפשרויות של שורת פקודה, שקול דחייה של התקנת לינוקס עד שכלי הממשק הגרפי יהיו מפותחים יותר, אלא אם כן יש לך מישהו שיעזור לך לנהל את המחשב שלך בינתיים.

מצד שני, אם יש לך חבר או חברה שלא כל כך מבינים שמחשבים, וכל מה שהיא/הוא עושים זה להפעיל חצי תריסר תוכניות שונות, אתה יכול להקים בשבילם ממשק גרפי נחמד עם אייקונים או כפתורים ותפריטים שהיא/הוא דורשים.

מנקודת מבט של המדריך הזה, שורת הפקודה פשוטה הרבה יותר לתיעוד. אייקונים ותפריטים נועדו להיות מותאימים ולכן האייקונים והתפריטים שלך יכולים להיות שונים משלי. בנוסף, כדי להקים אייקון או פקודה, אני צריך לדעת את הפקודה שעומדת מאחורי האייקון. בקצרה, הבנת שורת הפקודה הכרחית לכל סוג של עבודה מעבר לעבודה טריוויאלית.

לא כללנו שום צילומי מסך במדריך הזה-אנחנו מוצאים אותם חסרי שימוש בלימוד על מחשבים.

## 1.18 איך לשדרג את הפצת הלינוקס שלי?

לשדרוג מערכת מלא ( שידרוג דיסק הפצה  $RH 7.0 < RH 7.2$  ) "לך בעקבות העיקרון המכובד של שידרוג מערכות יוניקס: התקן התקנה נקייה ורעננה והחזר בחזרה את הנתונים שלך. כן, אנחנו מדברים על פירמוט מחדש של המחיצות שלך והתקנה מאפס."

חלוקה למחיצות של הדיסק שלך בצורה נכונה יפשט שידרוגים עתידיים (קבצי הנתונים שלך יכולים להישמר בספריית ה `/home` ).

רשימה שהכנתי לעצמי לשידרוג:

- כניסה למערכת כ `root` .

- יצירת עותק של הסיפרייה `/etc` במקום אחר שבו היא לא תושמד. (כונן זיפ, תקליטור). זה ייתן לי אפשרות להסתכל בתצורה הקודמת שלי למקרה שאני נתקל בבעיות.

- יצירת העתק של כל המידע החשוב בספרייה `/home` . זה רק למקרה שמשהו באמת לא יילך כמו שצריך, למשל, אם אני אפרמט בטעות את כל המחיצה.

- קביעה ורישום של כל נקודות ההמרה של מערכת הקבצים על ידי שימוש בפקודה `df` , `(diskfree=)` . במערכת שלי לדוגמא, היא מראה שמחיצת הבית שלי מומרת על גבי מחיצת דיסק קשיח נפרדת שנקראת `/dev/hda9` . ושהספרייה `/usr/local` נמצאת ב `/dev/hsa8` .

- ביצוע התקנת לינוקס תוך שמירה ואי פירמוט של המחיצות `/home` ו `/usr/local` . זה קריטי אם אתה מעוניין לשמור מידע.

- איתחול של ההתקנה החדשה וביצוע בדיקה שהיא אכן עובדת.

- יצירת זהות לכל משתמש במערכת עם השם ומספר הזהות הישן שלו: לדוגמא:

```
ls /home/maria/* -l
```

(בדוק את זהות המשתמש לחשבון "maria" ברשימת הקבצים במחיצת הבית שלה)

```
useradd maria -u 503
```

(צור חשבון משתמש בשם "maria" עם uid 503 –לדוגמא)

מכל ספריית בית של כל משתמש, מחק קבצים עם פוטנציאל לבעיות כמו:

"kde", ".ICE" וכו'. זו גם הזדמנות טובה להיפטר ולמחוק זבל. הקבצים יכולים להיות בעייתיים כי סביר להניח שגירסה חדשה של kde תתפקד טוב יותר עם ההגדות המעודכנות שלה (אני אצור אותן עם האיתחול הראשון).

```
cd /home/maria
rm -fr .kde
```

וודא שכל קובץ בספריות הבית שייך למשתמש המתאים. למשל :

```
cd /home/maria  
chown -R maria *  
chgrp -R maria *
```

הבא: חלק 2, מקורות לקריאה, עזרה וכמה לינקים

## חלק 2. מקורות לקריאה, עזרה וכמה לינקים

### תוכן החלק הזה:

2.1 חומר קריאה לגבי לינוקס? [עמ' 36]

2.2 האם ישנה פקודת עזרה? [עמ' 37]

2.3 איזה שהוא מילון לגבי מונחים? [עמ' 38]

2.4 חיפוש ברשת [עמ' 38]

2.5 קבוצות דיון [עמ' 39]

2.6 לינקים ללינוקס באינטרנט? [עמ' 40]

2.7 קוד מקור--המקור האולטימטיבי [עמ' 43]



## 2.1 חומר קריאה לגבי לינוקס?

תקליטורי ההפצה של רדהאט מכילים המון תיעוד. חלק ממנו בפורמט html והשאר בפורמט טקסט פשוט. אתה יכול לקרוא את הכל תחת דוס או מייקרוסופט ווינדוס לפני שאתה מתקין את לינוקס.

לדוגמא, ניתן לצפות בהעתק של החוברת של רדהאט בכל דפדפן html תחת ווינדוס, נטסקייפ או אינטרנט אקספלורר. פשוט גש אל הקובץ:  
D:\doc\rhmanual\manual\index.html (בהנחה שכונן התקליטורים שלך הוא D תחת ווינדוס).

בנוסף, בדוק גם את הספרייה \doc\LDP בשביל חוברות מצויינות של פרוייקט התיעוד של לינוקס. לדוגמא, אתה יכול לצפות במדריך ניהול הרשת של לינוקס על ידי גישה אל הקובץ \doc\LDP\sag\sag.html עם דפדפן האינטרנט החביב עליך.

בנוסף, בדוק את הספרייה \doc\HOWTO בשביל מסמכי הHOWTO, הספרייה \doc\HOWTO\mini בשביל הMINIHOWTO ואת הספרייה \doc\FAQ בשביל סט של שאלות נפוצות בנושאים שונים ("FAQ=frequently asked questions"). לדוגמא, פקודות אלו ייתנו לך לקרוא את המסמך Linux-FAQ (מסמך טקסט פשוט) תחת דוס:

```
D:  
cd \doc\FAQ\txt  
edit Linux-FAQ
```

תחת לינוקס, אתה יכול לקרוא את אותו התיעוד מהתקליטור על ידי, לדוגמא, פקודה זו:

```
lynx /mnt/cdrom/doc/rhmanual/manual/index.html
```

זה יתחיל את lynx, דפדפן html פשוט שעובד במצב טקסט, לצפייה בחוברת של רדהאט. בבקשה שים לב, שתחת לינוקס, יש צורך להמיר (mount) תחילה את כונן התקליטור, ושהפקודה למעלה מניחה שנקודת ההמרה היא /mnt/cdrom. אתה יכול להשתמש גם בנטסקייפ ללינוקס, staroffice או בכל דפדפן html אחר כדי לצפות בחוברת של רדהאט ובתיעוד אחר בפורמט html. אתה יכול לקרוא מסמכי טקסט פשוטים מהתקליטור על ידי שימוש, לדוגמא, בפקודות אלו:

```
cd /mnt/cdrom/doc/FAQ/txt  
less Linux-FAQ
```

(הפקודה less מאפשרת לך לגלול את התוכן של קובץ טקסט.)

לאחר התקנת לינוקס, התיעוד, לא משנה איזה חלק התקנת, נמצא בספרייה /usr/doc או /usr/share/doc. אם לא התקנת את התיעוד, חשוב על התקנת כולו כעת, ייתכן וזה יהיה שווה את זה. לדוגמא, הספרייה, /usr/doc/LDP מכילה את חוברות פרוייקט התיעוד של לינוקס. פקודות אלו יאפשרו לך לצפות במדריך הניהול של לינוקס:

```
cd /usr/doc/LDP/sag  
lynx sag.html
```

בנוסף בדוק את `/usr/doc/HOWTO` בשביל מסמכי ה-HOWTO, ואת `/usr/doc/HOWTO/mini` בשביל ה-MINIHOWTO. המיקום של התיעוד נמצא לפעמים ב-`/usr/share/doc`.

לתיעוד נוסף ועדכני, הסתכל ב: <http://www.ibiblio.org/mdw/index.html>

## 2.2 האם ישנה פקודת עזרה?

ניתן להריץ את רוב הפקודות בלינוקס עם האופציה "`--help`". לדוגמא, פקודה זו תתן לך עזרה לגבי הפקודה `cp` (copy):

```
cp -- help | less
```

מידע מקיף יותר נגיש משורת הפקודה על ידי שימוש במה שנקרא חוברת השימוש `man topic` (manual pages). לדוגמא:

```
man cp
```

תציג את עמוד ההסבר שבחוברת לגבי הפקודה `cp`. דפי ההסברים (manual pages) הם מערכת העזרה הסטנדרטית תחת לינוקס, ומכילה עושר רב של פרטים, מידע טכני מאוד, אבל בדך כלל דורשים קצת מאמץ כדי להיות מובנים אם הקורא הוא משתמש חדש.

הפקודה `man` משתמשת בכלי קטן בשם `less` שייאפשר לך לגלול את הטקסט. השתמש במקשי החיצים כדי לגלול, לחץ על "q" כדי לפרוש. למעשה, `less` יכול לעשות יותר מזה. לחץ על "h" לעזרה כשרתה מריץ את `less`, או למד יותר על `less` על ידי שימוש בפקודה

```
man less
```

ישנה גם פקודת המידע `info topic`. לדוגמא:

```
info cp
```

שתתן לך עזרה לפקודה "`cp`" (copy). לעיתים קרובות `info` מכילה מידע דומה ל-`man`, אבל עדכני יותר. לרוע המזל, כלי הניווט של `info` אינו כה אינטואיטיבי, אישית אני משתמש ב-`man` לעיתים תכופות יותר. ישנו גם `pinfo` (תחליף לממשק של `info`, אולי קל יותר לשימוש מאשר `info`).

אם אתה לא זוכר בדיוק את שם הפקודה שבה אתה צריך להשתמש, נסה את `apropos`. לדוגמא, כדי להניב רשימה של פקודות שיש להן איזה שהוא קשר אל "copy", אני מבצע את זה משורת הפקודה:

```
apropos copy
```

הפקודה `what` דומה ל-`apropos`, אבל מתאימה רק מילות מפתח, בעוד ש-`apropos` מחפש במסד הנתונים השלם (מילות מפתח והתיאור שלהן). כתוצאה מכך, `what` נוטה להניב פלט קצר יותר (אולי רלוונטי יותר).

בכמה תוכניות שמונעות על ידי תפריט, לדוגמא כשמקנפגים את שירותי המערכת שלך על ידי ntsysv (או setup או linuxconf), אתה יכול ללחוץ על f1 בשביל מידע לגבי מה עושה שירות מסויים.

הרשימה של פקודות bash מובנות ניתנת להצגה על ידי הקלדת help בשורת הפקודה. ולאחר מכן עזרה לכל פקודת bash מובנת ניתן להצגה על ידי, לדוגמא:

help cd

bash היא המעטפת (shell) הסטנדרטית של שורת הפקודה, המקבילה של לינוקס למעבד שורת הפקודה של דוס "COMMAND.COM".

סביבת KDE כוללת דפדפן עזרה ("help browser") מבוסס ממשק גרפי, שניתן להפעלה על ידי האייקון המתאים ב"Kpanel" (תפריט המערכת, בדרך כלל בתחתית המסך). דפדפן זה יכול לשמש לגישה אל עזרה ספציפית לגבי KDE כמו גם אל דפי המידע של המערכת (manual pages). שולחן העבודה Gnome מכיל מערכת עזרה דומה.

אם אתה רוצה ללמוד על החבילות הרבות שמגיעות על התקליטור בפורמט rpm, יכול להיות שתצצה להשתמש בkpackagege מבוסס הממשק הגרפי (הקלד kpackagege בטרמינל X) כדי לדפדף בחבילות, להציג את המידע שהן מכילות, ולהתקין אותן אם אתה רוצה (ההתקנה חייבת להתבצע על ידי root). במקום kpackage, הפצות ישנות יותר משתמשות ב glint (רדהאט 5.2) או בgnomrpm (רדהאט 6.0), שתי האחרונות איטיות יותר ופחות נוחות.

### 2.3 איזה שהוא מילון לגבי מונחים?

מילון זה די מקסימליסטי: "the new hackers dictionary" ידוע גם בשם "jargon file":  
<http://www.tuxedo.org/~esr/jargon/jargon.html>  
זהו לא רק מקור מצוין, אלא גם מספק קריא מבדרת מאוד. מומלץ.

כדי להוסיף בידור לבידור, הנה לינק נוסף שאני אוהב: "a girls guide to geek guys"  
<http://college.antioch.edu/~totally/geek.html>  
אם אתה זז את, יכול להיות שתעדיפי את: "a guy's guide to geek girls"  
[www.eecis.udel.edu/~masterma/guidetogeekgirls.html](http://www.eecis.udel.edu/~masterma/guidetogeekgirls.html)

רשימה די ארוכה של קיצורים שקשורים למחשבים וראשי תיבות ניתן למצוא ב:  
[http://www.geocities.com/ikind\\_babel/babel/babel.html](http://www.geocities.com/ikind_babel/babel/babel.html)

### 2.4 חיפוש ברשת

כרגע, מנוע החיפוש הטוב ביותר ברשת הוא Google, מדהים מה אתה יכול למצוא בעזרתו. גוגל ממש מהיר, מכיון שהוא רץ על לינוקס, אני לא צוחק. נסה את <http://www.google.com>. כדי לבחון אותו, בצע גלישת אגו (הקלד בתיבת החיפוש את: שם המשפחה שלך ומילה לבחירתך). ניתן להשתמש בגוגל כדי למצוא כמעט כל דבר רלוונטי לגבי לינוקס (או לגבי כל דבר אחר) ברשת. פשוט הקלד כמה מילות מפתח כדי למצוא את דוקומנטציית הלינוקס שאתה צריך.

## 2.5 קבוצות דיון

זה יכול להיות מקום קצת מפחיד לפעמים--נדמה שכל האנשים הכי מוזרים בעולם תמיד נוכחים בקבוצות הדיון. אני בדרך כלל בוחר להתעלם מהתגובות המטופשות או המעליבות או מהמייילים. לגבי אלו בעלי כוונות הזדון, אני מודיע למנהל הרשת של ספק הדואר שלהם. פרסומות שאני מקבל לאחר פרסום הודעה בקבוצת דיון נמחקות לפני שאני קורא אותן-- אני יודע שאני לא היחיד שנהג כך.

למרות החסרונות קבוצות דיון יכולות להיות דרך יעילה במיוחד למציאת המידע שאותו אתה צריך.

לפני שאתה פונה אל קבוצות דיון, הייתי ממליץ בחום לפנות אל ארכיבי הניוז של גוגל (<http://groups.google.com>), המוכרים כdejanews. זהו מאגר ענק של קבוצות דיון ואתה יכול לחפש בתוכו בעזרת כלי חיפוש טובים. בדרך זו, אתה יכול לעיתים קרובות למצוא תשובה לשאלה שלך ללא הצורך לעבור דרך טונות של זבל, ובלי לחשוף את עצמך לכעס לאחר הצבת שאלה "שכבר נשאלה 10 פעמים השבוע". יכול להיות שתופתע מכמות המידע שזמין דרך הארכיב של גוגל.

ישנן כמה קבוצות דיון שמוקדשות ללינוקס ונראה שהן טובות יותר מקבוצות דיון אחרות (מנוהלות יותר טוב על ידי מומחי לינוקס?). לפניך רשימה קצרה:

news:comp.os.linux.announce (moderated)

(הפניות עוברות סינון על ידי moderator, שבודק אותן לפני הפרסום)

news:comp.os.linux.hardware  
news:comp.os.linux.security  
news:comp.os.linux.misc (miscellaneous)  
news:comp.os.linux.advocacy

(השתמש בזו כדי לדון ביתרונות ובחסרונות של לינוקס ואולי גם להשוות בין לינוקס לבין מערכות הפעלה אחרות. זוהי קבוצת דיון מצויינת למי שאוהב להיכנס לוויכוחים שלא נגמרים)

news:Alt.linux.sux

(כאן אתה יכול לכתוב/לקרוא את הדעות האמיתיות על לינוקס)

news:comp.os.linux.x (X-windows)  
news:comp.os.unix

(קבוצת דיון ככלית לגבי יוניקס)

בבקשה שים לב שישנה אתיקה ("נטיקה") של קבוצות דיון, ואתה מסתכן בדחייה ואולי גם חושף את עצמך ללהבות אם אתה בוחר להפר אותה. הנקודות העיקריות:  
- אל תשלח נושא שאינו קשור לנושא של קבוצת הדיון.  
- אל תשלח את אותה ההודעה לכמה קבוצות דיון באותו הזמן (cross-post).  
- השתמש בascii נקי, אל תשלח אטצ'מנטס, תמונות, html וכו'.

- אל תפרסם (במיוחד לא מוצרים מסחריים).  
- אל תכתוב באותיות גדולות בלבד. (זה חצוף!)  
- השתמש בנושא פשוט לתיאור הנושא(סתם לכתוב "עזרה" לא יעזור) והסבר בקצרה את הבעיה שלך. כולל תיאור של מספר גירסה, וזהה את החומרה שלך (אל תכתוב מודם 56k - usf - רשום את מספר הדגם).

כדי לקרוא קבוצות דיון (נקראות גם usenet), עליך לקנפג את הגישה שלך אל שרת קבוצות דיון -newsserver. הדרך הפשוטה ביותר תהיה לקנפג את הדפדפן שלך על ידי ציון שרת הניוז שלך (ספק האינטרנט שלך, isp, צריך לספק לך את שם השרת) ואז להוסיף את קבוצת הדיון המתאימה אל תיבות הדואר המקומיות שלך, אם אינך יודע את שם שרת הניוז, נסה את: news.my\_isp\_provider\_name.and\_domain או אולי רק את "my\_isp\_provider\_name.and\_domain".

לקריאת ניוז, אני מעדיף את knod על פני נטסקייפ. התקנה ולמידה של קבוצות דיון היה בהחלט משהו ששוה את המאמץ.

## 2.6 לינקים ללינוקס באינטרנט?

יש בוודאי אלפי אתרים שמוקדשים ללינוקס. לפניך כמה לינקים ללינוקס שאני אוהב, לא מסודרים בסדר מסוים. אם אתה זקוק למשהו אחר, בטח תמצא הפניה מתאימה באחד מן האתרים האלו.

-<http://sunsite.dk/linux-newbie/>

האתר הראשי של מסמך זה (lnag) הוסף למועדפים.

-<http://www.linuxdoc.org/docs.html>

פרוייקט התייעוד של לינוקס- הבית של הרבה FAQs, howtos, minihowtos ומדריכים. תמיד מעודכן.

-<http://www.kalug.lug.net/linux-admin-FAQ>

FAQ לאדמניסטרטור לינוקס (שאינו ניוביי)

-<http://members.aa.net/~swear/pedia/learning-linux.html>

האנציקלופדיה של גרי- ללמוד לינוקס. הוסף למועדפים.

-<http://jgo.local.net/linuxguide/>

עמוד הבית של גוש. מקור טוב ללימוד לינוקס.

-<http://www.control-escape.com/>

אתר טוב לניובייז.

-<http://www.linuxninja.com/linux-admin/>

ניהול לינוקס בקלות. מומלץ.

-<http://metalab/unc/edu/mdw/index.html#guide>

המון תיעוד על לינוקס. הוסף למועדפים.

-<http://www.frankelinux.com>

עוד אתר שעוזר לנייבייז.

-<http://www.easyfeed.com/~jgo/linuxguide>

מדריך נוסף לנייבייז.

-<http://www.slashdot.org/>

דיונים לחנונים, האקרים, גורואים וכו' (!.=)

-<http://www.freashmeat.org/>

עידכון לגבי תוכנות לינוקס שיוצאות עכשיו.

-<http://linuxtoday.com/>

חדשות לינוקס- טוב לקריאה יומיומית. הוסף למועדפים.

-<http://www.llp.fu-berlin.de>

"פרוייקט המעבדה של לינוקס". למי שמתעסק במדעים.

-<http://linuxberg.com/>

פורטל גדול. יש להם הכל שם. אני אוהב את הדירוג שלהם לתוכנה של לינוקס, אני מתקין רק תוכנה שקיבלה שם 5 פינגווינים. מועדפים.

-[http://counter.li.org/linuxcounter\\_eng.html](http://counter.li.org/linuxcounter_eng.html)

המונה של לינוקס. רשום את עצמך כלינוקסאי.

- <http://counter.li.org>

ראה את המפות המרשימות של דניס הבליקס על הגדילה של לינוקס ועל התפוצה הגאוגרפית.

-<http://www.cl.cam.ac.uk/users/iwj10/linux-faq/index.html>

לינוקס faq

-<ftp://sunsite.unc.edu/pub/linux>

טונות של תוכנות ללינוקס בארכיב של sunsite. העדף את האתר.

-<http://stommel.tamu.edu/~baum/linuxlist/linuslist/linuslist.html>

אפליקציות ללינוקס

-<http://www.boutell.com/lsm/>

אפליקציות ללינוקס

-<http://www.linuxlinks.com/software/>

אפליקציות ללינוקס

-[http://directorysearch.mozilla.org/computers/operaring\\_systems/linux](http://directorysearch.mozilla.org/computers/operaring_systems/linux)

פורטל גדול (טוב יותר משל yahoo) עם לינקים מעולים לניובייז.

-<http://dir.yahoo.com/.../unix/linux>

הרשומות של yahoo לגבי לינוקס.

-<http://www.debian.org/>

האתר של דביאן לינוקס.

-<http://hardware.redhat.com/hcl/genpage2.cgi>

רשימת תאימות החומרה ללינוקס.

-<http://metalab.unc.edu.mdw/links.html>

הרבה לינקים מועילים שקשורים ללינוקס.

-<http://www.cse.unsw.edu.au/~conradp/linux/>

לינקים מעולים.

-<http://www.linuxlinks.com/>

ועוד לינקים.

-<http://www.gnu.org/>

האתר הראשי של gnu.

-<http://www.redhat.com/>

האתר של רד האט.

-<http://www.cs.helsinki.fi/u/torvalds/>

עמוד הבית של לינוס טרבורדה.

## 2.7 קוד מקור--המקור האולטימטיבי

מקור העיון האולטימטיבי תחת לינוקס הוא קוד המקור. אם התקנת אותו (מגיע עם הפצות סטנדרטיות, אנחנו באמת ממליצים להתקין אותו אם יש לך מקום על הדיסק), הוא נמצא ב-`/usr/src/linux/` (the kernel source) וב-`/usr/src/RPM/sources/` (קוד המקור לאיזון חבילות המקר).  
כיצד יכול קוד המקור לשמש משתמש חדש? ובכן, הוא מכיל את כל ההערות והתיעוד עד לפרט הקטן ביותר. לדוגמא, בהמשך המדריך אנחנו מראים איך לקרוא/להגדיר חלק מהפרמטרים של סביבת הריצה של הקרנל דרך מערכת הקבצים `/proc`. אתה יכול לקרוא את התיעוד המלא לכל הפרמטרים האפשריים על ידי:

```
less /usr/src/linux/documentaion/proc.txt
```

כדי להתקין את המקור של הקרנל, הייתי בוחר את חבילות המקר המתאימות בזמן ההתקנה הראשית שלי. כדי להתקין מקור לחבילות אחרות שמגיעות עם ההפצה שלי, הייתי מכניס את ה"source CD" לתוך כונן התקליטורים ועושה משהו כמו (כנס עם `cd` של רדהאט):

[מתקין את קוד המקור של גליון הנתונים gnumeric מהתליטור אל הדיסק הקשיח]

```
su
cd /mnt/cdrom/SRPMS/
rpm -ivh gnume<tab>
```

[עושה unzip לקוד המקור שזה עתה התקנתי]

```
cd /usr/src/RPM/SOUR<tab>
tar -xvzf gnumer<tab>
```

[קורא את הקוד לפונקציות סטטיסטיות בgnumeric]

```
cd gnumeric/src/functions
less fn-stat.c
```

זהו באמת המקור האמיתי לגבי איך גליון נתונים מסויים עובד, בלי צחוק.

עבור לחלק 3: פעולות בסיסיות



### 3.1 יסודות

3.1.1 שמות קבצים. [עמ' 46]

3.1.2 למה משמשות המחיצות השונות? [עמ' 49]

3.1.3 איך אני מריץ תוכנית? [עמ' 52]

3.1.4 כיצד אני יכול לשנות את הntיב? [עמ' 54]

3.1.5 איך אני יכול לכבות את המחשב? [עמ' 55]

3.1.6 איך אני מתמודד עם תוכנה תקוע? [עמ' 57]

3.1.7 אפשרויות פקודה. [עמ' 61]

### 3.2 משתמשים, סיסמאות, הרשאות קבצים, ואבטחה

3.2.1 ספריות בית, root הוספת משתמש [עמ' 62]

3.2.2 קצת על אבטחת סיסמא [עמ' 63]

3.2.3 שכחתי את הסיסמא של root [עמ' 65]

3.2.4 שכחתי את סיסמת המשתמש שלי [עמ' 68]

3.2.5 ביטול או הסרה של חשבון משתמש [עמ' 68]

3.2.6 יש לי בעייה עם הרשאות קבצים, הרשאות קבצים ובעלות על קבצים. איך זה עובד?  
[עמ' 69]

3.2.7 נגן mp3 שלי נחנק. הצליל לא נשמע טוב (איך להגדיר suid) [עמ' 73]

### 3.3 תזמון משימות עם "&", "at", "batch", ו-"cron"

3.3.1 איך אני מבצע פקודה ברקע? [עמ' 75]

3.3.2 איך אני מבצע פקודה בזמן מסויים (על ידי שימוש ב"at" או "batch")? [עמ' 75]

3.3.3 איך אני מגדיר את cron? [עמ' 77]

## **shell 3.4**

- 3.4.1 מה זה shell (מעטפת) והאם אני רוצה להשתמש באחרת? [עמ' 79]
- 3.4.2 כיצד אני מבצע התאמה אישית למעטפת שלי? [עמ' 80]
- 3.4.3 צבע בטרמינל טקסט [עמ' 81]
- 3.4.4 כיצד אני מדפיס סימנים בקונסול, באפליקציות מצב טקסט, וב-X? [עמ' 82]
- 3.4.5 איך אני כותב סקריפט מעטפת פשוט? [עמ' 83]
- 3.4.6 המשמעות של ציטוטים [עמ' 85]
- 3.4.7 הפניה מחודשת של פלט/קלט [עמ' 85]
- 3.4.8, תוי מעטפת מיוחדים (metachacters) [עמ' 88]

## **3.5 התקנת חבילה ומנהל החבילות rpm?**

- 3.5.1 איך אני מתקין תוכנית שהורדתי מאינטרנט? [עמ' 90]

### 3.1 יסודות

#### 3.1.1 שמות קבצים

לינוקס היא case sensitive . למשל, myfile, MyFILE, & myFILE הם שלושה קבצים שונים.

גם הסימא שלך והשם שלך, איתנו אתה נכנס למערכת, הינם case sensitive . (זה מסורתי, גם יוניקס וגם שפת התיכנות C הם case sensitive) . מוסכמת שמות לקבצים ולספריות הינן זהות. כל הקבצים והספריות שאני יוצר (לעצמי כמשתמש) הם באותיות קטנות, אלא אם כן ישנה סיבה מיוחדת לשנות את זה. רוב הפקודות של לינוקס גם הן נכתבות באותיות קטנות.

שמות קבצים תחת לינוקס יכולים להכיל עד 256 תווים ובדרך כלל מכילים אותיות, מספרים, נקודות ".", קווים תחתונים "\_", ומקפים "-". לא מומלץ להשתמש בתווים מיוחדים: כוכבית "\*", סימן שאלה "?", רווח " ", סימן דולר "\$", ו"&", וכו'.

זה נובע מהעובדה שלתווים מיוחדים אלו יש משמעות מיוחדת לshell של לינוקס. (shell - מעטפת- זה משהו כמו command.com תחת ווינדוס). שם קובץ יכול להכיל רווח, אבל לא היינו ממליצים על זה-- אנחנו משתמשים בקו תחתון "\_" במקום.

לא ניתן להשתמש בקו נטוי "/" כחלק משם קובץ מכיוון שהוא מייצג את ראש עץ הסיפרייה, ומשמש כמפריד בנתיבי שמות (מקביל ל "\" בדוס).

כמו בדוס, אי אפשר לקרוא לקובץ ". " או ".." (נקודה או שתי נקודות)--כי הן מסמלות את הסיפרייה הנוכחית ואת הסיפרייה האם בהתאמה. ממש כמו בדוס.

לפניך המשמעויות של כמה תווים מיוחדים (או metacharacters):

\*= מתאים לכל רצף של אפס תווים או יותר, פרט ל "." (נקודה) בהתחלת קובץ.

?=מתאים לכל תו יחיד.

[abC1]=מתאים לתו יחיד בסט . בדוגמא זו הסט מכיל: ; a, b, C ו-1.

[a-z]=מתאים לכל אות קטנה.

[A-F]= מתאים לכל אות גדולה מ A ל F .

[0-9]- מתאים לכל סיפרה אחת.

[a-zA-Z0-9]=מתאים לכל אות (קטנה או גדולה) ולכל סיפרה.

גם התו "\" back slash מיוחד. הוא גורם לתו מיוחד להיות בעל משמעות (המשך לקרוא).

דוגמאות. הפקודה הזו תציג כל קובץ בסיפרייה הנוכחית, פרט לקבצים שמתחילים ב". (נקודה):

ls \*

מקבילה לפקודה זו תהיה הקלדה של ls בלבד או dir בלי ה \*. קבצים עם שמות שמתחילים בנקודה ". לא יוצגו מכיוון האות הראשונה בשם קובץ לא תואמת ל"\*. ". תחשוב על קבצים שמתחילים בנקודה כמקבילה של קבצים נסתרים בדוס. השתמש ב ls-a (מציג רשימה עם האופציה "all" ) או ls.\* כדי להציג קבצי נקודה. קבצי נקודה שכיחים ב ספריות user home ובדרך כלל משמשים כדי להכיל קונפיגורציות ברמת משתמש.

פקודה זו תציג כל קובץ (בסיפרייה הנוכחית) שמכיל נקודה:

ls \*.\*

פקודה זו תציג כל קובץ שמכיל שתי נקודות:

ls \*.\*.\*

שים לב בבקשה שללינוקס אין "הארכות שמות קבצים" כמו שיש לדוס, אבל עדיין תוכל להשתמש בהן. למשל, יכול להיות שיהיה בבעלותי קובץ בשם my\_text.txt.zip. שמות קבצים אחרים (כמו "micro~1.doc" למשל) של דוס נעדרים כאן.

פקודה זו תאתר (בכל מערכת הקבצים) כל קובץ עם התוספת "htm" כשאחרי תו כלשהו:

locate \*.htm?

פקודה זו תראה את כל הקבצים בסיפרייה הנוכחית, שמתחילים ב "a" או "b", או בכל אות גדולה:

ls [abA-Z]

פקודה זו תציג כל קובץ שמתחיל ב "a" ונגמר ב "n" :

ls a\*n

**השלמה אוטומטית בשורת הפקודה.** (autocomplation) זוהי אופציה נהדרת של שורת הפקודה-- אני משתמש במקש [tab] המון כדי לחסוך הקלדה מיותרת. זה הופך את כל ההתעסקות עם שמות קבצים ארוכים לקלה ומהירה. לדוגמא, שימוש בשם קובץ כזה בשורת הפקודה ממש לא מהווה בעייה, במידה ואתה משתמש בהשלמה אוטומטית:

dir Eurosong\ 2000\ olson\ brothers\ -\ fly\ on\ the\ wings\ of\ love\ \ (denmark\)/mp3

אני פשוט מקליד:

```
dir Eu<tab>
```

ואם אין קבצים אחרים שמתחילים ב "Eu", שאר הקובץ מוקלד אוטומטית בשבילי. אחרת, אני אצטרך להסתכל באפשרויות (שמודפסות בשבילי) ולהקליד עוד תו אחד או שניים כדי להבדיל בין שם הקובץ המבוקש לבין השאר. ה backslash "\" בדוגמא למעלה הוא חלק משם הקובץ.

**צרות עם שמות קבצים מוזרים.** רוב הבעיות האלו יכולות להיפתר על ידי שימוש בהשלמה אוטומטית. בנוסף, כדי ל"נווט" עם שמות שמכילים סימנים מיוחדים, אני משתמש בזוג גרשיים ' ', כך שהסימנים המיוחדים מצוטטים והמעטפת (shell) לא מפרשת אותם כמשמעותם. כדוגמא, כדי לשנות את שם הקובץ "my file\*" אני מקליד:

```
mv 'my file*' filename_without_weird+characters.txt"
```

שים לב שהשתימתי בזוג ' מרכאות לציטוט. ציטוט עם זוג " (גרשיים) הינו באופן כללי חלש יותר מאשר ציטוט עם ' . אם תשתמש בגרשיים " , סימנים מיוחדים אחדים יכולים להתפרש על ידי המעטפת.

בהתאם למסורת של יוניקס, בלינוקס, אתה יכול ליצור קבצים בעלי שמות שמכילים כמעט כל תו או סימן, כולל תוים לא מודפסים (control). תוים אלו אינם נפוצים, אבל אם אתה נתקל בקובץ כזה, הוא יכול לגרום לך להרגיש ממש משונה. הייתי משנה שם קובץ כזה על ידי שימוש סימן מיוחד שמוקם בזירות בשם הקובץ. קודם כל הייתי משתמש ב ls כדי לראות אם הפעולה שלי אכן הצליחה ואחר כך משנה את שם הקובץ:

```
ls -l myfile*y.html
```

```
mv myfil*y.html myfile.html
```

כדוגמא לבעייה אולי הכי מוזרה שאתה עלול לפגוש בשימוש בתוים לא ממוצאים בתוך שם קובץ, נסה ליצור קובץ עם שם שמתחיל במקף ואז הסר אותו--נדמה שאין דרך לעשות את זה (מכיוון שמקף מסמן אופציות לפקודה)

הפקודה :

```
dir > -junk
```

תיצור קובץ מצחיק (כמו בדוס, הסימן > מפנה את הפלט מהפקודה dir אל קובץ בשם -junk). מכיוון שהדרך הרגילה להסרת הקובץ -junk לא עובדת, אני משתמש ב:

```
rm ./-junk
```

ההתחלה עם נקודה קו נטוי (dot slash) מתפרשת כ "בספריה הנוכחית" ומשרתת כאן את הרצון להסתיר את המקף כדי שלא ייתפרש כאופצייה לפקודה וכו'. הנקודה כאן היא שעדיף שאדבוק במוסכמות שמות מסורתיות מאשר אתקל בסיבוכים מיותרים.

מלבד שימוש בהשלמה אוטומטית, מרכאות וגרשיים, אני יכול לעשות מניפולציה על קבצים על ידי שימוש ב קו נטוי אחורי (backslash).  
לדוגמא, אני יכול ליצור קובץ בעל השם המשונה \*? [ על ידי שימוש בפקודה :

```
touch \*\?*
```

(הפקודה touch יוצרת קובץ ריק או, במידה והקובץ קיים, מעדכנת את התאריך/זמן של העידכון האחרון).

### 3.1.2 למה משמשות הספריות השונות?

בקצרה, לינוקס מכילה חמש מערכות קבצים. מערכות קבצים אלו שוכנות על גבי דיסק פיזי אחד או יותר (ואחר) או על גבי מחיצה אחת או יותר (ואחרת), תלוי בגודל ובצורך של המערכת שלך. (מערכת קבצים אחת יכולה להיות מופצת בין כמה התקנים פיזיים, במקרה הצורך).

#### the root "/" filesystem

מכילה את מערכת ההפעלה הבסיסית וכלי תחזוקה. התוכן של מערכת קבצים זו אמור להספיק להפעלה של המערכת ולביצוע תחזוקת חירום ותיקון אם יש צורך בכך.

#### /usr filesystem

מכילה את כל הפקודות, ספריות, תיעוד, וקבצים אחרים שלא משתנים בזמן תיפקוד נורמלי. סיפרייה זו מכילה גם אפליקציות רבות שמגיעות עם ההפצה שלך, netscape למשל.

#### /var filesystem

מכילה קבצים שמשתנים: ספריות spool, קבצי יומן (log), קבצי lock, קבצים זמניים, ודפי תיפעול (manual pages).

#### /proc filesystem

מכילה קבצים הזויים ודמיוניים בהחלט. הם לא קיימים באמת על גבי הדיסק והם אינם תופסים מקום (למרות ש ls -l תראה את הגודל שלהם) כשאתה צופה בהם, אתה למעשה ניגש למידע שמאוחסן בזיכרון. משמש לגישה למידע לגבי המערכת.

החלקים של מערכת הקבצים root הם:

/bin – executables (binaries) שבשימוש על ידי המערכת בזמן האיתחול, שיכולים להיות בשימוש על ידי משתמשים רגילים.

—/sbin executables (system binaries) לא מכוונים לשימוש על ידי משתמשים רגילים (משתמשים עדיין יכולים להשתמש בהם, אבל סיפרייה זו אינה הנתיב שלהם PATH).

—/etc קבצי קינפוג של כלל מערכת ההפעלה שלך.

`--/root` ספריית הבית של מנהל המערכת. (נקרא משתמש על – `super user` או `root`)

`--/dev` קבצי התקנים. התקנים מופיעים בלינוקס כקבצים כך שחומרה הופכת אבסטרקטית וקל לכתוב אליהם ולקרוא מהם.

`--/mnt` נקודות המרה למדיה ניידת (פלופי, `cdrom`, כונני זיפ), מחיצות למערכות הפעלה אחרות (ווינדוס), שיתופי רשת, וכל דבר אחר שמומר אל מערכת הקבצים באופן זמני. באופן נורמלי מכילה תת ספרייה נפרדת לכל נקודת המרה. תכולת כוננים/שיתופים אלו מופיע בתת ספריות אלו—אין להם אותיות כונן (C או D) בלינוקס.

`--/lib` ספריות משותפות לתוכניות ששוכנות על גבי מערכת הקבצים `root` ועל מודולי קרנל.

`--/boot` קבצים שבשימוש טוען איתחול (`LILLO` או `GRUB`), מה שנטען ראשון כשהמחשב מאתחל ואולי גם מה שנותן לך את האפשרות לבחור מערכת הפעלה לאתחול, אם יש לך יותר ממערכת הפעלה אחת על המחשב. בדרך כלל מכיל את הקרנל של לינוקס (בקובץ הדרוס, `vmlinuz`), אבל יכול להיות מאוחסן גם במקום אחר, במידה ו `lilo` מקונפגת לדעת היכן.

`--/opt` אפליקציות אופציונליות גדולות, כמו `kde` תחת רדהאט 5.2 (תחת רדהאט 6 `kde` מופץ ככל תוכנית `x-windos`, `executables` עיקריים נמצאים ב `/usr/bin`).

`--/tmp` קבצים זמניים. ייתכן וספרייה זו תתנקה אוטומטית.

`--/lost+found` קבצים ששוקמו בזמן תיקון של מערכת הקבצים.

החלקים הכי מעניינים של `/usr filesystem` הם:

`--/usr/x11r6` מערכת ה `x-windos` (גירסה 11, הפצה 6).

`/usr/x11` - זהה ל `/usrx11r6` (לינק סימבולי).

`--/usr/x11r6/bin` הרבה אפליקציות של `x-windows` ששוכנות בתת הספריות שלהן במקום אחר.

`--/usr/doc` תיעוד של לינוקס (במערכות חדשות, זה `zz` ל `/usr/share/doc`).

`--/usr/share` מידע עצמאי מארכיטקטורת המחשב שלך.

`/usr/bin` ו-`/usr/sbin` דומה למקבילות שלהן במערכת הקבצים `root` (`/bin` ו `/sbin`), אבל אינן דרושות לאיתחול בסיסי (בזמן תחזוקת חירום) רוב הפקודות יישכנו שם.

`--/usr/local` האפליקציות שמותקנות על ידי האדמיניסטרטור המקומי (ייתכן שכל אפליקציה בתת ספרייה נפרדת). לאחר ההתקנה ה"ראשית" ספרייה זו ריקה. התוכן שלה אמור לשרוד התקנה מחדש נורמלית או שידרוג של מערכת הפעלה.

`/usr/local/bin`—אולי `executables` קטנים יותר של המשתמש, פלוס לינקים סימבוליים `executables` גדולים יותר שנמצאים בתת ספריות נפרדות תחת `/usr/loacl`.

חשוב להבין שכל הספריות מופיעות בעץ ספרייה יחיד, אפילו אם הספריות נמצאות על גבי מחיצות שונות, דיסק פיזי אחר, או אפילו אם הן נמצאות ברשת. לכן אין אותיות כונן כמו בדוס תחת לינוקס. מה שיהיה כונן תחת דוס יופיע כתת סיפרייה במיקום המרה מיוחד תחת לינוקס.

מערכת הספריות (מדורים) מיושמת היטב תחת רוב ההפצות של לינוקס. היא גם די דומה למערכת שנמצאת תחת מערכות יוניקס מסחריות רבות.

### עוד קצת על מערכת הקבצים `/proc` (רק לסקרנים באמת).

מערכת הקבצים הפסוודונית `/proc` הינה מערכת קבצים ששוכנת בזמן אמיתי בזיכרון, ושעוקבת אחר המצב של ליבת מערכת ההפעלה (הקרנל-kernal) ואחרי התהליכים שרצים על המחשב שלך. מערכת הקבצים `/proc` היא ווירטואלית לחלוטין, היא אינה נכתבת על גבי דיסק כל שהוא או על מדיה אחרת, היא קיימת אך ורק בזיכרון המחשב, ומעודכנת באופן תמידי על מנת לשקף שינויים במערכת שלך. הגודל של מערכת הקבצים `/proc` הוא תמיד אפס וזמן העידכון האחרון הוא התאריך הנוכחי. במקרים מסויימים, ניתן לשנות את הגדרות המערכת על ידי שינוי ידני של תוכן הקבצים אשר בתוך `/proc`. כלים רבים של לינוקס משתמשים במערכת הקבצים `/proc` כמקור המידע שלהם, `dmesg`, `....ps`, `top`

התוכן של `/proc`.

ספריות עם שמות מספריים כמו "1" "170" "4980" הינם תגי זיהוי של תהליכים (processes) שרצים במחשב שלך. כל ספרייה מכילה כמה קבצים; `cmdline` (מכיל את כל שורת הפקודה ששימשה לעורר את התהליך), `cwd` (לינק סימבולי ל `cwd` של התהליך), `environ` (משתני הסביבה שמוגדרים עליד תהליך מסויים באופן `VARIABLE=value`), `exe` (לינק סימבולי לקובץ `executable` שהתהליך הנוכחי מקושר אליו), `fd` (רשימה של מילות מפתח שנפתחו על ידי התהליך), `mpas` (משרשר `-- pipe` שיכול לשמש לגישה אל זיכרון התהליך), `pipe=`מורה למערכת ההפעלה להפנות פלט של תוכנה `x` לקלט של תוכנה `y`—א.א]

`root` (לינק סימבולי אל הספרייה שהיא מערכת הקבצים `root` לגבי תהליך מסויים), `stat` (מידע לגבי מצב של התהליך).

קבצים נוספים ב `/proc`:

`/proc/cpuinfo`—מידע לגבי המעבד, כמו הסוג שלו, דגם, יצרן, וביצועים.

`/proc/devices`—רשימה של מנהלי התקנים שמקונפגים אל תוך הקרנל שרץ כרגע.

`/proc/dma`—ערוצי DMA שנמצאים בשימוש כרגע.



cat /proc/filesystems --/proc/filesystems מערכות קבצים שמקונפגים אל תוך הקרנל.

cat /proc/interrupts --/proc/interrupts אותות פסיקה שנמתאים בשימוש, וכמה מכל אחד היו בשימוש.

cat /proc/ioprocs --/proc/ioprocs פורטים של I/O שבשימוש כרגע.

לדוגמא, אני יכול לקרוא את מידע לגבי המעבד שעל המערכת על ידי הפקודה הבאה:

```
cat /proc/cpuinfo
```

### 3.1.3 איך אני מריץ תוכנית?

הקלדת השם של התוכנית (executable) בשורת הפקודה לא מניבה תוצאות? ישנן שלוש אפשרויות.

**האפשרות הראשונה:** לא הקלדת את שם התוכנית בצורה תקינה. בדוק איך כתבת אותה - לינוקס רגישה לנושא הזה. לדוגמא, הקלדת "pico" או "PICO" לא יפעיל את העורך pico.

**האפשרות השנייה:** אולי התוכנית אינה בנתיב (path) שלך. תחת לינוקס (או יוניקס), executable חייב להיות בנתיב שלך כדי להריץ אותו, והספרייה הנוכחית אינה הנתיב שלך.

הקלד את הנתיב המלא אל exec לפני שמו, או בצע:

```
cd the_program_directory
./program_name
```

אתה חייב לשים נקודה וסלאש לפני שם התוכנית אחרת התוכנית לא תרוץ. (אי הקלדת הסיפרייה בנתיב היא תכונת אבטחה. היא מקשה קצת את החיים על "סוסים טרווינים". "סוס טרוויני" הוא למעשה תוכנה זדונית שמעמידה פני תוכנה אחרת ממה שהיא באמת.) הנקודה מתפרשת כ"ספרייה הנוכחית", והסלאש "/" מפריד בין שם הסיפרייה ושם הקובץ (בדיוק כמו בדוס).

אתה יכול לבדוק את הנתיב שלך על ידי:

```
echo $PATH
```

כדי ללמוד איך לשנות את הנתיב שלך, או כדי להוסיף את הספרייה הנוכחית שלך אליו, הסתכל בתשובה הבאה [3.1.4]

אם התוכנית שלך אבודה איפה שהוא בעץ הסיפרייה שלך, ייתכן ותרצה למצוא אותה על ידי (לדוגמא):

```
find / -name "netscpae"
```

כדי למצוא קובץ בשם "netscape" על ידי חיפוש מספריית השורש "/" . תוכל להשיג להשיג את אותה תוצאה בצורה מהירה יותר על ידי שימוש ב:

```
locate netscape
```

(locate פועלת מהר יותר מכיוון שהיא נשענת על מסד נתונים על קבצים שמערכת שלך אשר נבנה מראש. מסד נתונים זה מעודכן על ידי הליך cron שבדרך כלל רץ בלילה, אז אל תסמוך על locate אם אתה נוהג לכבות את המחשב שלך בלילה, או אם אתה מחפש קובץ שזה עתה התקנת).

שים לב בבקשה שהנתיב בדרך כלל שונה למשתמש root ולמשתמשים רגילים (הנתיב של root כולל את /sbin: /usr/sbin בעוד שהנתיב של שאר המשתמשים לא). לכן משתמשים לא יכולים להריץ פקודות שממקומות בספריות "/sbin" אלא אם הם מציינים את הנתיב המלא אל הפקודה. בנוסף אם אתה הופך להיות סופר יוזר (super user) על ידי הרצת הפקודה su, אתה יורש את הנתיב של המשתמש, וכדי לבצע את הפקודה שממוקמת ב /sbin, עליך לציין את הנתיב המלא.

באופן הפוך, אם אתה צריך לדעת היכן ממקומת תוכנית שנמצאת בנתיב שלך על גבי המערכת (התוכנית רצה על ידי הקלדת שמה בכל מקום במערכת, אבל אתה מעוניין לדעת היכן היא נמצאת), אתה יכול להשתמש במשהו כמו זה:

```
which netscpae
```

which תראה לך את הנתיב המלא אל התוכנית שנקראת netscape .

**האפשרות השלישית:** אולי הקובץ אינו בר הרצה. אם הוא אמור להיות, שנה את ההרשאות כדי להפוך אותו לכזה. (כroots או כמשתמש שהוא בעל הקובץ):

```
chmod a+x my_file
```

פקודה זו תהפוך את הקובץ "my\_file" לבר הרצה לכל המשתמשים. בדוק אם זה עובד על ידי:

```
ls -l my_file
```

שים לב בבקשה שתחת לינוקס (או יוניקס), אין זו הסיומת של הקובץ (exe או com או bat למשל) שהופכת את הקובץ לבר הרצה. הקובץ צריך להיות במצב גישה שמאפשר לו להיות כזה ("executable file access mode") שאינו דומה כלל לאופן בו עובדים האטריביוטים של קבצים תחת דוס.

### 3.1.4 כיצד אני יכול לשנות את הntיב?

בדרך כלל, אינך צריך לשנות את הntיב שלך, אבל זה מועיל מאוד להבין מה זה נתיב (PATH).

הנתיב הוא רשימה של ספריות שבהן נערך חיפוש כשאתה מבקש להריץ תוכנית. אתה יכול לבדוק את הntיב שלך על ידי שימוש בפקודה זו:

```
echo $PATH
```

שבמערכת שלי, מראה שהנתיב של המשתמש yogin הוא:

```
/opt/kde/bin:/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/home/yogin/bin
```

הנקודותיים ":" משמשות כמפרידות לכן הntיב מייצג רשימת ספריות כלהלן:

```
/opt/kde/bin  
/usr/local/bin  
/bin  
/usr/bin  
/usr/X11R6/bin  
/home/yogin/bin
```

הנה פלט מהפקודה "echo \$PATH" תחת המשתמש root במערכת שלי:

```
/opt/kde/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/root/bin
```

אתה יכול לשנות את הntיב לכל המשתמשים במערכת על ידי עריכה של הקובץ /etc.profile והתאמת השורה שמתחילה ב "PATH=". אני עושה את זה על ידי שימוש בעורך (root) pico:

```
pico --w /etc/profile
```

(האופציה -w מכבה את warp לשורות ארוכות.)

היכנס מחדש למערכת כדי שהשינוי יוחל. כדי לקבוע את הntיב למשתמש יחיד בלבד, ערוך את הקובץ /home/user\_login\_name/.bahe\_profile (שים לב בבקשה לנקודה לפני שם הקובץ--קבצים שמתחילים בנקודה בדרך כלל בלתי נראים, אתה צריך להשתמש ב ls -a על מנת לראות אותם).

אם אתה באמת מעוניין שהספרייה הנוכחית תהיה בנתיב שלך, הוסף "." (נקודה) לנתיב שלך.

כשבשימוש במקום בו אמור להיות שם הספרייה, נקודה מתפרשת כ"ספרייה הנוכחית". ההנחיה לנתיב ב `/etc/.bash_profile` יכולה להראות כך:

```
PATH="$PATH:$HOME/bin:."  
export PATH
```

פקודה זו לוקחת את התוכן של המשתנה הסביבתי שנקרא PATH (כפי שנקבע לכל המשתמשים ב `/etc/profile`), ומחיל עליו את השם של ספריית הבית שלך כפי שנקבע על ידי המשתנה HOME בתוספת `"/bin"` ואז נקודה.

לבסוף, הפקודה מחילה את חזרה את המשתנה שנקרא PATH. זה הכרחי להשתמש בפקודה `"export"` לאחר שינוי PATH או כל משתנה סביבה שקשור למשתמש, ככה שמשנתנה הסביבה נראה מחוץ לסקריפט שקובע אותו.

### 3.1.5 איך אני יכול לכבות את המחשב שלי?

בטרמינל טקסט, לחץ על `<Ctrl><Alt><Del>` (הצדעת "שלוש האצבעות", לוחצים על שלושת המקשים ביחד), חכה שתהליך הכיבוי ייסתיים, ואז כבה את המכונה שלך רק לאחר שהיא מאתחלת שוב.

אם אתה נמצא ב X-windows, קודם עבוד לטרמינל טקסט על ידי לחיצה על `<Ctrl><Alt><f1>` (ביחד). אל תכבה את המכונה שלך ללא הכיבוי הראוי מכיון שאתה עלול להיתקל בהודעות שגיאה לגבי הדיסק בפעם הבאה שתאתחל. (בדרך כלל, השגיאות שנובעות מכיבוי לא תקין ייתוקנו באופן אוטומטי בזמן האיתחול הבא, אבל לפעמים בעיות רציניות יותר יכולות לנבוע מכיבוי לא נכון, ואז תאלץ לתקן את הקבצים ידנית או להתקין מחדש!)

אם אתה מעדיף שהמחשב שלך יעצור (יכבה) לאחר שאתה לוחץ `<Ctrl><Alt><Del>` (במקום האיתחול הדיפולטי), אתה יכול לקבוע את זה על ידי עריכת הקובץ `/etc/inittab`. הקובץ מציין משהו כמו זה:

```
#trap CTRL-ALT-DELETE  
ca::CtrlAltDel:/sbin/shutdown -t3 -r now
```

כ `root`, החלף את האופציה `"-r"` ל"`-h`".

השורה שמתחילה ב"`#`" היא רק הערה (לבני אנוש בלבד, אין לה כל השפעה על המחשב). האופציה `"-t3"` מורה לפקודת הכיבוי (`shutdown`) לחכות 3 שניות לפני שהיא מתחילה להרוג תהליכים.

האופציות "-r" ו"-h" מתפרשות כ"reboot" ו"halt" בהתאמה, ככה שהם מבצעות shutdown לאיתחול או לכיבוי המערכת.

root יכול גם להשתמש ישירות בפקודה shutdown. פקודה זו יכולה לשמש גם למחשב מקומי וגם למחשב מרוחק, אבל משתמשים בה בעיקר לכיבוי מרוחק כשהמקלדת המקומית אינה זמינה כך שאינך יכול להשתמש ב <Ctrl><Alt><Del>. היא יכולה להועיל גם אם תוכנית תוקעת ומנטרלת את המקלדת.

לדוגמא:

```
telnet name_of_machine_with_no+operable_keyboard
[login as a user]
su
[give password]
```

קעת או שתבצע more | ps axu, מצא את ה id ששייך לתהליך של הפקודה הפוגעת בפלט של ps ובצע:

```
kill pid_of_offending_process
```

או שתבצע איתחול מחדש למחשב שלך עם:

```
/sbin/shutdown -rn now
```

פקודה זו תבצע כיבוי מהר מאוד, ותדלג על פרוצדורת הכיבוי (הארוכה יותר) – שימושי כשהמערכת ממש מבוגגת (האופציה -n תגרום ל"shutdown" להרוג את כל התהליכים לפני האיתחול המחודש).

שים לב בבקשה שמסיבות שקשורות לאבטחה, אינך יכול להיכנס למכונה אחרת כroot (דרך telnet). אתה חייב להיכנס למערכת כמשתמש ואז לבצע su לתת סיסמא ואז להפוך לsuperuser (root).

הפקודה shutdown יכולה גם לבצע כיבוי מאוחר יותר (כ root):

```
/sbin/shutdown -r 23:59
```

תאתחל את המחשב דקה לפני חצות. אני יכול להשתמש גם ב:

```
/sbin/shutdown -r +1
```

שאתחיל את המערכת דקה מעכשיו. אני יכול לבטל כיבוי מתוזמן בעזרת:

```
/sbin/shutdown -c
```

אם הפקודה shutdown ארוכה מדי עבורך, ייתכן ותרצה להשתמש ב שתי הפקודות האלו, שייבצעו בדיוק את מה שהשמות שלהן מציעים (כ root):

```
reboot  
hAlt
```

דרך קלאסית לכבות את המחשב שלך תהיה להעביר את המערכת שלך ל runlevel 0 או ל (hAlt) או ל runlevel 6 (reboot) נסה את זה על ידי (כ root):

```
init 0
```

```
init 6
```

המשמעות של runlevels השונים מוסברת בקובץ /etc/inittab וכאן.

### 3.1.6 איך אני מתמודד עם תוכנית תקועה?

תוכניות מבוגגות אכן נתקעות תחת לינוקס. קריסה של אפליקציה לא אמורה, בכל מקרה, להשפיעה על מערכת ההפעלה עצמה כך שאינך צריך לאתחל את המחשב שלך לעיתים תכופות מדי.

שרתי לינוקס ידועים ככאלה שיכולים לרוץ במשך יותר משנה ללא איתחול מחדש של המערכת. מהנסיון שלנו, מערכת הפעלה שמתנהגת בצורה לא טובה יכולה להיות סימן לבעיות חומרה או קונפיגורציה:

באופן חוזר ונשנה נתקלנו בבעיות חימום ייתר של מעבד פנטיום (המעבד או המאוורר לא פעלו במהירות שהם היו אמורים לפעול בה או שהם עצרו, חריצי האיוורור שעל גבי הפנטיום היו ספוגים בליכלוך).

צ'יפים גרועים של זיכרון, תיזמונים שונים של צ'יפים שונים של זכרון (אתה יכול לנסות לשנות את הסדר של הצ'יפים, זה יכול לעזור), כיוונון לא נכון של BIOS (אתה יכול לכבות את כל ההגדרות ה"מתקדמות", לינוקס מטפלת בדברים בעצמה). הודעת השגיאה "signal 11"

קשורה בדרך כלל (99%) עם בעיות חומרה וסביר להניח שהיא תופיע כשאתה תבצע משימות מורכבות ואינטנסיביות על גבי המחשב: התקנת לינוקס, קומפילציית קרנל וכו'. אם לפנטיום שלך יש נטייה לחימום ייתר (נפוץ מאוד בפנטיומים מוקדמים), לפניך מספר טיפים כדי לשמור עליו קר, במיוחד בזמן מזג אויר חם: נקה את חריצי האיוורור שך המעבד, החלף את המאוורר של המעבד, תפעל את המחשב ללא המכסה וכוון מאוורר

פנימה, הגבר את מצב (wait state) בביוס, הקטן עומס מיותר, אל תבצע over clocking החלף את שומר המסך הסופר מפואר שלך במסך ריק.

**לא באמת תקועות.** חלק מהתוכניות יכולות ליצור רושם של תקועות, בעוד במציאות הן רק מחכות לקלט מהמשתמש. בדרך כלל, זה קורה כשהתוכנה מחכה לשם קובץ כארגומנט של שורת הפקודה ואף שם קובץ לא ניתן לה על ידי המשתמש, אז התוכנית נכנסת למצב ברירת המחדל שלה לקלט סטנדרטי (שהוא הקונסולה). לדוגמא, הפקודה הזו:

```
cat
```

יכולה להיראות כאילו היא תקועה בעוד למעשה היא מחכה לקלט מהמקלדת. נסה ללחוץ על `<Ctrl>d` (שמשמעותו "סוף החיים") כדי לראות שזה יספק את הפקודה `cat`. דוגמא נוספת: ראיתי הרבה שאלות בקבוצות דיון לגבי הפקודה ה"מבוגבת" `tar` ש"נתקעת" בזמן שמנסים לפרוס קובץ למשל:

```
tar -zxv my_tar_file [worng!]
```

גם זה מחכה לקלט מהמשתמש, מאחר והאופציה `"-f filename"` לא צויינה, השם `"my_tar_file"` לא הוכר כשם קובץ. הפקודה הנכונה היא:

```
tar -zxvf my_tar_filename
```

שים לב בבקשה ששם הקובץ חייב להופיע מייד אחרי האופציה `"f"` (שמייצגת `"filename"`). זה לא יעבוד (טעות נפוצה מאוד):

```
tar -zxfv my_tar_file [worng]
```

### **כל תוכנית (תקועה או לא) יכולה להיהרג.**

תוכנית במצב גרפי שרצה בחזית יכולה להיהרג על ידי לחיצה על `<Ctrl>c`. זה לא יעבוד עם אפליקציות גדולות שחוסמות את `<Ctrl>c`, כדי שבטעות לא יישתמשו בשילוב הזה כדי לפגוע בהם. עדיין אתה יכול לחזור לשליטה על ידי שליחת התוכנית לרקע על ידי שימוש ב `<Ctrl>z` (אין ערובות שזה יעבוד) או החלפה לטרמינל אחר, על ידי שימוש ב `<Ctrl><Alt><f2>` למשל וכניסה למערכת בשם אותו משתמש שתקע את התוכנית (זה אמור לעבוד תמיד). ברגע שאתה שוב בעיניינים, מצא את בתוכנה שאתה רוצה להשמיד, לדוגמא:

```
ps
```

פקודה זו מתפרשת כ `"print status"` ומראה את רשימת התוכניות שמורצות כרגע על ידי המשתמש הנוכחי. בפלט של `ps`, אני מוצא את מזהה התהליך (`process id=PID`) של התוכנית שנתקע וכעת אני יכול להרוג אותה. לדוגמא:

```
kill 123
```

תהרוג את התוכנית עם מזהה התהליך (PID) `"123"`.

כמשתמש, אני יכול להרוג רק את התהליכים ששייכים לי (אלו שאני התחלתי). root יכול להרוג כל תהליך. כדי לצפות ברשימה של כל התהליכים שרצים במערכת הקלד:

```
ps axu | more
```

זה מציג את כל התהליכים שרצים כרגע (האופציה "a" ), אפילו את אלו ללא הטרמינל (האופציה "x" ), וביחד עם שם הלוג אין של המשתמש שלו שייך כל תהליך ("u"). מאחר וסביר להניח שהתצוגה תהיה ארוכה יותר מן המסך, השתמשתי באופציה "more" כדי שהתצוגה תעצור לאחר כל מסך מלא.

לפקודה kill יש את קיצור הדרך killall כדי להרוג תוכניות לפי שם, לדוגמא:

```
killall netscape
```

שתהרוג כל תוכנית עם "netscape" בשם שלה, בעוד

```
killall pppd
```

קרוב לוודאי תנתק כל חיבור חיוג על ידי הריגת השירות pppd (daemon).

לתוכניות מבוססות x-windows אין טרמינלי שליטה ויכולות להיות הכי קלות להריגה על ידי שימוש בפקודה זו (מוקלדת בטרמינל x):

```
xkill
```

כשמקלידים את הפקודה הזו, סמן העכבר הופך למשהו שדומה לגזר דין מוות; ואז כשאתה מצביע על החלון של התוכנית אותה רוצה להרוג ומקליק על הכפתור הימני בעכבר החלון נעלם והתוכנית המקושרת אליו מחוסלת.

קיצור דרך אל הפקודה האחרונה הוא לחיצה על `<Ctrl><Alt><esc>`, ושוב אותו דבר לחיצה על החלון בעזרת הסמן עם גזר הדין והתוכנית המקושרת נעלמת יחד עם החלון.

אם מערכת x-windows שלך קורסת באופן כזה שהיא אינה יכולה להשתקם, או שאתה פשוט נתקע, זו יכולה להיות הכי קל להרוג את שרת הא על ידי לחיצה על `<Ctrl><Alt><backspace>`. לאחר מכן, ייתכן וזה יהיה רעיון טוב להריץ את ps axu, כדי למצוא תוכניות x שעדיין רצות, ולהרוג אותן. אם לא תעשה את זה, ובאמת יש תוכנית שמתנהגת לא טוב שגרמה למערכת x-windows שלך לקרוס, היא עלולה לגרום לצרות שוב.

אם יש לך תוכניות ברקע, מערכת ההפעלה תמנע מבעדך לצאת מן המערכת עם הודעה כמו "there are stopped jobs". כדי להתגבר על זה ולצאת בכל זאת, כל מה שעליך לעשות זה לחזור על הפקודה logout או exit -- התוכניות שברקע יושמדו באופן אוטומטי ואתה תיצא החוצה.



## קבצי ליבה.

core files

כשתוכנית קורסת, היא בדרך כלל "זורקת" "core" לתוך ספריית הבית שלך. תהליך זה מלווה בהודעה מתאימה. "core" הוא העתק של זיכרון (בתוספת debugging) שנועד להיות כלי ל debugging.

אם אתה משתמש שאינו מעוניין לעשות debugging לתוכנית, אתה יכול פשוט למחוק את coren :

```
rm core
```

או לא לעשות דבר (במידה ויזרק core חדש הוא ייכתוב על גבי הישן). אתה יכול גם לבטל את זריקת coren על ידי הפקודה:

```
ulimit -c 0
```

בדוק אם זה עובד על ידי:

```
ulimit -a
```

(זה מראה את ה"user limits", האופציה -a מתפרשת כ "all").

כדי להפוך את ביטול זריקת coren לקבוע לכל המשתמשים, ערוך את הקובץ /etc/profile (root) כשהוא ulimit מכוון. היכנס שוב למערכת כדי שהשינויים ל /etc/profile ייכנסו לתוקף.

אם אתה רוצה לראות כיצד ניתן להשתמש בקובץ core, (בספרייה שבה יש לך קובץ core):

```
gdb -c core
```

פקודה זו מפעילה את debugger של GNU על קובץ coren בשם "core" ומציגה את שם התוכנית שיצרה את coren הזה, את הסיגנל שבו תוכנה זו הושמדה, וכו'. הקלד quit כדי לצאת מהdebugger.

כדי ללמוד על המשמעות של הסיגנלים השונים, נסה את:

```
cat /usr/include/bits/sugnum.h | more
```

### 3.1.7 אפשרויות פקודה

רוב הפקודות מקבלות "אופציות" רבות. ניתן להציג אופציה עם "-" (מקף). לדוגמא:

`dir -l`

מראה את רשימת הספרייה הנוכחית בפורמט ארוך (פורמט ברירת המחדל הוא קצר). ניתן להציג כמה אופציות בשתי דרכים מקבילות:

`dir -l -a`

או

`dir -la`

כל אחת מהפקודות למעלה תציג לי רשימה של הספרייה הנוכחית בפורמט של קובץ ארוך (האופציה -l) ותכלול את כל הקבצים ברשימה, גם את הקבצים הנסתרים (האופציה -a).

רוב האופציות הפופולריות מסומנות באות אחת. זה ממשיך באופן מסורתי את הדרך שבה משתמשים באופציות ביוניקס. ישנו גם סטייל חדש, שנראה כך:

`dir --help`

כאן, אורכה של אופציה אחת הוא יותר מאות אחת, והיא חייבת להיות מוצגת עם שני מקפים.

הפקודה למעלה מציגה עזרה זריזה לפקודה דיר, כולל רשימה של כל האופציות. מכיון שיש הרבה כאלו (יותר ממסך מלא) הייתי מקליד אותה כך:

`dir --help | more`

## 3.2 משתמשים, סיסמאות, הרשאות קבצים, ואבטחה

### 3.2.1 ספריות בית, root, הוספת משתמשים

המקום (הכמעט) יחיד על גבי הדיסק הקשיח אליו משתמשים (שאינם root) יכולים לכתוב הוא ספריית הבית שלהם, שהיא `/home/user_login_name`.

ספריית הבית מיועדת לכל הקבצים של המשתמש: הגדרות, קבצי קונפיגורציה של תוכניות, תיעוד, נתונים, המטמון של הדפדפן, דואר, וכו'. כמשתמש, אתה יכול ליצור תת ספריות תחת ספריית הבית שלך כדי לארגן את עצמך. משתמשים אחרים אינם יכולים לקרוא את הקבצים שלך או לכתוב אל ספריית הבית שלך, אלא אם כן תיתן להם רשות לעשות את זה.

משתמשים רגילים יכולים לראות, לכתוב ולבצע קבצים רבים אחרים במערכת (בנוסף לספריית הבית שלהם), אבל בדרך כלל הם אינם יכולים לשנות או להסיר (למחוק) אותם.

root (נקרא גם super user) הינו חשבון אדמיניסטרטיבי מיוחד, בעל היכולת לשנות כל קובץ במערכת. אין זה מומלץ לעבוד כroot על המערכת באופן שוטף--אם לא תנהג כך יש סיכוי שתשלם מחיר גבוהה על הטעויות שלך. הקם לך חשבון משתמש רגיל לעבודה יומיומית, חשבון נוסף לבנך, ואחד נוסף לאישתך.

החשבון root הוא בדרך כלל החשבון היחיד שקיים במערכת לאחר ההתקנה (אם לא יצרת חשבונות נוספים במהלך ההתקנה עצמה-א.א.).  
לכן עליך ליצור חשבונות משתמשים לעבודה רגילה בלינוקס באופן יזום.

חשבון משתמש יכול להיווצר על ידי root כך למשל:

```
adduser joe
passwd joe
[הקלד את הסימא למשתמש joe]
[הקלד שוב פעם את הסימא למשתמש joe כדי להימנע מטעויות]
```

לכן קודם כל נכנסתי למערכת כroot. אחר כך, בשורת הפקודה, הקלדתי את הפקודה "adduser" עם הפרמטר "joe". זה יצר את החשבון "joe" בלינוקס שלי. לאחר מכן, הקלדתי את הפקודה "passwd joe" כדי לשנות את הסימא למשתמש "joe" למשהו קצת מאובטח.

כעת אני יכול לומר ל"joe" מהי הסימא שלה, ואז היא יכולה להיכנס למערכת ולשנות את הסימא שלה לפי מה שמתאים לה. שים לב שגם שם המשתמש וגם הסימא הינם case-sensitive.

root יכול לשנות את הסימאות של כל המשתמשים, למרות שהוא/היא אינם יכולים לקרוא אותן. [הסימאות מוצפנות, על ידי one way alogrithm ורק הגירסא המוצפנת מאוחסנת על גבי המערכת, בקובץ `/etc/passwd` (מערכות ישנות) או ב `/etc/shadow` (מערכות חדשות יותר, הגירסא ה"פתוחה" של הסימא לעולם אינה מאוחסנת. כשאתה נכנס למערכת, הסימא שאתה מקליד מוצפנת שוב באותו אלגוריתם (one way) ומושווה אל הגירסא המוצפנת שמאוחסנת ב `/etc/passwd` או `/etc/shadow`].

ההפרדה בין האדמיניסטרטור לבין המשתמש הופכת את מערכות לינוקס למאובטחות ועמידות--היא אפילו הופכת את החיים קשים יותר לוירוסים תחת לינוקס (התוכנית שמשתמש מריץ יכולה לכתוב רק אל תוך הספריות שלו ולכן אינה יכולה להשפיע על חלקים חיוניים במערכת ההפעלה).

הנוהג הוא, שמשתמש משנה את הסיסמא שלו מייד לאחר הכניסה הראשונה שלו למערכת, לדוגמא:

```
passwd
(current) UNIX password: pass_OLD
new UNIX password: pass_NEW
retype new UNIX password: pass_NEW
```

במציאות הסיסמא לא תופיע על המסך בזמן שאתה מקליד אותה (מסיבות שקשורות לאבטחה). קח את הזמן אם אתה משנה את הסיסמא שלך בפעם הראשונה--זה יכול להיות קשה להקליד על "עיוור".

בלינוקס, אותה סיסמא משמשת ל: כניסה לטרמינל טקסט. כניסה לממשק גרפי. שיחרור מנעילה של טרמינל טקסט. שיחרור מנעילה של שומר מסך מוגן סיסמא בממשק גרפי.

### 3.2.2 קצת על אבטחת סיסמא

סיסמאות חלשות הינם בדרך כלל המקור השכיח ביותר לבעיות אבטחה. אפילו בבית, אתה עלול לחשוף את עצמך לצרות רציניות בגלל שמישהו יכול לפרוץ אל המחשב שלך בזמן שאתה גולש באינטרנט ולמחוק/לקרוא את הקבצים שלך, או להשתמש במחשב שלך כדי לעשות משהו באמת מרושע למחשב של תחנת המשטרה המקומית. לכן, שמור את כל השמות/סיסמאות שלך מאובטחים, אפילו בבית. ברגע שמישהו ייכנס אליך למחשב, (אפילו כמשתמש רגיל), הוא יכול להשיג גישה root די בקלות (תלוי עד כמה המערכת שלך מתוחזקת/מעודכנת +עד כמה הוא האקר טוב).

לפניך כמה דוגמאות לסיסמאות מסוכנות:

- שום סיסמא(אפשרי!)
- המילה "password" (וואו, זו באמת אחת חלשה)
- שם הכניסה שלך (שם הלוג אין והסיסמא זהים? המממ...)
- השם הפרטי שלך או השם הפרטי של בתך, בנך, בעלך, אשתך, של החברה שלך, או כל שם פרטי אחר. מספר השמות הפרטיים הוא די מוגבל.
- שם המשפחה שלך או כל שם משפחה אחר. מספר שמות המשפחה מוגבל באופן מפתיע! אתה יכול לבדוק בנתונים של מפקד התושבים האחרון שנעשה בארה"ב ולהיות מופתע מהעובדה ששם המשפחה המשפחה ה"נדיר" שלך מהאי אבהמה מיוצג בצורה מוכבדת מאוד בין 89,000 שמות המשפחה השכיחים ביותר בארה"ב.

<http://www.census.gov.genealogy.html>

- שם החיבה של הכלב שלך, של האישה שלך, של התוכי הקנרי שלך או של המחשב שלך (בני אדם משתמשים במבחר מצומצם מאוד של שמות חיבה, בוודאי שפחות משמות פרטיים).

- השם של קבוצת הספורט האהובה עליך, סלבריטי, משחת השיניים שלך.  
הזהר משמות של קבוצת כדורגל פופלרית כמו מאש. כנ"ל לגבי להקות רוק (מוזיקה).  
- תאריך הלידה שלך, מספר תעודת זהות וכו'; רצף של ספרות יכול להיחשף בקלות.  
- שם החברה שלך, מחלקה, קבוצת עבודה וכו'.  
- סיסמא שכתובה על לוח השנה שלך או בצד המחשב.  
- סיסמא שמשמשת אותך במקומות ציבוריים נוספים, למשל חנות באינטרנט או רשימת תפוצה. באופן כללי עליך להשתמש בסיסמאות שונות למקומות שמנוהלים על ידי אירגונים שונים.

- כל מילה שנמצאת במילון האנגלי. המילון האנגלי לא מכיל כזו כמות של מילים כמו שזה נראה. האקר לא כל כך מיומן יכול להכין תוכנה שתצפין את כל המילים במילון (100,000? זה פחות מ1MB!) ולאחר מכן להשוות את כל המחרוזות המוצפנות אל מול הסיסמא המוצפנת שלך. נסה את התוכנית crack בעצמך כדי לגלות עד כמה זה קל.  
מילות שבועה, ביטויי שפה "מגניבים" הופכים את הסיסמא לפגיעה מאוד לפיצוח.  
- כל מילה אחרת, שם משפחה, שם פרטי, שם חייט מחמד, ולא משנה באיזו שפה.  
לקראקר, כיסוי רוב השפות זו רק קצת יותר עבודה במידה והוא כבר כיסה שפה אחת. כמה שפות משמעותיות יש בעולם? 40? הקראקר רק תופס עוד כמה קבצים ומוסיף אותם אל רשימת הפיצוח שלו. הנקודה כאן היא שסך כל המילים שבני אדם בדרך כלל משתמשים בהן נמוך מגבול הקומבינציות הרנדומליות שניתן ליצור באופן תאורטי משילוב של תווים.  
- כל ה"ל" בשילוב של אות/מספר בסוף או בתחילת הסיסמא כמו "yuoping1" למשל, יוצר באמת סיסמא חלשה.

סיסמא טובה צריכה להיות ארוכה באופן יחסי (מינימום 6 תווים, כמה מומחים ממליצים אפילו על מינימום של 10 תווים), שמכילה שילוב של אותיות ( גדולות וקטנות, אם אפשרי), ספרות ותווים מיוחדים, ושמתנה באופן די קבוע (8-16 שבועות?).

למרבה הצער, ככל שהסיסמא טובה יותר, כך קשה יותר לזכור אותה. אני פותר את הבעייה הזו באופן כזה, אני לוקח לעצמי 10 דקות כדי להמציא את ה"סכמה" של הסיסמא שלי. נאמר, אני תמיד מתחיל ומסיים את הסיסמא שלי עם "שטרודל" (@), ומשתמש בשתי מילים מחוברות ל סימן קריאה, כשהאות האחרונה בכל מילה היא אות גדולה, משהו כמו: "@whutE!houSE@". נראה כמו סיסמא ראויה, שקל לזכור אותה ברגע שאני יודע מה החוק של הסיסמא שלי. אם אתה בעל זיכרון גאוני, אתה יכול לשקול סיסמאות מצויינות באמת שמיוצרות על ידי mkpasswd ()):

מנהל המערכת, האדמיניסטרטור, יכול לקבוע מדיניות סיסמא (אורך מינימלי, דרישה לתווים מיוחדים, תאריך תפוגה) דרך כלי שכלול בתוכנית הקונפיגורציה הבאה (root):

linuxconf

תחת התפריט "user account - policies - password & account policies" משתמשים רגילים לא יוכלו לקבוע לעצמם סיסמא קצרה מדי, מילה שמופיע במילון, או שאינה מכילה את כמספר המומלץ של תווים שאינם מן האלף בית ( root יכול לשנות כל סיסמא לכל מה שעולה בדעתו, אבל יקבל אזהרה).

בנוסף וודא שלכל קובץ שמכיל כל סיסמא שלך ( /root/.kde/share/config/kppprc/ ) ישנן ההרשאות המתאימות ככה שאף אחד אחר לא יוכל לקרוא אותו. למשל, קורב לוודאי שתרצה:

```
chmod 600 kppprc
```

אם אתה משתמש בחיבור אינטרנט אנלוגי דרך טלפון רק למשך כמה שעות בשבוע, אתה יכול להסתדר אפילו עם סיסמא חלשה יחסית במערכת. אבל חשוב על אבטחת המערכת שלך אם אתה משתמש במודם כבלים, adsl, או אם אתה מחובר לאינטרנט למשך פרקי זמן ארוכים.

רוב משתמשי המחשב הסטנדרטיים משתמשים בסיסמאות חלשות להדהים. "קרוב ל-50 אחוזים ממשתמשי המחשב **מבוססים** את הסיסמאות שלהם על שם של קרוב משפחה, שותף לחיים או חיית מחמד. שלושים אחוזים משתמשים בשם של אליל הפופ או הספורט שלהם" כך מדווחת רשת CNN

[/http://www.cnn.com/2002/TECH/ptech/03/13/dangerous.passwords](http://www.cnn.com/2002/TECH/ptech/03/13/dangerous.passwords)

שים לב בבקשה לדגש על **מבוססים**. הוספת סיפורה למילה פשוטה כמעט שאינה הופכת את הסיסמא למאובטחת יותר.

### 3.2.3 שכחתי את הסיסמא של root

אפילו שאיני שוכח סיסמאות בכלל, הייתי לומד את הנושא הזה בפירוט בכל זאת מכיוון שהוא יכול לתת לי רמז לגבי איך אמא שלי עלולה לקרוא את היסטוריית שיחות הצ'אט שלי ICQ. (-):

**שיטה ראשונה.** הדרך הקלה ביותר לפתרון בעיית ה"שכחתי את הסיסמה" היא לאתחל את לינוקס בsingle-user mode, כל מה שעליך לעשות הוא פשוט להקליד את הפקודה הבאה בזמן promptn של LILLO :

```
linux single
```

זה יהפוך אותך לroot בלי לבקש ממך סיסמא. כעת כשאתה root, אתה יכול לשנות את הסיסמא של root על ידי שימוש בפקודה (אין צורך לדעת את הסיסמא הישנה) :

```
passwd
```

אם זה נשמע לך לא מאובטח, זה בגלל ששום מערכת הפעלה אינה מאובטחת אם לאנשים יש גישה פיזית אל החומרה שלך. אני באופן אישי לא אהבתי את פירצת האבטחה ש"linux single" יצר במחשב הביתי שלי וסתמתי אותה על ידי הוספת השורות הבאות לקובץ /etc/lilo.conf (בסוף של "image=")

```
password="my_password"  
restricted
```

[הסיסמא של "LILLO" דרושה למצבים בהם מישהו מקליד את המילה "linux" עם פרמטר כל שהוא prompt של lilo בזמן האיתחול (איתחול רגיל ללא כל פרמטרים יהיה עדיין אפשרי ללא כל סיסמא)]. כדי שהשינויים שנעשו ב /etc/lilo.conf ייושמו, אני חייב להריץ מחדש את הפקודה lilo. מכיון שסיסמת הול של lilo אינה מאובטחת, עליי להפוך את /etc/lilo.conf לקריא אך ורק על ידי root:

```
chmod 600 /etc/lilo.conf
```

[ הערת המתרגם: ניתן ליישם *linux single* גם אם אתה משתמש ב*grub* ולא ב*lilo*. העיניין מצריך קצת אקרובטיקה אבל בסך הכל עובד באופן די פשוט. בדקתי את זה וזה עבד, עשיתי את זה ככה:

1. במידה והתקנת את *grub* עם סיסמא, הקלד *p* והכנס את הסיסמא.

2. בחר את גרסת הלינוקס שלך והקלד *e* כדי לערוך.

3. בחר בשורה שמתחילה ב*kernel* ושוב, הקלד *e* כדי לערוך את השורה.

4. בסוף השורה הקלד רווח ואז *single*. לחץ על *ENTER* כדי לצאת ממצב העריכה.

5. חזור אל תפריט מערכות ההפעלה ואתחל את לינוקס. המערכת תאותחל אל *runlevel*

המבוקש. --א.א.]

**שיטה שניה.** דרך נוספת לפתרון בעיית "סיסמת root האבודה" היא לאתחל את המחשב שלך מדיסק (פלופי) האיתחול של לינוקס או מה *cd* של לינוקס. ואז, למצוא את המחיצה של root על גבי הדיסק הקשיח, להמיר אותה, ולערוך את הקובץ */etc/shadow*. (אני יכול לעשות את זה מכיון שלאחר איתחול מהפלופי אני נהיה root בלי שאתבקש להקליד סיסמא). בקובץ הסיסמא, אני מוחק את הסיסמא המוצפנת לroot (על ידי שימוש בעורך *pico* למשל), כך שלבסוף היא ריקה.

מידע על חשבון משתמש נמצא בקבצי הטקס הלא מוצפנים: */etc/passwd* וב-  
*/etc/shadow*.

הקובץ */etc/passwd* מכיל מידע קריא מאוד על כל החשבונות במחשב שלי. בכל שורה יש 7 שדות תחומים על ידי נקודותיים (זה אומר 8 פסקאות שמופרדות על ידי נקודותיים): שם *login*, האות "x", המזהה המספרי של המשתמש, המזהה הראשי של קבוצת המשתמש, שדה להערה (לדוגמא, השם המלא של המשתמש), ספריית ה *\$HOME* של המשתמש, שם המעטפת (הכוונה היא לתוכנית שרצה ב*login*).

האיזון של המידע לגבי חשבונות במחשב שלי מאוחסן בקובץ */etc/shadow*. קובץ זה מאובטח יותר מכיון שבאופן נורמלי רק root יכול לקורא אותו. בקובץ זה כל שורה מתארת מידע מוצל ("shadow" information) על כל חשבון משתמש, ומכילה 9 שדות תחומים

בנקודותיים: שם login, סיסמא מוצפנת, מספר הימים שחלפו מהשינוי האחרון של הסיסמא מאז 1 לינואר 1970, מספר הימים שבהם עדיין מותר לשנות את הסיסמא, מספר הימים שאחריהם חייבים לשנות את הסיסמא, מספר הימים לפני תפוגת הסיסמא על מנת להזהיר את המשתמש, מספר הימים לאחר תפוגת הסיסמא שבהם החשבון מבוטל, ושדה שמור.

ישנן מערכות יוניקס ולינוקס (ישנות יותר) שלא מכילות את הקובץ `/etc/shadow` ומאחסנות את הסיסמא המוצפנת של המשתמש בשדה השני בכל שורה בקובץ `/etc/passwd` (השדה שבמערכות חדשות יותר מכיל רק את האות x).

לדוגמא, השדה שמופיע אחרי `root` בקובץ `/etc/shadow` שלי נראה בערך ככה:

```
root:$1$BuPbmLAz$ 1G7 . evlChyqaEI0TIZp0F . :11071:0:99999:7:-1:-1:134540356
```

ולאחר שהסיסמא נמחקה, הוא נראה ככה:

```
root: :11071:0:99999:7:-1:-1:134540356
```

כעת, לחשבון של `root` אין סיסמא, לכן אני יכול לאתחל את המחשב שלי ובשורת הכניסה (login prompt) להקליד `root` ובמקום סיסמא פשוט ללחוץ `ENTER` (ריק, ללא כל סיסמא). לאחר כניסה מוצלחת למערכת, אני מייד קובע סיסמא ל `root` על ידי הפקודה:

```
passwd
```

נראה שלמרות מחיקת הסיסמא מ `/etc/shadow`, ההפצה דביאן לא תאפשר לך להיכנס למערכת ללא סיסמא (אבטחה משופרת?). במקרה כזה, מה שעליך לעשות זה להחליף את הסיסמא בקובץ `/etc/shadow` בסיסמא מוצפנת של חשבון אחר, שאת הסיסמא שלו אתה כן מכיר, לאחר מכן, אתה יכול להיכנס למערכת מכיוון שאתה יודע את הסיסמא.

שליחה של סיסמא מוצפנת בדואר אלקטרוני יכולה להיות דרך להקמת חשבון למישהו מרוחק: "אני מקים חשבון ftp על השרת שלי בשבילך. שלח אלי במייל את הסיסמא המוצפנת שלך". לאחר שקיבלת במייל את הסיסמא המוצפנת, אתה מכניס אותה בשדה המתאים ב `/etc/shadow`. כעת, המשתמש יכול להיכנס למערכת, מכיון שהוא יודע את הסיסמא אבל פרט אליו איש לא יודע אותה.

כדי ליצור "גישת פלופי" אל המערכת שלי הייתי צריך להתאמץ קצת יותר, חשבתי להריץ מערכת ללא פלופי כלל, אבל לרוע המזל, התקליטורים של לינוקס ניתנים לאיתחול. כיוונתי את רצף האיתחול בביוס ככה שהמערכת תאתחל מהדיסק הקשיח ולא מהפלופי או מה `cd`, והוספתי סיסמא לביוס. עדיין, ניתן לשבור סיסמאות ביוס די בקלות, אפשר להסיר את הסוללה הקטנה שנמצאת על גבי לוח האם ולאפס את ההגדרות האלו, או שמישהו יכול פשוט להסיר את הדיסק הקשיח ולחבר אותו אל מחשב אחר...

אני שוקל להתקין מערכת קבצים מוצפנת שזמינה ללינוקס כעת, אבל בהתחשב בכל הטירחה שקשורה בהקמה כזו, אולי אתפשר על נעילת החדר שלי וזהו. אם הכל נשמע לך פראנואידי מידי, זה כנראה באמת כך--זה רק מחדד את העובדה שיש מעט מאוד אבטחת מידע גם תחת לינוקס, אם לפורץ יש גישה פיזית אל החומרה שלך.



### 3.2.4 שכחתי את סיסמת המשתמש שלי

אם משתמש רגיל, שאינו root, שוכח את הסיסמא שלו, זו אינה בעיה נכיון ש root יכול לשנות כל סיסמא. לדוגמא (root):

```
passwd barbara
```

יבקש ממך להקליד סיסמא חדשה ל "barbara" (ללא צורך בידע של הסיסמא הקודמת). אם משתמש רגיל (שאינו root) מבקש לשנות את הסיסמא שלו, הוא עדיין ייצטרך להקליד את הסיסמא הישנה לפני כן. (זו תכונת אבטחה כדי שאיש לא ישנה את הסיסמא שלך במידה והשאררת את הטרמינל שלך עזוב).

### 3.2.5 ביטול או הסרת חשבון משתמש

ניתן לבטל חשבון משתמש באופן זמני או להסיר אותו לחלוטין מהמערכת.

כדי לבטל חשבון באופן זמני (לנעול), אין כל צורך לשנות את הסיסמא שלו. כל מה שעליך לעשות זה להניח "\*" כוכבית בתחילת השדה השני (לפני הסיסמא המוצפנת) בקובץ /etc/shadow. ה"מתפרשת כ: "לחשבון זה אסור להיכנס למערכת". כשתרצה לשחזר את החשבון, מחק את ה"כוכבית וחשבון המשתמש ייחזור לפעולה, עם הסיסמא הישנה שלו.

הנה דוגמא מ הקובץ /etc/shadow עם הסיסמא המבוטלת של "peter":

```
peter:*$1$narMEFm6$fhA1puOU422HiSL5aggLI/:11193:0:99999:7:-1:-1:134539288
```

יכולתי גם לנעול את חשבון המשתמש עם הפקודה הבאה:

```
passwd peter -l
```

ולשחרר אותה בעזרת:

```
passwd peter -u
```

כדי להסיר חשבון משתמש באופן בלתי הפיך מהמחשב, אני מבצע את המהלכים הבאים:  
- נכנס למערכת כ root  
- משנה את הזהות שלי למשתמש שאני מעוניין להסיר, כדי לבדוק אם ישנו דואר חשוב:

```
su doomed_user_login_name  
mail  
logout
```

- מוחק את חשבון המשתמש ואת הקבוצה שלו.

```
userdel doomed_user_login_name  
groupdel doomed_user_login_name
```

מסיר את הקשר של המשתמש לכל הקבוצות האחרות:

```
usermod -G doomed_user_login_name doomed_user_login_name
```

- כופה מחיקה של ספריית הבית של המשתמש עם כל התכולה שלה כולל כל תת הספריות:

```
rm -fr /hom/ doomed_user_login_name
```

### 3.2.6 יש לי בעייה עם הרשאות קבצים, הרשאת קבצים ובעלות על קבצים, איך זה עובד?

לינוקס (בדיוק כמו יוניקס) הינה מערכת הפעלה מרובת משתמשים מאובטחת, וזה יוצר רמה מסויימת של מורכבות עם "הרשאות קבצים". בעיות עם הרשאות קבצים יכולות להוביל לצרות נבזיות ולא צפויות. הבנת הרשאת קבצים חיונית לניהול כל מערכת הפעלה מרובת משתמשים (יוניקס, NT, או לינוקס) העצה שלי תהיה:

למד את מערכת הרשאות הקבצים של לינוקס (או של כל יוניקס) ; אתה לא תצטער על זה.

**בעלות על קבצים.** כל קובץ (או ספרייה) שייכים לבעלים (בדרך כלל שם login) ולקבוצה.

הבעלים הוא בדרך כלל האדם שיצר (או שהעתיק) את הקובץ. הקבוצה בדרך כלל מכילה אדם אחד--הבעלים, ויש לה שם זהה לשם של הבעלים, אבל זה לא חייב להיות ככה. קובץ יכול להימחק אך ורק על ידי הבעלים של אותו קובץ, או על ידי חבר בקבוצה שהקובץ נמצא בבעלותה, או על ידי root. משתמשים אחרים, בכל אופן, יכולים למחוק או לשנות את תוכן הקובץ אם ניתנה להם הרשות לכך. הבעלים והקבוצה שבבעלותה הקובץ ייתגלו מהפלט של הפקודה `ls -l` ("list in the long format"). לדוגמא, הפקודה הבאה:

```
ls -l junk
```

תפיק את הפלט הזה על המסך שלי:

```
-rwx----- 1 yogin inca 27 Apr 24 14:12 junk
```

זה מראה את הקובץ "junk", שנמצא בבעלות "yogin" והקבוצה "inca".

הבעלות על הקובץ ניתנת לשינוי על ידי שימוש בפקודות `chown` (change owner) ו-`chgrp` (change group), שבדרך כלל מבוצעות על ידי root:

```
chown peter junk
```

```
chgrp peter junk
```

```
ls -l junk
```

לאחר ביצוע 3 השורות שלמעלה, הפקודה `ls -l junk`, מפיקה את הפלט הזה על המסך שלי:

```
-rwx----- 1 peter peter          27 Apr 24   20:27 junk
```

שינוי בעלות על קובץ הופכת שימושית אם אתה מזיז/מעתיק קבצים כroots לשימוש על ידי משתמשים אחרים.

**הרשאות קבצים.** כעת, בעלים של קובץ יכול להפוך את הקובץ לנגיש בשלושה מצבים:

read (r), write (w) & execute (x)

לשלוש רמות של משתמשים :

owner (u), members of a group (g), others on the system (o)

אתה יכול לבדוק את הרשאות הגישה הנוכחיות על ידי שימוש ב:

```
ls -l filename
```

אם הקובץ נגיש לכל המשתמשים (owner, group, other) בכל שלושת המצבים (read, write, execute) הוא יראה:

```
-rwxrwxrwx
```

דלג על המקף "-" הראשון (הוא מראה "d" לספריות, "-" לקבצים נורמליים, "l" ללינקים, "c" לcharacter devices, "b" ל block devices, "p" ל named pipes, "f" ל stacks) לאחר תו זה, השלישייה הראשונה מראה את הרשאות הקבצים לבעלים על הקובץ, השלישייה השנייה מראה את הרשאות לקבוצה שבבעלותה הקובץ, השלישייה השלישית מראה את הרשאות למשתמשים אחרים. הרשאה שלילית נראת כ "-". לפניך פלט מפקודת `ls -l` לקובץ שבבעלות root, שבו לבעלים יש את כל הרשאות, אבל לקבוצה ולאחרים יש רק הרשאות קריאה וביצוע:

```
drwxr-xr-x 2 root root          21504 Apr 24 19:27 dev
```

האות הראשונה "d" מראה שהקובץ הוא למעשה ספרייה.

אתה יכול לשנות את הרשאות על קובץ שבבעלותך על ידי שימוש בפקודה `chmod`. לדוגמא, פקודה זו תוסיף את הרשאה לקרוא את הקובץ "junk" לכל. (=user+group+others)

```
chmod a+r junk
```

בפקודה למעלה, במקום "a" (=all), יכולתי להשתמש ב"u", "g" "o" (= user, group, others). במקום "+" (=הוספת הרשאה), יכולתי להשתמש ב"-" או ב"=" (הסר את ההרשאה או קבע את ההרשאה). במקום "r" (=read), יכולתי להשתמש ב"w" או ב"x" (=execute או write=).

דוגמא שנייה, פקודה זו תסיר את ההרשאה לבצע את הקובץ "junk" מאחרים:

```
chmod o-x junk
```

במקום אותיות, אתה יכול להשתמש גם במספרים כדי לציין את ההרשאות. על מנת להבין איך זה עובד, הסתכל בזה:

```
execute=1  
write=2  
read=4
```

סך ההרשאות למשתשים הוא סכום השלושה:

0 = שום הרשאה בכלל ( לא לכתוב, לא לקרוא, או לבצע) (שכיח)

1 = לבצע בלבד (לא נפוץ)

2 = לכתוב בלבד (לא נפוץ)

3 = לכתוב ולבצע (לא נפוץ)

4 = לקרוא בלבד (שכיח)

5 = לקרוא ולבצע (שכיח)

6 = לקרוא ולכתוב (שכיח)

7 = לקרוא, לכתוב ולבצע (שכיח)

ההרשאה לכל שלושת הרמות של המשתמשים (בעלים, קבוצה, אחרים), מושגת על ידי הדבקה של שלוש הספרות יחד, אחת אחת. לדוגמא, הפקודה:

```
chmod 770 junk
```

תיתן לבעלים ולקבוצה את מירב ההרשאות, אבל ללא הרשאות לאחרים. הפקודה:

```
chmod 666 junk
```

תיתן לכל שלושת רמות המשתמשים (בעלים, קבוצה, אחרים) את ההרשאות לקרוא ולכתוב

(אבל לא לבצע) את ומהקובץ "junk". שים לב בבקשה ל"666". זהו שילוב די נפוץ, לפחות לגבי אדם אחד שאני מכיר, זוהי ההוכחה שליניוקס (וכל יוניקס לצורך העניין) היא עבודה של השטן 0:-> .

פקודה זו:

```
chmod 411 junk
```

תיתן לבעלים את ההרשאה לקרוא בלבד, ולקבוצה ולשאר המשתמשים לבצע בלבד את הקובץ. זה לא נראה מועיל כל כך, אבל יכול להיות מצחיק, לפחות למשתמשי לינוקס מצפון אמריקה שמחייגים 411 (מספר טלפון) לעזרת מרכזנית.

הצורה המספרית להצגת הרשאות קבצים נקראת "octal" בגלל שלמספרים יש הבסיס 8 (הבסיס של המערכת הדצימלית הוא 10). הספרה הגבוהה ביותר המערכת האוקטלית הוא 7 (למערכת האוקטלית יש 8 ספרות : 0-7, כלמערכת הדצימלית יש 10 ספרות : 0-9). התצוגה באוקטלית היא סימון נח לייצוג הבינארי של הרשאות קבצים, כשכל הרשאה מסומנת ל"set" או ל"denied" עם אחד או אפס והסך הכל מיוצג כרצף של אפסים או אחדים, כמופיע בדיאגרמה הבאה:

users			
class:	owner	group	
others			
exemple permissions:	rwX	rw -	r - -
absent permissions:	- - -	- - X	- wx
binary representation of the permissions:	111	110	100
octal representation of the binary:	7	6	4

## הרשאות לספריות.

הרשאות לספריות הינן בעלות משמעות שונה מאשר הרשאות לקבצים "רגילים". לקבצים רגילים:

z=הרשאה לקרוא את תכולת הקובץ

w=הרשאה לשנות את תוכן הקובץ

x=הרשאה לבצע את הקובץ.

לספריות:

z=הרשאה להציג את רשימת הקבצים בסיפרייה

w=הרשאה ליצור או למחוק קבצים בתוך הספרייה

x=הרשאה לגשת אל הספרייה.

מכל בחינה אחרת הרשאות נקבעות לספריות ממש כמו לקבצים רגילים.

## ברירת המחדל של הרשאות קבצים עם umask.

כשקובץ חדש נוצר, ניתנות לו הרשאות ברירת מחדל. במערכת שלי הן:

- rw - r- -r - -

זה אומר שקבצים שנוצרו על ידי משתמש יכולים להיקרא ולהיכתב על ידי המשתמש הזה; הקבוצה והאחרים יכולים רק לקרוא את הקובץ. עדיין, במערכת הרדהאט שלי, כברירת מחדל, משתמשים אינם יכולים לקרוא את התוכן של ספריות הבית של משתמשים אחרים מכיון שההרשאות של ספריות הבית הן:

```
drwx - - - - -
```

אני יכול לבדוק את הרשאות ברירת המחדל לקובץ שזה עתה נוצר על ידי שימוש ב:

```
umask -S
```

(האופציה "-S" מתפרשת כ "symbolic" ואומרת ל`umask` להציג את ההרשאות באופן נוח לקריאה, המקום מצב התצוגה הדיפולטי המספרי)

אני יכול לשנות את הרשאות ברירת המחדל לקבצים החדשים שלי על ידי שימוש בפקודה כמו:

```
umask u=rw,g=,o=
```

שתיתן לבעלים את הרשאות הכתיבה והקריאה לקבצים חדשים שיווצרו (`r+w`), ושום הרשאה לקבוצה ולאחרים.

שימוש במספרים כדי לקבוע הרשאות ברירת מחדל עם `umask` הוא עניין קצת יותר מורכב. המספר מראה את ההרשאות שאתה שולל ממשתמשים (בניגוד ל `chmod`). לכן:

```
umask 000
```

ייתן הרשאות מלאות לכולם על קבצים חדשים שיווצרו. הדוגמא הבאה תיתן הרשאות כתיבה וקריאה לבעלים, ואפס הרשאות לכל השאר (וזוה כנראה מה שתרצה לעשות):

```
umask 177
```

כדי להפוך את השינויים לקבועים לכל המשתמשים במערכת, התאם את השורה (שורות) הרלוונטית בקובץ `/etc/profile`.

### 3.2.1 נגן mp3 שלי נחנק. הצליל לא נשמע טוב (איך להגדיר `suid`)

יכול להיות שלנגן mp3 אין די כח מעבד (הוא דורש הרבה). יכול להיות שהמחשב שלי פשוט גרוע. או שיכול להיות שאתה מריץ יותר מדי תוכניות שדורשות כח עיבוד רב, בבת אחת. או, מה שסביר להניח שקורה, אתה צריך להריץ את הנגן עם עדיפות גבוהה יותר. (העדיפות של תוכנית יכולה להיקבע בעזת הפקודה `nice`--בדוק את `man nice` או את `info man`). נסה להפעיל את הנגן `root`--לתוכניות שמופעלות על די `root` ניתנת עדיפות גבוהה יותר מאשר לאלו שמופעלות על ידי משתמשים רגילים. אם זה פותר את הבעיה, קבע את ה "`suid`" לכל התוכניות ככה שלכל המשתמשים תינתן "זהות משתמש אפקיבית" לגבי הבעלים של הקובץ (בדרך כלל `root`) כמשפעילים אותן, לדוגמא:

```
chmod a+s /usr/bin/xmms
```

ייעשה את הטריק לתוכנית xmms. הפלט של:

```
ls -l /usr/bin/xmms
```

במחשב שלי הוא כעת:

```
-rwsr-sr-x 1 root root 908k Feb 22 2000 /usr/bin/xmms
```

ה "s" הראשונה מצביעה על כך שה `suid=substitute-user-id` קבוע-מוגדר. ה "s" השנייה מצביעה על כך ש ה `sgid=substitutue- group-id` גם כן קבוע-מוגדר. לכן לכל מי שמבצע את xmms ניתנת הזכות האפקטיבית של הבעלים של התוכנה וזכות הקבוצה של הקבוצה שבבעלותה התוכנית, אשר בדוגמה למעלה שניהם `root`.

הגדרת `suid` לתוכנית יכולה להוות פירצת אבטחה במערכת שלך. זהו אינו המקרה ברשת ביתית סגורה וכשמגדירים `suid` לתוכנית שהמקור שלה ניתן למעקב. בכל מקרה, אפילו בבית, לא הייתי מגדיר `suid` לחתיכת קוד שהמקור שלו אינו וודאי, אפילו שהוראות ההתקנה האיצו בי לנהוג כך. בנוסף, זה בהחלט רעיון רע מאוד להגדיר `suid` לתוכניות רבות המערכת שלך--זה פוגע בכל הרעיון שעומד מאחורי האבטחה של יוניקס.

בכל מקרה, ישנן כמה תוכניות שדושרות `suid` לתיפקוד תקין, `kppp` (חיבור המודם הפופולרי תחת `kde`) למשל. זה נובע מהצורך בגישה ישירה לחומרה--לפעמים רק ל`root` מותר לעשות את זה.

אם אתה נתקל בבעיות חוזרות לגבי תיפקוד "חלק" של המערכת שלך, או שחלק מהחומרה שלך (צורב למשל) נוטה לקרוס, נסה להפחית את מספר ה`daemons` במערכת הלינוקס שלך. הרץ `roots` את הפקודה `setup` (פקודה של `RH`) ובטל את כל ה"שירותים" (`services`) שאתה לא באמת צריך. לחילופין, אתה יכול לעבור לשורת פקודה, לכבות את הממשק הגרפי (הפקודה `3 roots init`), מה שיגרום לביצועי המערכת שלך להשתפר ללא ספק.

לאילו שצריכים (אוהבים) מערכת הפעלה "אוניברסלית" (תחנת עבודה, שרת, מחשב משרדי, קופסת משחקים, מולטימדיה, וכו', והכל בעת ובעונה אחת), ישנם טלאים מיוחדים לקרנל של לינוקס: "low latency patch" & "pre-emptive kernel patch" שתוקפים באופן אגרסיבי את בעיית ה"latency" (זמן איחזור) שמערכות עמוסות מדי מציגות.

הבא: תיזמון משימות בעזרת "&", "at", "batch" ו-"cron"

### 3.3 תזמון משימות עם "&", "at", "batch", ו-"cron"

#### 3.3.1 כיצד אני מבצע פקודה "ברקע"?

על ידי שימוש ב"&" בסוף הפקודה. למשל, זה יתחיל את licq (קליינט של icq) בטרמינל של x ברקע, ככה שלאחר החלת הפקודה, ה x טרמינל שלי אינו חסום:

```
licq &
```

מספר הזיהוי של התהליך, job\_number, מודפס על המסך, כך שאתה יכול להשתמש בו עם פקודות שקשורות אליו. הפקודות הן:  
fg job\_number (= "foreground", הבא את התהליך שרץ ברקע חזרה אל התצוגה/שליטה המיידית שלי, התחל אותו מחדש אם הוא נעצר).  
bg job number (= "background", שלח את התהליך לרקע, התחל אותו מחדש אם הוא נעצר, בדיוק כאילו הוא הותחל על ידי שימוש ב"&").  
z <Ctrl> (שלח את התהליך שרץ כרגע בחזית לרקע ועצור אותו).  
Jobs (הצג רשימה של עבודות פעילות).  
Kill process\_ID (השמד את התהליך, השתמש בפקודה ps כדי למצוא את המזהה של התהליך אותו מעוניין להשמיד).

כדי לגרום לתהליך שרץ ברקע להמשיך לרוץ לאחר שאתה מתנתק, אתה יכול להשתמש בפקודה nohup (= no hangup) למשל:

```
nohups make &
```

#### 3.3.2 איך אני מבצע פקודה בזמן מסויים (על ידי שימוש ב "at" או "batch")?

הפקודה "at" תבצע פקודה (דות) שאתה מציין בזמן ותאריך לפי בחירתך. למשל, אני יכול להתחיל לנגן מוזיקה מכונן התקליטורים שלי בשעה 7 בבוקר:

```
at 7:00
```

```
cdplay<Ctrl>d
```

בדוגמא למעלה, הכנסתי את השורה "at 7:00" בשורת הפקודה ואז לחצתי על ENTER. לזה, הפקודה at הגיבה בהצגת פרומפט, ">at". שם הקלדתי את הפקודה "cdplay" ואז לחצתי על Ctrl d, לחצתי ENTER, ה">at" הבא הופיע, שבו הייתי יכול להכניס את הפקודה הבאה שתבוצע מייד לאחר "cdplay", גם כן ב 7:00.  
וכן הלאה, יכולתי לתזמן כך הרבה פקודות אחת אחרי השנייה. לאחר שהקלדתי את הפקודה האחרונה, סיימתי את הקלט שלי עם Ctrl d. חשוב על Ctrl d כמה ששולח "סוף החיים" לקלט הנוכחי. אל תלחץ על Ctrl d פעמיים מכיון שזה יוציא אותך החוצה--זה מה Ctrl d עושה כשמשתמשים בו ישירות בשורת הפקודה של לינוקס.



אתה יכול להציג את העבודות שתיזמנת לביצוע על ידי שימוש ב:

at -l

שייתן לך רשימה ממוספרת של עבודות שמחכות.

במידה ושינת את דעתך אתה יכול להסיר עבודה מהרשימה, לדוגמא:

atrm 8

תסיר את העבודה עם המספר 8 ברשימה.

יכולתי גם לתזמן עבודה לביצוע בשלב מאוחר יותר, לדוגמא:

at 23:55 12/31/00

startx

תתחיל את מערכת ה x-windows שלי בדיוק בזמן למילניום החדש (5 דקות לפני חצות ב31 בדצמבר 2000).

אם אינך יכול לבצע את הפקודה at, בדוק אם השירות (daemon) "atd" טעון (כroots, השתמש ב ntsysv). אם אינך יכול לבצע את הפקודה at כמשתמש רגיל למרות שהיא עובדת עם root, בדוק אם הקובץ הריק /etc/at.deny קיים ושאינ כל קובץ בשם /etc/at.allow. זו אמורה להיות הגדרת ברירת המחדל והיא מאשרת לכל המשתמשים לבצע את הפקודה at. אם אתה מעוניין שמתמשים מסויימים בלבד יוכלו להשתמש בat, צור קובץ בשם /etc/at.allow ומנה את המשתמשים הללו שם.

לאופציות אחרות, בדוק את:

man at

אם אתה מבקש לבצע עבודה שדורשת כח עיבוד רב ברקע בזמן שהעומס על המערכת נמוך, אתה יכול לבחור בפקודה batch. אני יכול, לדוגמא, להריץ את setiathome (תוכנית שמעבדת נתונים בכדי לעזור בחיפוש אחר אינטליגנציה חוצונית, SETI) על ידי שימוש ב:

batch

at>setiathome<Ctrl>d

בדוגמא זו, הכנסתי את הפקודה batch ולאחר מכן את הפרומפט של "at>", הכנסתי את הפקודה שרציתי שתבוצע ברקע. העבודה מנסה להתחיל מייד, אבל מתקדמת רק כשהעומס נמצא מתחת ל 0.8 אתה יכול לבדוק את העומס של המערכת על ידי בחינת התוכן של הקובץ (הוירטואלי) /proc/laodavg. למשל:

```
cat /proc/loadavg
```

כשקובץ batch מסיים, הפלט נשלח אלי דרך דואר אלקטרוני.

### 3.3.3 איך אני מגדיר את cron?

cron (תהליך של לינוקס שמבצע עבודה ברקע, בדרך כלל בלילה) מוגדר כברירת מחדל במערכת redhat שלך. כך שאינך צריך לעשות דבר לגביו אלא אם אתה מעוניין להוסיף משימות שיבוצעו במערכת שלך על בסיס קבוע או לשנות את הזמן בו cron מבצע את החובות המוטלות עליו.

שים לב בבקשה שייטכן וחלק מהעבודה ש-cron מבצע יכולה להיות חיונית לתיפקוד התקין של המערכת שלך לאורך זמן. בנוסף לדברים אחרים cron יכול:

- לבנות מחדש את קבצי מסד הנתונים שנמצאים בשימוש בזמן שאתה מחפש קבצים על ידי שימוש בפקודה locate.
- לנקות את הספרייה /tmp.
- לבנות מחדש את דפי ההסברים של לינוקס (manual pages).
- לעשות רוטציה לקבצי הרישום (log files), להיפטר מקבצי הרישום הישנים ביותר, לשנות את שמם של קבצי הרישום, וליצור קבצי רישום חדשים.
- לבצע בדיקות נוספות אחרות, כמו הוספת פונטים שהעתקת לאחרונה אל המערכת.

לכן, כיבוי מכונת הלינוקס שלך בלילה לא יהיה הרעיון המוצלח ביותר-- במקרה כזה ל-cron לא תהיה ההזדמנות לעשות את עבודתו. במידה ואתה מעוניין לכבות את המחשב שלך ללילה, יכול להיות שתרצה לכוון את cron ככה שהוא יבצע את החובות המוטלות עליו בזמן אחר.

כדי לבדוק מתי cron מתעורר כדי לבצע את עבודתו, הסתכל בקובץ /etc/crontab, לדוגמא:

```
cat /etc/crontab
```

ייטכן והוא יכיל משהו כזה:

```
# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * * root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.mothly
```

אתה יכול לראות שישנן ארבע קטגוריות של עבודות cron: שמבוצעות מדי שעה, מדי יום, שבוע, וחודש. אתה יכול לשנות אותן או להוסיף קטגוריה משלך. ככה זה עובד:

העמודות מצביעות על: דקות (0-59), שעות (0-23), היום בחודש (1-31), החודש בשנה (1-12), היום בשבוע (0-6 ראשון עד שבת). ה"\*\*\*" מתפרש כ"כל ערך אפשרי".

לכן, בדוגמא המצוטטת, העבודות שמכוונות לביצוע כל שעה, מבוצעות בכל פעם שהשעון של המחשב מראה "ודקה אחת", מה שמתרחש בכל שעה, דקה אחת לאחר השעה הנתונה. העבודות היומיות מבוצעות בכל פעם שהשעון מראה 2 דקות לאחר השעה 4, מה שקורה

פעם ביום. העבודות השבועיות מבוצעות 22 דקות לאחר השעה 4 בבוקר בימי ראשון. העבודות החודשיות מבוצעות 42 דקות לאחר השעה 4 ביום הראשון של כל חודש. הספרייה עם קובץ הסקריפט שמכיל את הפקודה (פקודות) שאמורות להתבצע, נראה ברשומה האחרונה בכל שורה.

אם אתה רוצה שהעבודות שלך יבוצעו בצהריים במקום ב4 בבוקר, שנה את ה4 ל12. cron מתעורר בכל דקה ובודק אם הקובץ /etc/crontab שונה ככה שאין צורך להפעיל מחדש משהו לאחר שביצעת את השינויים שלך.

אם אתה רוצה להוסיף עבודה אל הcron שלך, הנח סקריפט שמריץ את העבודה המבוקשת (או לינק אל הסקריפט שלך) בספרייה /etc/cron.hourly או /etc/cron.weekly או /etc/cron.monthly.

לפניך דוגמא לרשומה ב /etc/crontab שגורמת לעבודה להתבצע שלוש פעמים בשבוע (שני, רביעי, שישי):

```
02 4 * * 1, 3, 5 root run-parts /etc/cron.weekly
```

דוגמא שנראתה ב usenet אך לשלוח באופן אוטומטי במייל קובץ רישום:

Re: help in crontab

From: Dean Thompson <Dean.Thompson@csse.monash.edu.au> Date:2001-03-03 16:35

newsgroups: comp.os.linux.admin.networking,comp.os.linux.security

>How can I set the job mail abc@abc.com < /var/log

>every day in the /etc/crontab -e file?

You could try the following entry and see if you meet with any success:

```
0 0 * * * (/bin/mail abc@abc.com < /var/log/messages) > /dev/null 2>&1
```

shell : הבא

## shell 3.4

### 3.4.1 מה זה מעטפת (shell) והאם אני רוצה להשתמש באחרת?

מעטפת היא התוכנית שמפענחת את מה שאתה מקליד אל תוך שורת הפקודה ומחליטה מה לעשות עם זה. ניתן לעורר מעטפת גם בצורה שאינה אינטרקטיבית, למשל לבצע רשימה של פקודות שהוקלדו מראש והוכנסו אל תוך קובץ טקסט ("shell script" a). אתה יכול לחשוב על מעטפת כמקבילה של "command.com" מדוס (מפענח שורת הפקודה) ועל shell script כמקביל של קבצי batch בדוס (\*.bat). בהשוואה לדודנים שלהם מדוס, המעטפת והסקריפטים נמצאים על סטרואידיים.

ישנן כמה מעטפות זמינות בלינוקס (במידה והתקנת אותן):

bash : ("bourne again" shell) sh, bourne shell, הסטנדרט במערכות יוניקס רבות),  
csh (C shell, בעל תחביר דומה ל"c" שפת התיכנות, זמין ברוב מערכות היוניקס).  
pdksh (public domain korn shell), tcsch (tiny C shell, נמצא בשימוש במערכות קטנות  
בדרך כלל) sash (stand alone shell, ניתן לשימוש כשספירות אינן זמינות).  
zsh, ash, וכנראה עוד כמה.

מעטפת ברירת המחדל במערכת שלי (וסביר להניח שגם בשלך) היא bash, שהינה מעטפת מצויינת וסטנדרטית, ואני לא מוצא סיבה משכנעת למה ניוביי כמוני ירצה לשנות את זה. bash תואמת לאחור באופן מלא עם bourne shell (המעטפת הכי פופלרית ביוניקס) ומכילה הרבה שיפורים ותכונות טובות ממעטפות אחרות. מנקודת מבט של משתמש מתחיל, המעטפות השונות כלולות בלינוקס מסיבות היסטוריות ובגלל תאימות לאחור של סקריפטים שיידרשו מעטפת מסויימת בכדי לרוץ. [ ישנן מעטפות שיועילו אם אתה כותב תוכניות שמכוונות להתקנים מיוחדים ו"מוטמעים" (embedded), שייתכן ויריצו מעטפת קטנה (a" tiny" shell)].

אתה יכול לבדוק איזו מעטפת אתה מריץ על ידי שימוש ב:

```
echo $SHELL
```

אם אתה מעוניין לנסות מעטפת אחרת, הקלד, לדוגמא:

```
tcsch
```

מה שיגרום למעטפת c קטנה (tiny c shell) להתחיל לפעול. כשסיימת הקלד:

```
exit
```

שיחזיר אותך למעטפת הקודמת (שימוש בexit במעטפת הראשונה שלך יוציא אותך אל מחוץ למערכת-log out). אתה יכול לבדוק כמה מעטפות ערמת אחת על השניה על ידי

הצגת המשתנה הסביבתי של רמת המעטפת (shell level):

```
echo $SHLVL
```

בפקודה למעלה, ה\$ מתפרש כ"הרחב את הערך של משתנה המערכת הסביבתי",  
"SHLVL" הוא שם המשתנה, ו"echo" היא פקודה שמדפיסה דברים.

המעטפת לכל משתמש מצויינת בשדה האחרון בקובץ הסיסמאות /etc/passwd.  
אם אתה באמת רוצה לשנות אותה, ערוך (root) את הקובץ הזה והחלף את ה"/bin/bash"  
במעטפת שבחרת.

### 3.4.2 כיצד אני מבצע התאמה אישית למעטפת שלי?

במכונה שלי שורת הפקודה נראית כך:

```
[stan@marie stan] $ _
```

כאן stan הוא השם איתו אני נכנס למערכת (log in), marie הוא השם של המחשב,  
stann השני הוא השם של ספריית העבודה הנוכחית שלי, וה\_ מייצג את הסמן.

שורת הפקודה (prompt) מוגדרת על ידי משתנה סביבתי שנקרא PS1. כדי להציג את  
ההגדרות הנוכחיות שלי, אני יכול להשתמש ב:

```
echo $PS1
```

ההגדרות הכלל מערכתיות של הפרומפט (לכל המשתמשים במערכת) נמצאות בקובץ  
/etc/bashrc שבמערכת שלי מכיל שורה כזו:

```
PS1="[u@h W] \$ "
```

כדי לבצע התאמה אישית לפרומפט, אני יכול לערוך את הקובץ /etc/bashrc (כ root)  
ולהכניס כמעט כל טקסט בתוך סימני הציטוט ("). הנה המשמעות של כמה קודים מיוחדים  
שאני יכול לבחור לשימוש:

- \u שם המשתמש של המשתמש הנוכחי (\$LOGNAME=),
- \h שם המחשב שמריץ את המעטפת (שם המארח),
- \H שמו המלא של המארח,
- \W הבסיס של השם של ספריית העבודה הנוכחית,
- \w השם המלא של ספריית העבודה הנוכחית,
- \\$ מציג "\$" למשתמשים רגילים ו"#" לroot,
- ! מספר ההיסטוריה של הפקודה הנוכחית,
- # מספר הפקודה הנוכחית (כפי שמבוצעת במעטפת הנוכחית),
- \d התאריך הנוכחי,
- \t השעה הנוכחית (24 שעות),
- \T השעה הנוכחית (12 שעות) - bahe 2.0 בלבד,
- \@ השעה הנוכחית (am-pm) - bahe 2.0 בלבד,
- \s שם המעטפת,

\a - אזעקת צליל (ביפ),  
 \j - מספר העבודות שיש למשתמש,  
 \n - שורה חדשה,  
 \\ - בקסלאש,  
 \[ - התחל רצף של תוים לא ניתנים להדפסה,  
 \] - סיים רצף של תוים לא ניתנים להדפסה,  
 \ch - תו ה ASCII שמתייחס אל המספר האוקטלי חחח,  
 \$(date) - פלט מהפקודה date (או כל פקודה אחרת לצורך העניין),

לפניך דוגמא שמראה איך להוסיף צבע. הסתכל בפרק הבא לגבי פרטים על צבע:

```
PS=1" \ [033[1;32m\] [\u@\h \W] \$\ [033[0m\] "
```

ישנה גם רמה שניה של פרומפט, היא נקבעת על ידי המשתנה שנקרא PS2. המעטפת משתמשת ברמה שניה של פרומפט כאשר היא מחכה לקלט נוסף. במערכת שלי הפרומפט המשני הוא ">". אני לא דואג יותר מדי לגבי PS2, אבל אם יכולתי הייתי מגדיר אותו כמו PS1. ישנם גם PS3 ו-PS4 אבל הם נראים לעיתים נדירות.

### 3.4.3 צבע בטרימינל טקסט

ניתן להפיק צבע בטרימינל טקסט על ידי שימוש ב "ANSI escape sequences". לדוגמא:

```
echo -e "\033[44;37;5m ME \033[0m COOL"
```

הנ"ל מגדיר את הרקע ככחול, את החזית כלבן, מסך מהבהב, ומדפיס ME, לאחר מכן מאפס את הטרימינל חזרה להגדרות ברירת המחדל שלו ומדפיס COOL. ה -e הוא אופציה ספציפית לפקודה echo--היא מאפשרת את הפיענוח של התוים המיוחדים. ה"033" מייצג את escape sequences. ה m קובעת אטריביוט ולכן מסיימת את הרצף. הקודים הממשיים בדוגמא למעלה הם "44;37;5" ו-"0".

שנה את ה "44;37;5" כדי ליצור שילובי צבע אחרים--המספר/סדר של הקודים אינו משפיע. הקודים הניתנים לבחירה רשומים למטה:

0	אפס את כל האטריביוטים להגדרות ברירת המחדל שלהם
1	הגדר bold (bold bit oriented line discipline) (א.א)
2	הגדר חצי-בהיר (בתצוגת צבע על צבע)
4	הגדר קו תחתון (בתצוגת צבע על צבע)
5	הגדר היבהוב
7	הגדר וידאו הפוך
22	הגדר בהירות נורמלית
24	כבה קו תחתון (underline)
25	כבה היבהוב
27	כבה וידאו הפוך
30	הגדר חזית שחורה
31	הגדר חזית אדומה

32	הגדר חזית ירוקה
33	הגדר חזית חומה
34	הגדר חזית כחולה
35	הגדר חזית לאדום ארגמן
36	הגדר חזית לחן (בין כחול לירוק)
37	הגדר חזית לבנה
38	הפעל קו תחתון (underscore)
39	כבה קו תחתון (underscore)
40	הגדר רקע שחור
41	הגדר רקע אדום
42	הגדר רקע ירוק
43	הגדר רקע חום
44	הגדר רקע כחול
45	הגדר רקע לאדום ארגמן
46	הגדר רקע לחן
47	הגדר רקע לבן
49	הגדר רקע ברירת מחדל

קודים מענינים אחרים:

\033 [2j	נקה מסך
\033 [0q	נקה את כל נוריות הבקרה שעל המקלדת (לא יעבוד מטרמינל X)
\033 [1q	קבע את ה"scroll lock led"
\033 [2q	קבע את ה"num lock led"
\033 [3q	קבע את ה"caps lock led"
\033 [15;40H	הזז את הסמן לשורה 15, עמודה 40
\007	פעמון (ביפי)

LED's (=light emitting diodes) הן הנוריות שעל המקלדת שמסמנות את <capslock>, <numlock> ו- <scrolllock> מופעלים.

הסתכל ב man console\_codes לעוד.

#### 3.4.4 כיצד אני מדפיס סימנים בקונסול, באפליקציות מצב טקסט, וב-X?

הפרוצדורה שמתוארת כאן יכולה לתת לי גישה מהירה אל סט התווים המורחב של ה־cs שלי (קודים 128-255) והיא די נייחת בעולם ה־cs: היא עובדת במייקרוסופט ווינדוס, דוס (אם יש לך מנהל התקן ANSI מותקן), ובתוך כל אפליקציות מצב טקסט בלינוקס (כולל מעטפת שורת הפקודה). מצאתי שזה שווה את הזמן אותו הקדשתי לשינון הקודים לתווים המעטים בהם אני נוטה להשתמש הכי הרבה.

זה עובד כך. וודא ש- <numlock> פועל. לאחר מכן לחץ על <Alt> והחזק אותו לחוץ. בזמן ש- <Alt> לחוץ, הקלד את 4 הספרות האלו בפד הספרות המרובע שבצד ימין על

המקלדת: 0181. כעת שחרר את <Alt> והאות היונית mu "µ" תופיע. מצאתי כמועילים את התווים הבאים: ° 176 (מעלות), ± 177 (פלוס מינוס), ² 178 (בריבוע), ³ 179 (בשלישית), µ 181 (mu יוני), · 0183 (נקודת הכפל), è 232 ( האקסנט הצרפתי agrave ), é 233 ( האקסנט הצרפתי aigu ), ä 228 (ה a-umalaut הגרמני), ó 243 (ה u-zamknieten הפולני), ø 248 (ה o-barn הסקנדינבי) ü 252 (ה u-umalaut הגרמני).

כעת, אם אני באמת רוצה, אני יכול להיות הבעלים של קובץ בשם  $\mu\text{m}\cdot\text{c}\pm\text{b}^3$ . מייקרוסופט ווינדוס, ANSI של דוס, ויוניקוד (unicode) שונים במקצת מהתווים למעלה, אבל ה"ליבה" השימושית נשארת זהה. הסתכל ב: <http://www.hclrss.demon.co.uk/demos/ansi.html> אם אתה מעוניין לדעת את פרטי ההבדלים. לינוקס משתמשת בסטנדרט של יוניקוד.

תחת X, שילובי המפתח שלמעלה לא יעבדו. אבל אני יכול להשתמש ב:

kchselect&

או ב:

gchormap&

כדי לבחור תו יוניקוד ולהעתיק אותו אל האפליקציה שלי. לא כל תו יוניקוד זמינים עדיין, אבל רבים כן זמינים. מעמודי יוניקוד אחרים מאשר עמוד 0, ייתכן והתווים יוצגו או שלא, תלוי באפליקציה שלך ובזמינות של glyph בפונט שלך. לדוגמא, אני יכול בוודאות להשתמש בתווים הבאים ברוב האפליקציות שב KDE (אם הם מוצגים בדפדפן שלך, תלוי בדפדפן שלך ובזמינות של פונט יוניקוד מתאים):

יונית (עמוד יוניקוד 3, תווים 913-969)

רוסית (עמוד יוניקוד 4, תווים 1040-1103)

ורבים אחרים. אתה יכול למצוא תו יוניקוד נפוצים (מספריים) ואת ייחוס הhtml הסימבולי שלהם ("character entity") ב:

[http://www.hclrss.demon.co.uk/demos/ent4\\_frame.html](http://www.hclrss.demon.co.uk/demos/ent4_frame.html)

### 3.4.5 איך אני כותב סקריפט מעטפת פשוט?

צור קובץ טקסט (ASCII) שיכיל את סקריפט המעטפת (shell script). לדוגמא, אני אשתמש בעורך pico כדי לכתוב סקריפט שיריץ את התוכנית tar עם כל הפרמטרים שבדרך כלל דרושים כדי לפרוס קבצי tarball שאני מוריד מהאינטרנט ( אני אף פעם לא מצליח לזכור את האופציות של tar ). החלטתי לקרוא לסקריפט שלי "untar":

pico untar



מכיון שהקובץ "untar" לא קיים בספרייה הנוכחית שלי, הוא נוצר על ידי עורך הטקסט pico.  
כעת, אני מקליד את התוכן של הסקריפט שלי:

```
#!/bin/bash
echo this is the script file $0
echo untarring the file $1
# this calls tar with options -xvzf (extarct,
# verbose, filter through gzip, input filename)
tar -xvzf $1
```

אני שומר את הקובץ עם `<Ctrl>o` ויוצא עם `<Ctrl>x`

השורה הראשונה של הסקריפט, מתחילה ב"#!" (נקרא pound-bang), הינה מיוחדת--היא אומרת למעטפת באיזו תוכנית להשתמש כדי לפענח את הסקריפט שלי. בדוגמא זו, הסקריפט ייפוענח על ידי המעטפת `bash`: `/bin/bash`. השורה הראשונה חייבת להתחיל ב"#!" או שהסקריפט לעולם לא ירוץ (הקובץ ייפוענח כקובץ טקסט רגיל). שורות אחרות שמתחילות ב"#" הינן הערות למחבר (קוראים, משתמשים) המעטפת והן זוכות להתעלמות מוחלטת מצד המחשב.

ה `$0`, `$1`, `$2` ... בסקריפט שלי הם הפרמטרים שהועברו אל הסקריפט שלי. לדוגמא, אם אני מריץ סקריפט בשם "myscript" עם שבע פרמטרים כמו זה:

```
myscript a b c d e f g
```

אז `$0` יהיה בתוך "myscript" כבעל הערך "myscript", ל `$1` יהיה הערך "a", `$2` יהיה "b", `$3` יהיה "c", וכן הלאה.

בשורה השניה ושלישית בסקריפט הדוגמא שלי, הפקודה `echo` מדפיסה על המסך כל מה שעוקב אחרי אותה שורה, מרחיבה את `$0` ו-`$1` לערכי הפרמטרים שהועברו לסקריפט. השורות הרביעית והחמישית מכילות הערה שכתבתי לעצמי כדי להזכיר לעצמי מה ניסיתי להשיג, רק למקרה שאי פעם אצטרך לשנות את הסקריפט שלי. השורה האחרונה מבצעת את העבודה האמיתית.

ברגע שהסקריפט נכתב, אני הופך את הקובץ לבר הרצה לבעלים של הקובץ (`"u"=user`):

```
chmod u+x untar
```

והסקריפט שלי מוכן להרצה באופן הבא:

```
./untar my_tar.tar.gz
```

הסקריפטים של לינוקס בהחלט עשירים, גמישים, עוצמתיים ויכולים להיות מורכבים. בכל מקרה, אין צורך בידע מיוחד כדי לכתוב סקריפטים פשוטים לאוטומציה של משימות פשוטות. אתה רק צריך לאסוף ביחד קבוצה של פקודות שנמצאות בשימוש שכיח, אחת אחרי השניה, אל תוך קובץ. אני משתמש בסקריפטים בגלל שאני עצל מדי כדי להקליד את אותן קבוצות של פקודות שוב ושוב.

רצף פקודות פשוט באמת יכול להיות מוקלד אל תוך קובץ טקסט ולהיות מועבר אל תוך מעטפת לביצוע מידי על ידי שימוש ב:

```
source my_file
```

[אין צורך ב"pound bang" הראשוניים או בהרשאות הרצה.]

### 3.4.6 המשמעות של ציטוטים

בדרך כלל סימנים אלו מיוחדים במשמעותם לגבי המעטפת:

```
\ ' " ` < > [ ] ? | ; # $ ^ & * ( ) = <Space> <Tab> <Newline>
```

ישנם ארבעה סוגים שונים של ציטוטים: קו נטוי אחורה (\), מרכאות ('), מרכאות כפולות ("), מרכאות הפוכות (').

הקו הנטוי אחורה \ מתפרש כ: בטל את המשמעות המיוחדת של התו שבא אחרי.

ציטוט עם ' ' (שתי מרכאות בודדות, לא כפולות) מתפרש כ: צטט בדיוק, בטל כל תו מיוחד בתוך המרכאות.

ציטוט עם " " (שתי מרכאות כפולות) מתפרש כ: בטל את התוים המיוחדים בתוך המרכאות פרט ל \$ ' \

זוג ה " " (שתי מרכאות הפוכות) מתפרש כ: בצע תחליף לפקודה בתוך המרכאות תחילה. כך שמה שבתוך המרכאות הפוכות מבוצע על ידי המעטפת תחילה, ולאחר מכן הפלט מועבר אל הפקודה מחוץ למרכאות. אותו הדבר ניתן ליישום בעזרת \$ (command).

דוגמאות. אני יכול ליצור ספרייה מצחיקה שנקרת "\*" על ידי \ ציטוט או " ציטוט:

```
mkdir \ *  
mkdir ' * '
```

זה מחביא את המשמעות המיוחדת של ה "\*" מהמעטפת (ללא הציטוט זה ייתפרש כ"כל הקבצים בתוך הספרייה הנוכחית").

### 3.4.7 הפניה מחודשת של פלט/קלט

ישנם שלושה זרמים חשובים של פלט/קלט: קלט סטנדרטי ("standard input-"stdin), פלט סטנדרטי ("standard output-"stdout), ופלט שגיאה סטנדרטי ("standard error output-"stderr). הם כולם ברירת מחדל לקונסול ("קונסול"=המקלדת לקלט והמסך לפלט), אבל ניתן להפנות אותם מחדש.

בכדי להפנות מחדש פלט סטנדרטי אני משתמש ב ">". לדוגמא:

```
dir my_dir > filelisting.txt
```

יפנה את הפלט הסטנדרטי של הפקודה dir אל תוך קובץ הטקסט filelisting.txt ודבר לא אמור להופיע על גבי המסך שלי. ניתן לערוך את הקובץ (בעזרת pico filelisting.txt) או להטמיע אותו בתוך מסמך.

בכדי להפנות את השגיאה הסטנדרטית, אני צריך להשתמש במבנה ">2". לדוגמא:

```
dir my_dir 2> errorlisting.txt
```

הפקודה שלמעלה תשלח פלט נורמלי אל המסך שלי ושום דבר אל הקובץ אלא אם כן dir מייצר שגיאה. כשתתרחש שגיאה, דבר לא יופיע על המסך שלי, והקובץ errorlisting.txt יכיל את הודעת הגיאה, שיכולה להיות משהו כמו:

```
dir my_dir > file_and_error_listing.txt 2>&1
```

מה שיפנה תחילה את הפלט הסטנדרטי אל קובץ טקסט, ולאחר מכן תפנה את השגיאה הסטנדרטית אל מיקום זהה למיקום הפלט הסטנדרטי. קצת מעוות, אך שזה עובד, אבל זה עובד.

בדוגמאות למעלה, אם הקובץ (שאליו מתבצעת ההפניה) כבר קיים, האינפורמציה תיכתב עליו. כדי להוסיף לקובץ קיים, אני משתמש ב ">>" כמו בדוגמאות אלו:

```
dir my_dir >> filelisting.txt
dir my_dir 2>> errorlisting.txt
dir my_dir >> file_and_error_listing.txt 2>&1
```

אם תהיה לגבי הסימן ">2", כאן, בזריזות, רציונליזציה של העניין. לזרמים הסטנדרטיים יש מתארים (descriptors) סטנדרטיים. "0" הוא קלט סטנדרטי, "1" קלט סטנדרטי ו"2" שגיאה סטנדרטית.

```
dir my_dir > file.txt
```

זה קיצור ל

```
dir my_dir 1> file.txt
```

ולכן הדוגמא למטה מפנה את השגיאה הסטנדרטית:

```
dir my_dir 2> file.txt
```

ניתן גם להשתמש בסימן "ן" כדי לשלוח ("pipe") את הפלט מפקודה אחת כקלט לפקודה אחרת. בדוגמא פופלרית זו, הפלט מ dir נשלח ("piped") אל more (more=פקודה שעוצרת את התצוגה לאחר כל מסך מלא):

```
dir | more
```

ניתן גם לפצל את הפלט ככה שהוא יילך גם אל הקובץ וגם אל המסך על ידי שימוש ב"tee":

```
dir | tee filelisting.txt
```

הפקודה נקראת "tee" כאנלוגיה לאות-צורה-T שבה אינסטלטורים משתמשים, שמחלקת את הזרימה.

חלק זה עסק עד כה בהפניית פלט סטנדרטי. הפניית קלט סטנדרטי אינה מועילה כמו הפניית הפלט, אבל ניתנת לביצוע על ידי מבנה כזה:

```
cat < my_file
```

ישנו גם משהו שנקרא הפנייה בתוך שורה (in-line redirection) של הפלט הסטנדרטי, שמיישם בעזרת "<<". שכח מזה, נראה לי כחסר כל שימוש. בכל זאת, הנה דוגמא למקרה שאי פעם באמת תזדקק לזה (כאן, ה">" מתפרש כפרומפט משני):

```
cat << my_marker  
> my_line_from_the_keyboard  
> another_line_from_the_keyboard  
> my_marker  
[הסמן שבו אני בוחר מסיים את הפניית התוך שורה]
```

פרט להפנייה אל קבצים רגילים ו"פילטרים" (כמו שמראה הדוגמא למעלה), ניתן להפנות מאל התקנים וקבצים מיוחדים אחרים. כמה דוגמאות.

דוגמא להפנייה אל קובץ התקן. הדוגמא הבאה מציגה רשימה של קבצים בטרמינל הטקסט הרביעי:

```
dir > /dev/tty4
```

דוגמא להפנייה לקובץ "FIFO" מיוחד. פקודה זו שולחת את ההודעה "you are lucky" אל משתמש ה ICQ בר המזל UIN 77777777 (בהנחה שאתה מחובר אל שרת icq עם תוכנת licq שלך):

```
echo message 77777777 "you are lucky" > ~/.licq/licq_fifo
```

הנ"ל עובד מכיון שהקובץ "licq\_fifo" בספריית licq שלך הוא קובץ תור (queue) "fifo" (first-in-first-out) מיוחד. כיצד הפקודה למעלה יכולה להיות שימושית יותר משליחת הודעה על ידי שימוש בממשק הגרפי היפה של licq? לדוגמא, אתה יכול לכתוב סקריפט קצר כדי להרשים משתמשי icq עמיתים בעזרת הודעות מרובות (זהות):

```
#!/bin/bash  
echo Messaging UIN: $1 Message $2 Times: $3  
# The next command puts your licq in the status  
"on-line, invisible".  
echo 'status *online' > ~/.licq/licq_fifo  
c=0  
while [ $c -le $3]  
do  
echo message $1 $2 > ~/.licq/licq_fifo  
c='expr $c + 1'  
echo $c " "
```

```
done
echo 'status offline' > ~/.licq/licq_fifo
echo "all done"
```

הדוגמא שלמעלה נותנת לך רעיון לגבי איך ניתן להשתמש ב־licq לאוטומציה, בזכות מודל התקשורת החכם של licq (עם קובץ fifon) והפניית קובץ פשוטה.

### 3.4.8, תוי מעטפת מיוחדים (metacharacters)

בדרך כלל לתוים אלו יש משמעות ציודת כלפי המעטפת:

= \* & ^ \$ # ? [ ] ( ) <newline> <tab> <space> ; | < > " ' \

הנה המשמעות של כמה מהם:

\ " ' משמשים לציטוט.

< > משמשים להפנייה מחדש של קלט/פלט.

| מעביר (pipes) את הפלט של הפקודה שמשאל לסימן pipen " | " אל הפקודה שמימין לסימן pipen.

; מפריד פקודות מרובות שנכתבות בשורה אחת.

<space> -ו <tab> מפרידות את מילות הפקודה.

<newline> משלים את הפקודה או סט של פקודות.

( ) סוגר פקודה או פקודות כדי שישוגרו במעטפת נפרדת (subshell) למשל ( dir ) .

{ } סוגר קבוצה של פקודות כדי שישוגרו על ידי המעטפת הנוכחית. [ dir ] . הרווח דרוש.

& גורם לפקודה הקודמת להתבצע ברקע (באופן אסינכרוני, כשהליך נפרד ועצמאי) ככה שהפקודה הבאה לא תחכה לסיימה.

\* כששם קובץ

?

[ ]

&& הינם "AND" שמחברים שתי פקודות.

command1 && command2 ייבצע את command2 רק אם command1 תצא עם סטטוס היצאה 0 (ללא שגיאה). לדוגמא: cat file1 && cat file2 יציג את file2 רק אם התצוגה של file1 הצליחה.

|| הוא "OR" שמחבר שתי פקודות.

command2 || command1 ייבצע את command2 רק אם commadn1 יוצאת עם סטטוס היציאה ללא אפס (עם שגיאה). לדוגמא: cat file2 | cat file1 יציג את file2 רק אם file1 לא הצליח.

= מעניק ערך למשתנה.

לדוגמא: פקודה זו:  
me=blahblah

מעניקה את הערך "blahblah" למשתנה שנקרא "me". אני יכול להדפיס את שם המשתנה על ידי שימוש ב:

```
echo $me
```

\$ מקדים את שם המשתנה שיורחב.

המשתנים מוענקים על ידי שימוש ב "=" או הינם אחד מהמשתנים המוגדרים מראש (שלא ניתן להעניק אליהם):  
\$0 שם המעטפת או המעטפת שמבוצעת.  
\$# מספר פרמטרי המיקום אל הפקודה.  
\$1 הערך של הפרמטרים המוצבים המועברים אל הפקודה. \$2 הוא הפרמטר המיוחס השני המועבר אל הפקודה. וכו' עד \$9.  
\$\* מרחיב את כל הפרמטרים המוצבים שמועברים אל הפקודה.  
@\$ מרחיב את כל הפרמטרים המוצבים שמועברים אל הפקודה, אבל מצוטטים ביחידות כש"\$@" נמצא בשימוש.  
הסתכל ב man bash אם אתה באמת זקוק ליותר.

## 3.5 התקנת חבילה ומנהל החבילות rpm?

### 3.5.1 איך אני מתקין תוכנית שהורדתי מאינטרנט?

התשובה תלויה בסוג החבילה שהורדת. אתה יכול להימנע מהרבה כאבי ראש שקשורים בהתקנה אם תוריד תוכניות בצורת חבילות הבינארי של רד האט rpm\*. (זה הפורמט שאני בוחר אם ניתנת לי האפשרות לבחור).

#### התקנת חבילות הבינארי (binary) של רדהאט

\* אם התוכנית שאני מעוניין להתקין היא חבילת בינארי של רדהאט (rpm.\*). אני יכול או להשתמש בשורת הפקודה, או בכלי ממשק גרפי. אני אוהב להשתמש בכלי של שורת הפקודה בגלל שהוא מהיר וחופשי מבעיות. כלי ההתקנה מנהל החבילות של רדהאט נקרא rpm. ראשית אני קורא מידע על תוכן החבילה (אופציונלי):

```
rpm -qpi ny_new_file.rpm
```

זה מתשאל (המצב "q", חייב להיות האות הראשונה אחרי המקף) את החבילה שעדיין לא הותקנה (האופציה "p") כך שהיא מציגה את המידע (האופציה "i") שהחבילה מכילה. אם אני מעוניין להתקין את החבילה, אני מריץ (root):

```
rpm -ihv my_new_file.rpm
```

הפקודה למעלה מבצעת את עבודת ההתקנה. היא מריצה את rpm ואומרת לו להתקין את החבילה (המצב "i", חייב להיות האות הראשונה לאחר המקף) בזמן שהיא מדפיסה על גבי המסך מידע נוסף מהרגיל (האופציה "h" = מציגה "hashes" כדי להראות את התקדמות הפריקה, האופציה "v" = verbose). התכנים של החבילה מופצים אל הספריות שאליהן הם שייכים (rpm יודע להיכן הם שייכים). לאחר התקנה זו, התוכנית מוכנה להרצה, אני רק צריך לדעת את שם התוכנית (executable) ואת מיקומה. אם אני לא מצליח למצוא את התוכנית, זה מציג רשימה של הקבצים שהחבילה מכילה ביחד עם ספריות היעד שלהם:

```
rpm -qp1 my_new_file.rpm
```

פקודה זו מתשאלת (האופציה "q") את החבילה שעדיין לא מותקנת (האופציה "p") ככה שהיא מציגה (האופציה "i") את כל הקבצים שהחבילה מכילה.

כלי הממשק הגרפי (GUI) של rpm הם: gnoptro (הגירסה הישנה, שהגיע עם רדהאט 6 מבלבלת אבל הגרסאות החדשות משופרות הרבה יותר), kpackage (זמין רק עם ההפצות האחרונות), glinti (הישנה איטית מאוד, הגיע עם רדהאט 5.2).

טיפול בבעיות. rpm אמור להיות מנהל תוכנה אינטליגנטי. אם ההתקנה נכשלת אני קורא את הודעת השגיאה ואולי יכול לחשוב מה לעשות:  
(1) ההתקנה נכשלה בגלל שיש ברשותי גירסה קודמת של אותה חבילה והגרסאות מתנגשות. פתרון: אל תתקין, אלא שדרג את החבילה.

(2) ההתקנה נכשלה בגלל שיש צורך בחבילה אחרת לפני כן. עליי למצוא את החבילה החסרה ולהתקין אותה קודם, ואז לנסות ולהתקין שוב את החבילה המבוקשת. במקרים קיצוניים, אני יכול לבחור להתעלם מהחבילות שבהן החבילה שלי תלויה (dependencies) (אני באמת צריך לדעת מה אני עושה כאן כי ייתכן והתוכנה תחדל מלתפקד כראוי):

```
rpm -ivh --nodeps my_new_file.rpm
```

או אולי אפילו:

```
rpm -ivh --nodeps --force my_new_file.rpm
```

### התקנה מ-source-code tarballs

\* במידה ומה שהורדתי מהאינטרנט הוא קוד מקור של לינוקס בצורת טרבל דחוס (\*tar.gz, \*.tgz), פרוצדורת ההתקנה ארוכה יותר וקצת יותר מורכבת מאשר rpm בינארי בלבד. אני בדרך כלל מתקין את התוכנית roots.

**ראשית**, אני משנה את ספריית העבודה הנוכחית שלי ל/usr/local:

```
cd /usr/local
```

**שנית**, אני פורס את קובץ הטרבל שהורדתי מהאינטרנט:

```
tar -xvzf /home/the_dir_where_the_tarball_is/my_tarball.tar.gz
```

זה פותח (האופציה "x") את התוכן של הטרבל \*.tar.gz (או \*.tgz), מצבע עליו unzip (האופציה "z"), בזמן שהוא מדבר אלי יותר מן הרגיל (האופציה "v" verbose). בבקשה שים לב שהאופציה "f" מתפרשת כקובץ (file), ככה ששם הקובץ עוקב מיד אחרי האות "f". התכנים של הטרבל נפרסים אל תוך תת תיקיה שנקרא tar יוצר תחת ספריית העבודה הנוכחית שלי, שבמקרה הזה היא /usr/local/. הטרבל יודע כיצד צריכה להיקרא הספרייה החדשה.

אם הטרבל אינו דחוס (\*tar), אני יכול להשתמש ב:

```
tar -xvf /home/the_dir_where_the_tarball_is/my_tarball.tar
```

**שלישית**, עלי לפענח כיצד נקראת הספרייה החדשה, ואז אני עושה cd אליה:

```
dir
cd _the_new_program_subdir
```

מאחר לכמה מהספריות יש שמות ארוכים, אני משתמש באופציה ההשלמה האוטומטית כדי לחסוך בהקלדה--אני מקליד רק את הכמה אותיות הראשונות ואז לוחץ על <tab>.

**רביעית**, רוב התוכניות מקומפלות על ידי ביצוע שלושת הפקודות האלו:

```
./configure
make
make install
```



ייתכן ויקח לפקודות למעלה כמה זמן (דקה? חצי שעה?) להשלים את עבודתן. אם אחת מהן נכשלת, יהיה זה רעיון טוב לקרוא את README או את INSTALL או מה שלא מגיע עם התוכנית החדשה. ייתכן וחלק מהתוכניות יידרשו קוסטמיזציה של הסביבה (הגדרת הנתביב שלהן) או התקנת ספרייה נוספת, או משהו אחר. זה יכול לעצבן לפעמים. ייתכן ותוכניות פשוטות מאוד לא ייצטרכו את /configure. / או את "makeinstall", יכול להיות שרק "make" יספיק.

**חמישית**, אם הכל הולך כשורה, אני מוצא את התוכניות החדשות שקומפלו. שמן של התוכניות מופיע בירוק כשאני מריץ את הפקודה הזו:

```
ls --color
```

כעת, אני יכול להריץ את התוכנית, לדוגמא:

```
./the_executable
```

כמה מהתוכניות יתקינו באופן אוטומטי את התוכנית אל /usr/local/bin, ככה שאני יכול לרצות ולנסות את:

```
/usr/local/bin/the_executable
```

**שישית**, אם אני מתכוון להריץ את התוכנית לעיתים קרובות, אני יוצר לינק סימבולי אליה מהספרייה /usr/local/bin:

```
cd /usr/local/bin
```

```
ln -s /usr/local/the_new_program_subdir/the_executable
```

באופן זה, התוכנית (למעשה, לינק סימבולי אליה) נמצאת בנתיב שלי וניתנת להרצה פשוט על ידי הקלדת שמה (אין צורך להקליד את הנתיב המלא אליה יותר). כמה תוכניות יתקינו את התוכנית (או לינק אליה) אל תוך הספרייה "bin" במקרה זה דלג על הצעד האחרון.

### התקנה מחבילת rpm קוד מקור

ישנן גם תוכניות שמופצות כחבילות "source code rpm". הן דורשות התקנה של חבילת rpm. \* ביחד עם כלי rpm כמתואר בחלק הראשון של פרק זה. אבל מאחר וה "rpm" מתקין את קוד המקור (בדרך כלל בקוד מקור של שפת "C"), עליי לקמפל את קוד המקור על ידי ביצוע הרצף: "configure ; make ; make install". כפי שעשיתי לקוד מקור שמופץ כטרבול (ראה תשובה קודמת).

חלק 5: קיצורי דרך ופקודות

## חלק 5 : קיצורי דרך ופקודות

### תוכן החלק הזה:

- 5.1 קיצורי דרך חיוניים ופקודות שפיות
- 5.2 פקודות עזרה
- 5.3 מידע מערכת
- 5.4 פעולות בסיסיות
- 5.5 ניהול קבצים
- 5.6 צפייה ועריכת קבצים
- 5.7 מציאת קבצים
- 5.8 היסודות של X-windows
- 5.9 אפליקציות רשת
- 5.10 דחיסה ופריסה של קבצים
- 5.11 שליטה על תהליכים
- 5.12 פקודות ניהול בסיסיות
- 5.13 כלי ניהול דיסק
- 5.14 ניהול חשבונות משתמשים והרשאות קבצים
- 5.15 התקנת תוכנה
- 5.16 גישה לכווננים/מחיצות
- 5.17 כלי ניהול רשת
- 5.18 פקודות שקשורות למוזיקה
- 5.19 פקודות שקשורות לגרפיקה
- 5.20 משחקים קטנים

**הקדמה.** זהו מבחר פרקטי של פקודות בהן אנו משתמשים לעיתים קרובות, פקודות שאנו מוצאים כיעילות, וכאלה שהגיעו עם `cd` של הפצות הלינוקס שלנו ( `RH` או `MDK` ). הקלק על `<tab>` בשורת פקודה ריקה כדי לראות רשימה של כל הפקודות האפשריות (בנתיב שלך). במערכת הביתית הקטנה שלי, הפקודה מראה שיש 3786 תוכניות ברות הרצה בנתיב שלי. ניתן לגשת לרבות מ"פקודות" אלו מהממשק הגרפי האהוב עליך ( `KDE` או `Gnome` ) על ידי הקלקה על התפריט או הפתור המתאימים. את כולן ניתן להריץ משורת הפקודה (אלא אם כן לא התקנת את החבילה, אבל כולן הגיעו עם `cd` שלנו). תוכניות שדורשות ממשק גרפי ( `GUI` ) צריכות להיות מורצות תחת ממשק גרפי, למשל מטרמינל שנפתח תחת `KDE` או `Gnome` ( `Xterm` ). כלים מתקדמים יותר (פחות שימושים למשתמש מתחיל?) מתוארים בחלק "לומדים עם לינוקס" במדריך הזה.

## הערות לאלו שלא מכירים `unix`:

1. לינוקס היא `CASE-SENSITIVE`. לדוגמא: `Netscape`, `NETSCAPE`, `nEtscape`, ו- `netscape` יהיו ארבע פקודות שונות ( אבל מתוך הארבע, רק `netscape` זמינה במערכת שלי). בנוסף גם `my_file`, `my_file`, `my_file` ו- `my_FILE` הינם שלושה קבצים שונים. גם שם הלוגאין שלך כמו הסיסמא שלך, הינם `case sensitive`. (זה נובע מהמסורת של יוניקס ושל שפת התיכנות `c`, שניהן `case sensitive`).
2. שמות קבצים יכולים להכיל עד 256 תווים ולהכיל אותיות, מספרים, נקודות, `_` קו תחתון, - מקפים, ועוד כמה תווים לא מומלצים אחרים.
3. קבצים עם שמות שמתחילים ב". (נקודה) בדרך כלל לא מוצגים על ידי הפקודה `ls (list)` או על ידי הפקודה `dir`. תחשוב על קבצים אלו כעל קבצים "נסתרים". השתמש ב `ls -a` (עם האופציה "all") כדי לראות קבצים אלו.
4. "/" הוא המקבילה של ה"\" מדוס (ספריית השורש, ההורה של כל שאר הספריות, או מפריד בין שם ספרייה לתת תיקיה או שם קובץ). לדוגמא, `cd /usr/doc` נסה.
5. תחת לינוקס, כל הספריות מופיעות תחת עץ אחד (אין אותיות כונן כמו בדוס). זה אומר שספריות וקבצים מכל ההתקנים הפיזיים מתמזגים אל עץ אחד ויחיד.
6. בקובץ קונפיגורציה, שורה שמתחילה ב# היא הערה. כשאתה משנה קובץ קונפיגורציה אל תמחק את ההגדרות הישנות--סמן את השורות המקוריות ב#. הכנס תמיד תאור הסבר קצר שמתאר מה עשית (לטובתך!).
7. לינוקס מרובת משתמשים באופן תורשתי. ההגדרות האישיות שלך (וכל שאר הקבצים האישיים) נמצאות בספריית הבית שלך תחת `/home/your_user_login_name`. הגדרות רבות נשמרות בקבצים בעלי שמות שמתחילים בנקודה "." כדי להשאיר אותן מחוץ לדרך שלך (הסתכל ב3 למעלה).
8. הגדרות שנוגעות לכלל המערכת נשמרות בספרייה `/etc`.
9. תחת לינוקס, כמו בכל מערכת הפעלה מרובת משתמשים, לספריות ולקבצים יש בעלים וסט של הרשאות. בדרך כלל תורשה לכתוב רק אל ספריית הבית שלך. למד כיצד להשתמש בהרשות קבצים, לינוקס תציק לך הנושא הזה.
10. אופציות של פקודה מוצגות על ידי "- מקף, כשבעקבותיו מופיעה אות אחת --

כשהאופציה היא יותר מאות אחת). מקף "-" הוא המקבילה של ה"/" מדוס. לדוגמא, נסה את  
rm --help.

11. הקלד command& (שם הפקודה עם &) כדי להתחיל פקודה ברקע. זוהי בדרך כלל  
הדרך המועדפת להתחלת תוכנית מטרמינל X-windows.

### 5.1 קיצורי דרך חיוניים ופקודות שפיות

<Ctrl><Alt><F1>

עבור אל טרמינלי הטקסט הראשונים. תחת לינוקס יכולים להיות לך כמה (6 בתצורה  
סטנדרטית) טרמינלים פתוחים באותה העת. זהו קיצור דרך שאומר: "לחץ על המקשים  
קונטרול ואלט, החזק אותם, כעת לחץ על <F1>. שחרר את כל המקשים".

(n=1..6) <Ctrl><Alt><Fn>

עבור אך טרמינל הטקסט הבא. (ניתן להשיג את אותה התוצאה בעזרת הפקודה הלא נפוצה  
chvt n .chvt "זה "change virtual terminal". בטרמינל טקסט (מחוץ ל X), אתה יכול  
גם להשתמש ב<Alt><Fn> (אין צורך במקש <Ctrl>).

tty

הצג את השם של הטרמינל בו אתה מקליד פקודה זו. אם אתה מעדיף את המספר של  
הטרמינל הפעיל (במקום שמו), אתה יכול להשתמש בפקודה fgconsole (= "foreground"  
console).

<Ctrl><Alt><F7>

עבור אל טרמינל ה GUI הראשון (אם X-windows עובד בטרמינל ה7, היכן שהוא בדרך כלל  
נמצא).

(n=7..12) <Ctrl><Alt><Fn>

עבור אל טרמינל ה GUI הח (במידה וטרמינל GUI עובד). כברירת מחדל, שרת ה X הראשון  
רץ בטרמינל 7. כברירת מחדל כלום לא רץ בטרמינלים 8-12--אתה יכול להפעיל שם שרת X  
חליפי.

<Tab>

(בטרמינל טקסט או ב X) השלמה אוטומטית לפקודה אם ישנה רק אופציה אחת, אם ישנן  
כמה אופציות זמינות מראה את כולן. במערכות חדשות יותר לחץ <Tab><Tab>. זהו קיצור  
דרך מעולה, הוא באמת יכול לחסוך לך המון זמן.

<ArrowUp>

(בטרמינל טקסט או ב X) גלול וערוך את היסטוריית הפקודה. לחץ על <Enter> כדי לבצע  
פקודה מההיסטוריה (כדי לחסוך בהקלדה). <ArrowDown> גלול חזרה.

<Shift><PgUp>

גלול את פלט הטרמינל למעלה. זה עובד בזמן הלוגאין, כך שאתה יכול לגלול דרך הודעות  
האיתחול. גודל זיכרון הוידאו שלך קובע כמה רחוק אתה יכול לגלול את התצוגה שלך  
אחורה. <Shift><PagDown> יגלול את פלט הטרמינל למטה.

<Ctrl><Alt><+>

<X-windows> שנה לרזולוצייה הבאה בשרת הX (אם התקנת את שרת הX ליותר מרזולוצייה אחת). כדי להשיג כמה רזולוציות בכרטיס/מסך הVGA הסטנדרטי שלי, יש לי את השורה הבאה בקובץ /etc/X11/XF86Config (הרזולוצייה הראשונה מתחילה כברירת מחדל, הרזולוצייה הגבוהה ביותר קובעת את גודל ה"מסך הווירטואלי"):

```
modes "1024x768" "800x600" "640x480" "512x384" "480x300" "400x300"
"1152x864"Z
```

ברור, שלפני כן עליי לקנפג את שרת הX, על ידי שימוש ב Xconfigurator, xf86config, או באופן ידני על ידי עריכת הקובץ /etc/X11/XF86Config, ככה שהוא יתמוך ברזולוציות הנ"ל (בדרך כלל החסרת הסימון מהשורה שמגדירה את הצ'יפסט של כרטיס הווידאו שלי, וציון תדירויות הסינכרון שבהן המסך שלי תומך). XFdrake (כלי הגדרת התצורה של מנדרייק) יכול לעשות את זה מתוך GUI. הסתכל גם בפקודות xvidtune ובxvidgen.

<Ctrl><Alt><->

(X-windows) משנה אל רזולוציית שרת הX הקודמת.

<Ctrl><Alt><Esc>

(X-windows, KDE) הרוג את החלון שעליו אני הולך להקליק עם סמן העכבר שלי (הסמן משתנה אל משהו שנראה כמו סמל של מוות). ניתן להשיג תוצאה דומה על ידי שימוש בפקודה xkill (מוקלדת בטרמינל X). שימושי כאשר תוכנית X-windows לא רוצה להיסגר (תקועה?).

<Ctrl><Alt><BkSp>

(בטרמינל X-windows) הרוג את שרת הX הנוכחי. השתמש בזה אם לא ניתן לצאת משרת הX באופן נורמלי.

<Ctrl><Alt><Del>

(בטרמינל טקסט) כבה את המערכת ואתחל. זוהי פקודת כיבוי נורמלית למשתמש בקונסול מצב טקסט. אל תלחץ סתם על "reset" כדי לכבות את המחשב!

<Ctrl>c

הרוג את התהליך הנוכחי (עובד בעיקר עם אפליקציות מצב טקסט קטנות).

<Ctrl>d

(מוקלק בתחילתה של שורה ריקה) בצא לוג אוטו מהטרמינל הנוכחי. ראה גם את הפקודה הבאה.

<Ctrl>d

שולח [End-of-File] לתהליך הנוכחי. אל תקליק אותו פעמיים אחרת תצא מהטרמינל (ראה את הפקודה הקודמת).

<Ctrl>s

עצור את ההעברה לטרמינל.

<Ctrl>q

המשך את ההעברה אל הטרמינל. נסה את זה אם באופן מיסטורי הטרמינל שלך מפסיק להגיב. ראה את הפקודה הקודמת.

<Ctrl>z  
שלח את התהליך הנוכחי אל הרקע.

exit  
צא מהמערכת (logout). אני יכול להשתמש גם בlogout לאותה תכלית. (אם התחלת מעטפת שניה, על ידי שימוש בbash, פקודה זו תוציא אותך מהמעטפת השניה, ואתה תחזור אל המעטפת הראשונה, ולא תצא מהמערכת. השתמש בעוד exit בשביל לצאת לגמרי.)

reset  
שחזר טרמינל דפוק (טרמינל שמראה תוים מוזרים) אל הגדרות ברירת המחדל שלו. השתמש בזה אם ניסית לעשות cat לקובץ בינארי. ייתכן ולא תוכל לראות את הפקודה בזמן שאתה מקליד אותה, אבל היא עדיין עובדת.

<MiddleMouseButten>  
הדבק את הטקסט שמודגש כרגע למקום אחר. זוהי פעולת ה"copy-paste" הנורמלית בלינוקס. זהו תחלף מהיר ועוצמתי לפעולת ה"copy-paste" מבוססת התפריט שמוכרת למשתמשי ממשק גרפי. (היא אינה עובדת עם גרסאות ישנות של נטסקייפ אשר משתמשות ב"copy-paste" בסגנון Mac/Ms. היא כן עובדת בטרמינל טקסט אם איפשרת את השירות "gpm" על ידי setup. היא עובדת גם בכל קופסאות הדיאלוג, וכו'.--מאוד נח!).  
נח להשתמש בפונקציה זו בעיקר בעכברים בעלי 3 כפתורים (לוג'יטק ודומיו) או שתקבע את הפונקצייה "3 mouse button emulation". הכפתור האמצעי בעכבר מדומה בדרך כלל על ידי לחיצה על שני הכפתורים של העכבר סימולטנית.

~  
(tilde) ספריית הבית שלי (בדרך כלל הספרייה /home/my\_login\_name). לדוגמא, הפקודה cd ~/my\_dir תשנה את ספריית העבודה שלי אל תת הספרייה "my\_dir" תחת ספריית הבית שלי. הקלדה של cd לבד היא המקבילה של הפקודה "cd ~". אני שומר את כל הקבצים שלי בספריית הבית שלי.

.  
(נקודה) הספרייה הנוכחית. לדוגמא, /my\_program . תבצע את הקובץ "my\_program" שממוקם בספריית העבודה הנוכחית שלך.

..  
(שתי נקודות) ספרייה הורה לספרייה הנוכחית. לדוגמא, הפקודה . cd תשנה את ספריית העבודה הנוכחית לרמה אחת למעלה.

כמה קיצורי דרך נוספים של KDE (עוזר, אבל לא הכרחי)

<Alt><Tab> עבור דרך חלונות. כדי לחזור, ללכת אחורה: <Alt><Shift><Tab>  
<Ctrl><Tab> עבור דרך שלוחנות עבודה. כדי לחזור: <Ctrl><Shift><Tab>  
<Ctrl><Esc> הראה את טבלת התהליכים שרצים במערכת שלי. אפשר לי להרוג כל אחד מהתהליכים שהתחלתי (או לשלוח אליהם סיגנלים אחרים).  
<Alt><F1> גישה אל ה K-menu ("המקבילה של התפריט start במייקרוסופט ווינדוס").  
<Alt><F12> דמה את העכבר על ידי שימוש במקשי החיצים שעל המקלדת.  
<Alt><LeftMouse><Button> גרור חלון כדי להזיז אותו. בדרך כלל, אני מזיז חלון על ידי

גרירת הבר העליון שלו, אבל לעיתים אני מצליח להוציא אותו אל מחוץ למסך. בעזרת קיצור דרך זה, אני יכול לגרור על ידי כל חלק מהחלון.

<Alt><Printscreen> צלם תמונת מסך של החלון הנוכחי אל תוך הקליפבורד.  
<Ctrl><Alt><PrintScreen> צלם תמונת מסך של כל שולחן העבודה אל תוך הקליפבורד.  
<Ctrl><Alt><L> נעל את שולחן העבודה.  
<Ctrl><Alt><D> הראה/הסתיר את שולחן העבודה (נהדר אם אתה רוצה להחביא משחק סוליטר כשהבוס שלך נכנס לחדר).

<Alt><SysRq><command\_key>  
(לא חיוני ממש) קבוצה שלי קומביניציות שמוטמעת אל תוך רמת הקרנל של לינוקס ( a low level). זאת אומרת, הסיכויים הם שקומביניציות אלו יעבדו רוב הזמן. קומביניציות אלו ייעדו בעיקר ל debugging ומקרי חירום (למפתחים בעיקר) כדאי שתנסה פתרונות אחרים, בטוחים יותר, לפני כן. המקש <SysRq> ידוע גם כ <PrintScreen>. ניתן לבטל/לאפשר את הקומביניציות הללו על ידי קביעת משתנה קרנל רלוונטי ל "1" או ל "0", זאת אומרת, `echo "1" > /proc/sys/kernel/sysrq`  
<Alt><SysRq><k> הרוג את כל התהליכים (כולל X) שרצים בקונסול הווירטואלי הפעיל הנוכחי. קומביניציה זו ידוע גם כ- "secure access key" (SAK).  
<Alt><SysRq><i> שלח את האות KILL לכל התהליכים הפעילים פרט ל init. זה יכול להיות יעיל יותר בחיסול תהליכים ש"בורחים" מאשר השילוב הקודם, אבל יכול לגרום לכמה מהם לצאת באופן אבנורמלי.  
<Alt><SysRq><l> שלח את האות KILL לכל התהליכים, כולל init. המערכת לא תהיה פונקציונלית.  
<Alt><SysRq><s> הרץ סינכרון חירום (cache write) על כל מערכות הקבצים המומרות. זה יכול למנוע אובדן נתונים.  
<Alt><SysRq><u> המר שוב את כל מערכות הקבצים כקריאה בלבד. בעל אותו האפקט כמו לקומביניציית הסינכרון למעלה, אבל עם ייתרון אחד חשוב: אם הפעולה מוצלחת, fsck לא יאלץ לבדוק את כל מערכות הקבצים לאחר reset של חומרת המערכת.  
<Alt><SysRq><r> כבה את keyboard raw mode. זה יכול להועיל כשיח X שלך נתקע. לאחר הזנת פקודה זו תוכל להשתמש ב <CTRL><ALT><DEL>.  
<Alt><SysRq><b> אתחל מחדש מייד ללא סינכרון או המרה של הדיסקים. קרוב לוודאי שתיים עם שגיאות במערכת הקבצים.  
<Alt><SysRq><o> כבה את המערכת שלך (אם מוגדר ונתמך).  
<Alt><SysRq><p> השלך (dump) את הרשומות והדגלים הנוכחיים שלך אל הקונסול.  
<Alt><SysRq><t> השלך רשימה של מטלות נוכחיות ואת המידע שלהן אל הקונסול שלך.  
<Alt><SysRq><m> השלך זיכרון אל תוך הקונסול שלך.  
<Alt><SysRq><digit> הסיפורה היא "0" עד "9". קבע את רמת הרישום של הקונסול, שולט באיזה הודעות יודפסו אל תוך הקונסול שלך. לדוגמא, "0" ייגרום להודעות חירום בלבד להופיע על הקונסול שלך, כמו PANIC או OOPS.  
<Alt><SysRq><h> מציג עזרה. בנוסף, כל קומביניציית <Alt><SysRq><key> שאינה נתמכת תציג את אותה עזרה.

## 5.2 פקודות עזרה

any\_command --help | more  
מציג עזרה זריזה של פקודה (עובד עם רוב הפקודות). לדוגמא, נסה `cp --help | more`.  
"--help" עובד בדומה ל"/h" מדוס. השירשור "more" הכרחי כשהפלט ארוך יותר מגודל המסך.

man topic  
מציג את התוכן של דפי ההסברים על הנושא. לחץ על "q" כדי לצאת ממצב צפייה. נסה את `man man` אם אתה זקוק לאופציות מתקדמות. הפקודה `topic info` עובדת בדומה ל `man topic`, אבל יכולה להכיל מידע עדכני יותר. דפי ההסברים יכולים להיות קשים לקריאה--הם נכתבו למתכנתי יוניקס. נסה את `any_command --help` לעזרה קלה יותר לעיקול לגבי פקודה. כמה תוכניות מגיעות עם README או עם קבצי מידע אחרים--הסתכל בספרייה `/usr/share.doc` כדי להציג דף הסברים מחלק ספציפי. אני יכול להשתמש במשהו כמו: `man 3 exit` (זה מציג מידע לגבי הפקודה `exit` מהחלק 3 של דפי ההסברים) או אולי `man -a exit` ( זה מציג דפי הסברים לגבי הפקודה `exit` מכל החלקים). החלקים של `man` הינם:  
חלק 1-user commands, חלק 2-system calls, חלק 3-subroutines, חלק 4-devices,  
חלק 5-file formats, חלק 6-games, חלק 7-miscellaneous, חלק 8-system  
administration, חלק 9, חלק n-new. כדי להדפיס דף הסברים, אני משתמש ב:  
`man topic | col -b` (האופציה `col -b` מסירה כל רווח או תוים אחרים שעלולים לגרום לדפי ההסברים המודפסים להיות קשים לקריאה).

info topic  
מציג את התוכן של מידע לגבי פקודה ספציפית. `info` הינו התחליף לדפי ההסברים נכה שהוא מכיל את המידע העדכני ביותר לתייעוד המערכת. השתמש ב<Space> וב-<BkSpace> כדי לנוע, ייתכן ואתה עלול להתבלבל. הקלד על "q" כדי לצאת. תחליף לצפייה המבלבלת משהו ב`info` יכול להיות `pinfo`- השתמש בו אם אתה מוצא אותו מתאים יותר.

apropos topic  
תן לי את רשימת הפקודות שיש להן איזה שהוא קשר אל הנושא שלי.

whatis topic  
תן לי רשימה קצרה של פקודות שתואמות לנושא שלי. `whatis` דומה ל`apropos` (ראה את הפקודה למעלה)--שניהם משתמשים באותו מסד הנתונים. אבל, `whatis` מחפש מילות מפתח בעוד `apropos` מחפש גם את התיאור של מילות המפתח.

help command  
הצג מידע זריז לגבי פקודה שמוטמעת בתוך `bash` (shell). שימוש ב`help` ללא פקודה מציג רשימה של כל הפקודות המוטמעות אל תוך `bash`. קרוב לודאי שרשימה קצרה של פקודות תכיל את: `alias, bg, cd, echo, exit, export, fg, help, history, jobs, kill, logout, pwd, set, source, ulimit, umask, unalias, unset`.

kdehelp  
kdehelpcenter  
(בטרמינל X, שתי פקודות, השתמש בזו שעובדת במערכת שלך). צפה בכל העזרה של



המערכת על ידי שימוש במנווט גרפי של KDE. בדרך כלל, העזרה של KDE מופעלת על ידי לחיצה על האייקון המתאים בלוח בקרה של KDE. השתמש ב `gnome-help-browser` בשביל המקבילה של GNOME.

### 5.3 מידע מערכת

`pwd`  
הצג את השם של הספרייה הנוכחית שלי על המסך.

`hostname`  
הצג את השם של המארח המקומי (המכונה בה אני עובד).

`whoami`  
הצג את שם הלוג אין שלי.

`id user name`  
הצג מזהה המשתמש (`uid`) ואת מזהה הקבוצה שלו (`gid`), המזהה האפקטיבי שלו (הוא שונה מהמזהה האמיתי) ואת הקבוצות הנוספות שלו.

`date`  
הצג את התאריך הנוכחי של מערכת ההפעלה, את הזמן ואזור הזמן. בשביל פורמט בסטנדרט ISO, עליי להשתמש ב:

`date -lseconds`  
אני יכול לשנות את התאריך והזמן ל 2000-12-31 23:57 על ידי שימוש בפקודה:  
`date 123123572000`

או על ידי שמוש בשתי פקודות אלו (קלות יותר לזכירה):  
`date --set 2000-12-31`  
`date --set 23:57:00`

כדי לקבוע את שעון החומרה (BIOS) מתוך שעון המערכת (LINUX), אני יכול להשתמש בפקודה (כroots):  
`setclock`

לפורמט הסטנדרט הבינלאומי (ISO 8601) לתאריך/שעה מספריים יש את הצורה: 2001-01-31 (כמו ללוקליזציה ברירת המחדל של "C" בלינוקס). אתה יכול להיות דייקן יותר אם אתה רוצה להשתמש לדוגמא ב: 2001-01-31 23:59:59.999-0500. (מיצג מילישניה אחת לפני פברואר 2001, באזור זמן שנמצא 5 שעות אחרי קורדינטת הזמן האוניברסלית (UTC). התצוגה ה"כשרה" ביותר לאותה נקודת זמן יכולה להיות: 20010131t235959,999-0500, הסתכל בסטנדרט: <ftp://ftp.qsl.net/pub/glsmd/8601v03.pdf>.

`time`  
קבע את משך הזמן שלוקח לתהליך להשלים+חשבונות תהליכים אחרים. אל תתבלבל עם הפקודה `date`. אני יכול למצוא כמה זמן לוקח להציג תוכן של ספרייה על ידי שימוש ב: `time ls`. או שאני יכול לבחון את תיפקוד הזמן על ידי `time sleep 10` (מודד זמן לפקודות שלא עושות דבר למשך 10 שניות).

`clock`  
`hwclock`  
(שתי פקודות, השתמש בשניהן). השג תאריך/זמן משעון חומרת המחשב (זמן אמיתי, מונע

סוללה). אתה יכול להשתמש באחת מהפקודות הללו כדי לקבוע את זמן שעון החומרה, אבל setclock יכול להיות פשוט יותר (ראה שתי פקודות למעלה). דוגמא:  
utcsystoch --hwclock, קובע את שעון החומרה (UTC) מתוך שעון המערכת.

who

הצג את מי שנמצא בלוגאין במערכת.

w

הצג מי נמצא בלוגאין במערכת, מה הוא/הם עושים, השימוש שלהם במעבד, וכו'.  
פקודה שימושית לאבטחה.

rwho -a

הצג משתמשים שמחוברים למחשבים אחרים ברשת שלך. השירות rwho חייב להיות פעיל כדי שפקודה זו תעבוד. אם הוא אינו פעיל, הרץ את setup (RH) כroots כדי לאפשר את "rwho".

finger user\_name

מידע מהמערכת אודות משתמש. נסה את finger root. אתה יכול להשתמש בfinger מול כל מחשב ברשת שחושף את השירות finger לעולם. אני יכול לנסות:  
finger@finger.kernerl.org

last

הצג רשימה של משתמשים שהיו מחוברים (logged-in) למערכת שלך. זה רעיון טוב לבדוק את הנושא הזה מפעם לפעם כמדד לאבטחה של המערכת שלך.

lastb

(=last bad) מראה את נסיונות הלוג אין האחרונים שנכשלו. זה לא עבד במערכת שלי, אז התחלתי אותו בעזרת: touch /var/log/btmp.

"ישנה סיבה טובה לעובדה ש/var/log/btmp אינו זמין בכל מערכת שמוגדרת בשפיות- זהו קובץ קריא לעולם כולו שמכיל נסיונות לוג אין שנכשלו. מכיון שאחת מטעויות הלוג אין הנפוצות ביותר היא להקליד את הסיסמא במקום את שם המשתמש, הקובץ /var/log/btmp הוא מתנה לקרקרים" (תודה לברוס ריצ'רדסון). נראה שניתן לפתור את הבעיה על ידי שינוי ההרשאות על הקובץ ככה שרק root יוכל להשתמש בlastb.  
chmod -o-r /ar/log/btmp

history | more

מראה את הפקודות האחרונות (בערך 1000) שבוצעו משורת הפקודה לחשבון הנוכחי. ה more | גורם לתצוגה להעיצר לאחר כל מסך מלא. כדי לראות מה משתמש אחר עשה במערכת שלך, בצע לוג אין כroots ובדוק את ה"history" שלו/שלה. ההיסטוריה נשמרת בקובץ bash\_history. בספריית הבית של המשתמש (כן, ניתן לשנות את הקובץ ולמחוק אותו).

uptime

הצג את פרק הזמן שעבר מאז האיתחול האחרון.

ps

(= print staus או process status) הצג את התהליכים שמבוצים כרגע על ידי המשתמש הנוכחי.

ps axu | more  
הצג את כל התהליכים שרצים כרגע, אפילו את אלו שללא טרמינל השליטה, ביחד עם שם המשתמש שבבעלותו נמצא כל תהליך.

top  
תצוגה מתעדכנת של התהליכים שרצים כרגע על גבי המחשב שלי, מסודרים לפי רמת שימוש בcpu.

process identification=PID

USER=השם של המשתמש אשר התהליך נמצא בבעלותו (התחיל את התהליך?)

priority of the process=PRI (ככל שהמספר גבוה יותר כך העדיפות שלו נמוכה יותר. מאפס הוא המצב הנורמלי, העדיפות הגבוהה ביותר היא 20-, הנמוכה ביותר היא 20.

niceness level=NI (אם התהליך מנסה להיות "נחמד" על ידי כיוונון העדיפות לפי המספר הנתון). ככל שהמספר גבוה יותר, ככה רמת ה"נחמדות" שלו גבוהה יותר (העדיפות שלו נמוכה יותר).

SIZE=קילו בייטס של code+data+stack שנלקחו מהזיכרון על ידי התהליך.

RSS=קילו בייטס של זיכרון פיזי (סיליקון) שנלקחו.

SHARE=קילו בייטס של זיכרון שנחלק עם תהליכים אחרים.

STAT=מצב התהליך: S=SLEEPING, R=running, T=stopped or traced, D=uninterruptable sleep, Z=zombie.

%CPU=החלק מהשימוש בcpu (מאז עידכון המסך האחרון).

%MEM=החלק מהשימוש בזיכרון פיזי.

TIME=סך כל זמן המעבד שבשימוש תהליך (מאז שהתהליך התחיל).

COMMAND=שורת הפקודה שהפעילה את המשימה (היה זהיר עם סימאות וכו', בשורת הפקודה, כל מי שרשאי להריץ "top" יכול לראות אותן).

gtop

ktop

(בטרמינל X) שתי אפשרויות בחירה לממשק גרפי לtop. החביבה עלי היא gtop (מגיעה עם gnome). בKDE, ktop זמינה גם מה menu "K" תחת "task manager"- "system".

uname -a

(="UNIX NAME" עם האופציה "all") מידע לגבי השרת (המקומי) שלך. אני יכול להשתמש גם בguname (בטרמינל X) כדי להציג מידע בצורה נעימה יותר.

XFree86 -version  
הראה את הגירסה של X-windows שמותקנת אצלי במערכת.

cat /etc/issue  
בדוק באיזו הפצה אתה משתמש. אתה יכול לשים הודעה משלך בקובץ טקסט זה--הוא מוצג בכניסה למערכת. בדרך כלל נהוג לשים את הודעת הכניסה של האתר שלך בקובץ /etc/motd ("motd"="message of the day")

free  
מידע לגבי הזיכרון (kilobytes). זיכרון משותף -"shared"- הוא הזיכרון שניתן לשתף בין תהליכים. זיכרון "buffered" ו-"cached" הוא החלק ששומר חלקים מקבצים שניגשו אליהם לאחרונה--הוא יכול להתכווץ אם תהליכים זקוקים לזיכרון נוסף.

df / -bh | more  
(disk usage=) מדפיס מידע מפורט לגבי שימוש בדיסק לכל תת סיפריה אשר מתחילה בספריית ה"/(root), (באופן קריא לבני אנוש).

cat /proc/interrupts  
מציג את האותות הפסיקה שבשימוש. יכול להיות שמיש לפני התקנת חומרה חדשה.

cat /proc/version  
גירסת הלינוקס ומידע נוסף.

cat /proc/filesystems  
מציג את סוגי מערכות הקבצים שבשימוש כרגע.

cat /proc/printcap | more  
מראה את הגדרת המדפסות.

lsmod  
(="list modules", root, השתמש ב /sbin/lsmod כדי לבצע פקודה זו כשאינך root).  
מציג את מודולי הקרנל שטעונים כרגע.

set | more  
הצג את סביבת המשתמש הנוכחית (במלואה). בדרך כלל יותר מדי בכדי לטרוח.

echo \$PATH  
הצג את התוכן של המשתנה הסביבתי "PATH". פקודה זו יכולה להיות בשימוש גם כדי להראות משתנים סביבתיים אחרים. השתמש בset כדי לראות את הסביבה המלאה (ראה פקודה קודמת).

dmesg | less  
הצג הודעות קרנל (התוכן של מה שנקרא kernel ring buffer). לחץ על "q" כדי לצאת מ"less". השתמש ב /var/log/dmesg כדי לראות מה "dmesg" "זרק" אל תוך קובץ זה לאחר איתחול המערכת האחרון.

change -l my\_login\_name  
מצגי את תאריך תפוגת הסיסמא שלי.

quota  
הצג את מיכסת הדיסק שלי (ההגבלות על השימוש שלי בנפח דיסק).

sysctl -a | more  
הצג את כל פרמטרי הקינפוג של לינוקס.

runlevel  
מצגי את runlevel הקודם והנוכחי. הפלט "N5" מתפרש כ:"אין runlevel קודם" ו-"5" הוא runlevel הנוכחי". כדי לשנות את runlevel, השתמש בinit.  
init 1 מעביר את המערכת לsingle user mode.

runlevel הוא המצב של פעולת לינוקס. ניתן לשנות runlevel "על המקום" על ידי שימוש בפקודה init.  
לדוגמא, init 3 (root) יעביר אותי לrunlevel 3. runlevels הבאים הם הסטנדרט:  
0-כיבוי (אל תגדיר את initdefault לזה).  
single user mode-1.  
2-multiuser, ללא NFS (זהה ל3 אם אין לך רשת).  
3-מצב multiuser מלא.  
4-לא בשימוש.  
5-X11.  
6-איתחול (אל תגדיר את initdefault לזה).  
runlevel ברירת המחדל של המערכת נקבע בקובץ: /etc/inittab.

sar  
צפה במידע שנלקח מקובץ רישום הפעילות של המערכת (/var/log/sarxx) כשא הוא מספר היום הנוכחי).  
sar יכול להוציא סוגים רבין של סטטיסטיקות לגבי המערכת כולל ממוצעי עומס של המעבד, סטטיסטיקות לגבי קלט/פלט (i/o), סטטיסטיקות לגבי תעבורת הרשת ליום הנוכחי וגם (בדרך כלל) מספר ימים אחורה.

## 5.4 פעולות בסיסיות

ls  
dir  
הצג את התוכן של הספירייה הנוכחית. הפקודה dir היא שם נוסף (alias) ל ls ככה שלבסוף שתי פקודות אלו עושות את אותה הפעולה. רשימת הקבצים מקודדת בצבעים בדרך כלל:  
כחול כהה=ספריות  
אפור בהיר=קבצים רגילים  
ירוק=קבצים שניתנים להרצה (executables)  
ארגמן=קבצים גרפיים  
אדום=קבצים דחוסים (zipped)  
כחול בהיר=לינקים סיבוליים  
צהוב=קבצי התקנים  
חום=FIFO

ls -al | more  
הצג את תוכן הספרייה הנוכחית, את כל הקבצים (כולל אלו שמתחילים בנקודה), ובאופן התצוגה הארוך שלהם. שרשר את הפלט דרך הפקודה "more", ככה שהתצוגה תעצור לאחר כל מסך מלא. לפקודה ls יש כמה אופציות שימושיות. לכמה מהן יש קיצורי דרך (aliases) כדי להימנע מהקלדה מרושלת. נסה את ll (alias, long ls=) לפקודה (ls -l). אופציה נוספת שבה אני משתמש באופן די קבוע היא ls -ad (הצג את כל תת הספריות בספרייה הנוכחית שלי, אבל אל תציג את התוכן שלהן).

cd directory  
שנה ספרייה. שימוש בcd ללא שמה של הספרייה ייקח אותך אל ספריית הבית.  
- "cd" תיקח אותך אל הספרייה הקודמת שלך, זו דרך טובה לנוע בין שתי ספריות.  
.. "cd" תיקח אותי אל ספרייה אחת למעלה (שימושי מאוד).

./program\_name  
הרץ קובץ בר הרצה (executable) בספרייה הנוכחית. יש צורך ב./ כשהקובץ אינו בנתיב שלי. קובץ בר הרצה שנמצא בנתיב שלי ניתן להרצה פשוט על ידי שימוש ב:program\_name.

shutdown -h now  
(כ root) כבה את המערכת לגמרי. בדרך משמש לכיבוי מרחוק. השתמש ב<Ctrl><Alt><Del> לכיבוי מהקונסול (על ידי כל משתמש).

hAlt  
reboot  
init 6  
(כ root, שלוש פקודות) כבה או אתחל את המכונה. משמש לכיבוי מרחוק, פשוט יותר להקלדה מהפקודה הקודמת. בנוסף, מועיל כשהמחשב "נתקע" (אני מאבד שליטה על המקלדת)--אני פותח telnet ממכונה אחרת ברשת ומאתחל את המחשב מרחוק. אני משתמש ב<Ctrl><Alt><Del> לכיבוי נורמלי מול הקונסול למחשב מקומי.

vlock  
(לא זמין בגירסאות ישנות של RH) נעל טרמינל (מצב טקסט) מקומי. אני יכול להשתמש בvlock גם כדי לנעול את כל הטרמינלים (כנראה לא רעיון טוב). נראה שהכי מועיל יהיה לצאת מהמערכת בלוג אאוט. אתה לא משתמש בvlock בממשק גרפי--מנהלי החלונות באים עם שומר מסך מוגן סיסמא וכלי נעילה (האייקון הקטן עם המנעול בKDE, קיצור הדרך <Ctrl><Alt><I>).

## 5.5 ניהול קבצים

cp source destination  
מעתיק קבצים, cp /home/stan/existing\_file\_name יעתיק קובץ אל ספריית העבודה הנוכחית שלי. השתמש באופציה -R (מתפרש כ "recursive") כדי להעתיק תוכן של עץ ספרייה שלם. cp -R my\_existing\_dir/ ~ יעתיק תת ספרייה תחת ספריית העבודה הנוכחית שלי אל ספריית הבית שלי.

`mcopy source destination`  
העתק קובץ מ/אל מערכת קבצים של DOS (המרה של מערכת הקבצים אינה הכרחית).  
למשל, `mcopy a:\autoexec.bat ~/junk`. ראה את `man mtools` לפקודות אחרות  
שיכולות לגשת אל קבצי DOS ללא המרה: `mmdir, mcd, mren, mdel, mmd, mrd, mformat`....  
איננו משתשים בפקודות `mtool` באופן יומיומי--פעולות על קבצי DOS/MS  
windows יכולות להתבצע על ידי שימוש בפקודות לינוקס רגילות לאחר המרת מערכת  
הקבצים של DOS/MS windows.

`mv source destination`  
העבר קבצים או שנה את שמם. אותה הפקודה משמשת גם לקבצים וגם לתיקיות.

`rename string replacement_string filename`  
כלי גמיש לשינוי חלקים של שמות קבצים. לדוגמא:  
`rename *.htm *.html`

`ln source destination`  
צור לינק קשיח (hard link) שנקרא `destination` אל הקובץ שנקרא `source`. הלינק מופיע  
כהעתק של הקבצים המקוריים, אבל במציאות רק העתק אחד של הקובץ נשמר באמת, בעוד  
שתי מבואות ספרייה (או יותר) מצביעות אליו. כל השינויים לקובץ שקופים ללינק. כשמבואת  
ספרייה אחת מוסרת, האחרות נשארות שלמות. המגבלות של לינקים קשיחים הן: הקבצים  
צריכים להיות על גבי אותה מערכת קבצים, לינקים קשיחים לספריות או אל קבצים מיוחדים  
אינם אפשריים.

`ln -s source destination`  
צור לינק סימבולי (רך) שנקרא `destination` אל הקובץ שנקרא `source`. הלינק הסימבולי  
בסך הכל מציין נתיב אל הקובץ ה"אמיתי". בניגוד ללינקים קשיחים, המקור והיעד אינם  
חייבים להיות על גבי אותה מערכת קבצים. בהשוואה ללינקים קשיחים, החסרונות של  
לינקים סימבוליים הם: אם הקובץ המקורי מוסר, הלינק "שבור"--הוא מצביע אל שום מקום;  
לינקים סימבוליים יכולים ליצור הפניות מעגליות (כמו בגליון נתונים או במסד נתונים, ז"א  
"a" מצביע אל "b" ו "b" מצביע אל "a" בחזרה) בקיצור, לינקים סימבוליים הם כלי נהדר  
ומאוד נפוצים (נפוצים יותר מלינקים קשיחים), אבל הם יכולים ליצור רמה נוספת של  
מורכבות.

`rm files`  
הסר (מחק) קבצים. אתה חייב להיות הבעלים של הקובץ כדי להסיר אותו (או להיות root).  
במערכות רבות, אתה תיתבקש לאשר את המחיקה; אם אינך מעוניין בכך, השתמש ב  
אופציה "-f" (force=). משהו כמו `rm -f *` יסיר את כל הקבצים בספריית העבודה  
הנוכחית שלי, ללא שאלות.

`mkdir directory`  
צור ספרייה חדשה.

`rmdir directory`  
הסר ספרייה ריקה.

`rm -r files`  
(הסר רקורסיבית) הסר קבצים, ספריות, ואת תת הספריות שלהן. הזהר עם פקודה זו

כשאתה root--אתה יכול להסיר בקלות את כל הקבצים שבמערכת עם פקודה כזו מבוצעת בראש עץ הספריות שלך, ואין שיחזור מחיקה בלינוקס (עדיין). אבל אם אתה באמת מעוניין לעשות זאת (שקול זאת שוב), הנה כיצד (כ root):

```
rm -rf /*
```

```
rm -fr files
```

(recursive force remove). כמו הנ"ל, אבל מדלג על הבקשה לאישור, אם זו מוגדרת במערכת שלך. היזהר עם פקודה זו במיוחד כשאתה root--ראה למעלה.

```
mc
```

שגר את מנהל הקבצים "midnight commander" (נראה כמו "norton commander" ללינוקס). על פי כמה דינוזאורי מחשבים, זהו מנהל הקבצים הטוב ביותר שהיה אי פעם.

```
konqueror &
```

(בטרמינל X) שגר את מנהל הקבצים של KDE. ייתכן וזהו מנהל הקבצים האולטימטיבי. הרבה יותר טוב מ"windows explorer" של מייקרוסופט. מוטמעת בו צפייה באינטרנט, כמו גם צפייה בקבצי pdf ועוד. באמת cool.

```
xwc
```

(בטרמינל X). עוד מנהל קבצים מצויין (נקרא "X win commander"). מהיר יותר מkonqueror, אבל לא עמוס בתכונות כמוהו.

```
nautilus &
```

(בטרמינל X). מנהל קבצים cool באמת. איטי יותר מאשר konqueror, אבל מציע לי הרבה דברים טובים כמו צפייה מוקדמת דרך אייקונים בתוכן של הקבצים. הוא אפילו מציג את התוכן של קבצי קול! הוא מהיר, הוא רץ נהדר על ה-GHZ 1.33 שלי, אבל אינני משתמש בו במחשב ה-MHZ133 שיש לי.

## 5.6 צפייה ועריכת קבצים

```
cat filename | more
```

צפה בתוכן של קובץ טקסט שנקרא "filename", עמוד אחד בכל פעם. ה"|" הוא סמל ה"pipe" (במקלדות אמרקאיות רבות הוא חולק את מקומו עם ה"\"). more גורם לפלט לעצור לאחר כל מסך מלא. לקבצים ארוכים, נוח יותר לעיתים להשתמש בפקודות head ו-tail אשר מציגות רק את ההתחלה והסוף של הקובץ, או בפקודה less שמאפשרת לגלול למעלה ולמטה. אם במקרה השתמשת בcat כדי להציג קובץ בינארי והטרמינל שלך מציג תווים מוזרים, אתה יכול לשחזר(לבדוק) אותו בעזרת הפקודה reset.

```
cat filename | less
less filename
```

(שתי פקודות השתמש השתמש בכל אחת מהן) גלול תוכן של קובץ טקסט. הקלד על q כשסיימת. "less" היא כמעט מקבילה ל"more", הפקודה שאתה מכיר מדוס, אבל "less" נוחה יותר לשימוש מכיון שהיא מאפשרת לך לגלול למעלה ולמטה.

```
head filename
```

הדפס לפחות 10 שורות מתוך קובץ טקסט ארוך או מתוך קובץ שגדל. השתמש ב tail -f filename כדי tail יעקוב אחרי קובץ בזמן שהוא גדל--שימושי מאוד לבחינה מתמשכת של קבצי לוג.



pico filename  
עורך קובץ טקסט על ידי עורך טקסט פשוט וסטנדרטי שנקרא pico. השתמש ב <Ctrl> כדי לצאת. ישנם עורכי טקסט רבים ללינוקס, כולל כמה אשר מבוססים על ממשק גרפי. nano הוא עורך טקסט תואם pico.

pico -w filename  
עורך קובץ טקסט, ללא סידור גלישת שורות (wrap). שימושי לעריכת קבצי קונפיגורציה, כמו /etc/fstab.

kwrite  
(בטרמינל X) נחמד מאוד, תומך בבחירת טקסט אנכי.

kate  
kedit  
gedit  
(בטרמינל X). עורכי טקסט מבוססי משק גרפי.

gxdit  
(בטרמינל X) עוד עורך טקסט בעל יכולות מגוונות. כולל גיבוי מתוזמן.

latte  
(בטרמינל X) עורך "code", עורך טקסט שנועד לכתיבת תוכניות.

nedit  
(בטרמינל X) עוד עורך תיכנות. נחמד מאוד ועמוס.

bluefish  
(בטרמינל X) עורך html (מקור עם הדגשת תחביר, כולל המון כלים ואפשרויות).

ispell filename  
בודק איות בקובץ טקסט ASCII. AbiWord, WordPerfect, StarOffice ומעבדי תמלילים אחרים מגיעים עם בדיקת איות "בזמן שאתה מקליד", לכן אין צורך שתדאג לגבי ispell אלא אם כן אתה זקוק לו. הפצות לינוקס חדשות יותר מכילות מודול בדיקת איות שנקרא aspell, אבל, הפקודה הנ"ל עדיין עובדת.

look thermo  
חפש במילון שמערכת שלך (/usr/share/dict/words) מילים שמתחילות ב "thermo".

wvHtml ms\_word\_document.doc > filename.html  
ממיר מסמך word לפורמט html.

## 5.7 מציאת קבצים

find / -name "filename"  
מצא את הקובץ שנקרא "filename" במערכת הקבצים שלך כשהחיפוש מתחיל בספריית השורש "/". הקובץ יכול להחיל תוים כלליים (\*,?).

הפקודה find עוצמתית מאוד. יש לה אופציות רבות שיאפשרו לך לחפש קבצים במגוון דרכים. לפי תאריך, גודל, הרשאות, בעלים... אך עדיין עריכה של כמה משאלות החיפוש יכולות לקחת לך יותר מדקה. הסתכל findb.info. לפניך כמה דוגמאות מורכבות יותר לשימוש findb כדי להשלים מספר משימות מועילות.

```
find $HOME -name core -exec rm -f {} \;
```

הפקודה למעלה מוצאת קבצים בשם core, ומתחילה בספריית הבית שלך. לכל קובץ שנמצא היא מבצעת את הפקודה "rm -f" (כופה מחיקה של הקובץ). ה { } מתפרש כקובץ שנמצא, וה"\" מסיים את רשימת הפקודות.

```
find /dev -user "peter" | more
```

פקודה זו מציגה את שם הקובץ לכל ההתקנים שנמצאים בבעלות המשתמש "peter". הצגת שם הקובץ היא פעולת ברירת המחדל של find, ככה שאין היא צריכה להיות מצויינת כאופציה אם זה כל מה שדרוש לי.

```
find /home/peter -nouser -exec ls -l {} \; -ok chown peter.peter {} \;
```

מוצאת קבצים ללא משתמש תקף בספרייה /home/peter. מציגה את הקובץ בפורמט הארוך. לאחר מכן מבקשת לשנות את הבעלות על הקובץ למשתמש "peter" ולקבוצה "peter". קרוב לוודאי שתצטרך להיות root כדי למסור בעלות על קובץ.

locate filename

מוצא את שם הקובץ שמכיל את החלק "filename". קל ומהיר יותר מהפקודה הקודמת אבל תלוי במסד הנתונים שבדרך כלל נבנה בלילה, לכן אינך יכול למצוא קובץ שזה עתה נשמר אל מערכת הקבצים. כדי לכפות עידכון מיידי על מסד הנתונים, אני יכול להשתמש (roots) ב:updatedb&

```
which executable_name
```

הצג את הנתוב המלא אל התוכנית שתרוץ אם אקליד את שמה בשורת הפקודה.

```
לדוגמא, פקודה זו:  
which netscape  
במערכת שלי מניבה את:  
/usr/bin/netscape
```

```
whereis command
```

מציג את המיקום של קיבצי הבינארי, המקור, וחוברת ההסברים של הפקודה "command".

```
rgrep -r 'celeste' . | more  
grep -r 'celeste' . | more
```

(שתי פקודות, השתמש בזו שעובדת במערכת שלך). חפש בכל הקבצים בספרייה הנוכחית ובתת הספריות שבה (האופציה "-r" מתפרשת כ"recursive") את סטרינג "celeste" שבדוגמא. הצג את שם הקובץ ואת השורה בקובץ שמכילה את את הסטרינג המבוקש.

```
kfind &
```

(בטרמינל X). ממשק גרפי לfind ולgrep. נחמד מאוד. ה& בסוף הפקודה גורם לkfind לרוץ ברקע ככה שטרמינל X נשאר זמין.

## 5.8 היסודות של X-windows

```
xinit &
```

מתחיל שרת X "נקי" (לא מנהל חלונות). ה& גורם לפקודה לרוץ ברקע.

```
startx &
```

מתחיל שרת X-windows ואת מנהל החלונות של ברירת המחדל. עובד כמו הקלדת "win" תחת דוס ב3.11.win.

`startx -- :1 &`  
התחל שיח X-windows נוסף בתצוגה 1 (שיח ברירת המחדל נפתח בתצוגה 0). אתה יכול ליצור מצב בו יהיו לך כמה טרמינלים של ממשק גרפי שירוצו במקביל. החלף ביניהם על ידי שימוש ב-`<Ctrl><Alt><F7>`, `<Ctrl><Alt><F8>`, וכו'.

`xterm`  
(בטרמינל X) הרץ טרמינל X-windows פשוט. הקלדת `exit` תסגור אותו. ישנם טרמינלים נוספים, טרמינלים "ווירטואלים" מתקדמים יותר ל-X-windows. אני אוהב את הפופלריים יותר: `konsole` ו-`kvt` (שניהם מגיעים עם `kde`) ישנו גם `gnome-terminal` (מגיע עם `gnome`). אם אתה זקוק למשהו מפואר יותר, נסה את `Eterm`. למשהו פשוט ומהיר, יכולתי לבחור ב-`rxvt`.

`startkde`  
`gnome-session`  
`xfce`  
`afterstep`  
`Anotherlevel`  
`fvwm2`  
`fvwm`  
(בטרמינל X, 7 פקודות שונות, השתמש בזו שמתחילה את מנהל החלונות המועדף עליך)  
התחל את מנהל החלונות המועדף עליך בטרמינל X או בשרת X "נקי".

## 5.9 אפליקציות רשת

`netscape &`  
(בטרמינל X) מריץ את נטסקייפ.

`netscape -display host:0.0 &`  
(בטרמינל X) הרץ את נטסקייפ במכונה הנוכחית שלי והפנה את הפלט אל מכונה בשם "host" בתצוגה 0 מסך 0. למכונה הנוכחית שלך חייבת להיות הרשאה להציג במכונה המארחת "host" (בדרך כלל ניתן על ידי ביצוע הפקודה `current_machine_name` בטרמינל X של המכונה המארחת. ניתן להריץ אפליקציות X-windows נוספות באותה דרך).

`lynx file.html`  
צפה בקובץ `html` או גלוש באינטרנט במצב טקסט. למרות המראה ונוחות השימוש של `lynx` הוא לא נח יעיל כמו דפדפן מבוסס ממשק גרפי. הוא בעל משקל קל, כמעט תמיד עובד, ואינו מצריך קונפיגורציה, כל עוד הרשת שלך פונקציונלית.

`konqueror &`  
(בטרמינל X) מנהל קבצים ודפדפן אינטרנט בכלי אחד. נחמד מאוד, טוב יותר מנטסקייפ במובנים רבים יותר.  
מגיע עם KDE.

`pine`  
קורא דואר במצב טקסט. קורא דואר נוסף הוא `elm`. קליינט הדואר של נטסקייפ יקרא דואר

מחשבון האינטרנט שלך. pine יאפשר לך לקרוא את הדואר "המקומי" שלך-הדואר שהבן שלך או תהליך cron שולחים לך ממחשב ברשת הביתית שלך. הפקודה mail יכולה גם כן לשמש לקריאת/כתיבת דואר, אבל היא תהיה לא כל כך נוחה-- היא נועדה לשימוש בסקריפטים לאוטומציה.

mutt

קורא דואר בסיסי מאוד אך שימושי בצורה קיצונית וגם מהיר מאוד.

kmail &

(בטרמינל X), תוכנית דואר נחמדה, בעלת ממשק גרפי. אני משתמש בkmail, היא הרבה יותר טובה מהתוכנה המקבילה של נטסקייפ. אני יכול להיות הבעלים של כמה חשבונות ולמשוך דואר משרת smtp (מקומי) ומשרתי pop3 (מבוססי ספק אינטרנט) אל אותה התיבה. פשוט ואלגנטי. תומך בחתימות דיגיטליות.

licq &

(בטרמינל X) קליינט icq. אחד נוסף וטוב הוא kxicq. להפצות ישנות יותר אין קליינט icq מותקן, ייתכן ויהיה עליך להוריד אחד ולהתקין אותו.

knobe &

(בטרמינל X) מתחיל את קורא קבוצות הדיון (usenet) המועדף עליי. הוא הרבה יותר טוב מזה שמובנה אל תוך נטסקייפ.

talk username1

דבר עם משתמש אחר שמחובר כרגע למכונה שלך (או השתמש ב"talk" username1@machinename" כדי לדבר עם משתמש ממחשב אחר). כדי לקבל את ההזמנה לשיחה, הקלד את הפקודה "talk username2". אם מישהו מנסה לדבר איתך והעיניין מפריע לעבודך, אתה יכול להשתמש בפקודה "mseg n" כדי לסרב לקבלת הודעות. ייתכן ותירצה להשתמש ב"who" או ב"rwho" כדי לקבוע מי הם המשתמשים שמחוברים כרגע. talk הוא אחד מהכלים ה"סטנדרטיים" הישנים של UNIX, ועדיין, זה יכול להיות cool שימושי בסיטואציות מסוימות.

telnet server

התחבר למכונה אחרת על ידי שימוש בפרוטוקול TELNET. השתמש בשם או בכתובת ip של מכונה מרוחקת. אתה תיתבקש להקליד שם לוגאין וסיסמא--אתה חייב להיות בעל חשבון במכונה המרוחקת כדי להיכנס. טלנט יחבר אותך אל המכונה המרוחקת ויאפשר לך לתפעל אותה כאילו אתה יושב אל מול המקלדת שלה (כמעט). טלנט איננו מאובטח מדי--כל מה שאתה מקליד נע כטקסט פתוח דרך רשתות, אפילו הסיסמא שלך! מנהל רשת מוכשר יכול לקרוא "על הדרך" את מה שהקלדת. השתמש בssh (דורש קצת הגדרות) בשביל תשדורת מוצפנת.

rlogin server

(remote login=) התחבר למכונה אחרת. שם הלוגאין/הסיסמא מהשיח הנוכחי שלך ישמשו לאוטנטיקציה, במידה והם יידחו, תתבקש להקליד אותם.

rsh server

(remote shell=) דרך נוספת להתחבר אל מכונה מרוחקת. שם הלוגאין/הסיסמא מהשיח הנוכחי שלך ישמשו לאוטנטיקציה, במידה והם יידחו, תתבקש להקליד אותם.

ssh servername -l username  
(secure shell) התחבר לשרת (remote login) על ידי שימוש בקישור מאובטח. ssh  
מאובטח מכיון שנוא מצפין את כל הנתונים שמועברים אל הרשת על ידי שימוש בזוג  
"מפתחות-RSA-פרטיים". במידה ולא ציינת שם משתמש, שם המשתמש הנוכחי שלך יהיה  
בשימוש.

גם הקליינט וגם השרת צריכים להריץ את השירות (daemon) ssh. הם בדרך כלל זמינים בהפצות לינוקס  
חדשות (7.0 RH ואילך). לפני שתשתמש בssh, כמה הגדרות תצורה תהינה הכרחיות. המשתמש יוצר את זוג  
מפתחות RSA שלו (למטרת הצפנה) על ידי הרצת הפקודה ssh-keygen. זה מאחסן את המפתח הפרטי  
בקובץ \$HOME/.ssh/identity.pub ואת המפתח הציבורי ב-\$HOME/.ssh/identity.pub בספריית הבית של  
המשתמש. כדי לאפשר לוגאין אוטומטי, על המשתמש להעתיק את identity.pub אל  
\$HOME/.ssh/authorized\_keys שבספריית הבית שיש לו על גבי המכונה המרוחקת. לאחר מכן,  
המשתמש יכול לבצע לוגאין ללא הצורך לתת את הסיסמא. ייתכן והדרך הנוחה ביותר לשימוש באוטנטיקציית  
RSA תתבצע בעזרת סוכן אוטנטיקציה. הסתכל ב man | ssh-agent למידע נוסף. במידה והאוטנטיקציה  
האוטומטית נכשלת, ssh מבקש שם משתמש וסיסמא. הסיסמא נשלחת אל המארז המרוחק לבדיקה; בכל  
מקרה, מכיון שכל התקשורת מוצפנת, הסיסמא לא יכולה להיחשף על ידי מישהו שמצוטט לרשת.

ftp server  
פתיחת שיח ftp עם מכונה אחרת. (ישנו גם ncftp שמוסיף תכונות נוספות וגם gftp לממשק  
גרפי.) ftp טוב להעתקת קבצים מ/אל מכונה מרוחקת. נסה את המשתמש anonymous אם  
אין לך חשבון על השרת המרוחק. לאחר יצרת הקשר, השתמש ב"?" כדי להציג את רשימת  
פקודות ftp האפשריות. פקודות ftp העיקריות הן: ls (מציג את הקבצים במערכת  
המרוחקת), ASCII, בינארי (קבע את מצב העברת הקבצים לטקסט או לבינארי, חשוב  
שתבחר באפשרות המתאימה), get (מעתיק קובץ מהמכונה המרוחקת אל המקומית),  
mget (העתק-get כמה קבצים בבת אחת), put (העתק קובץ מהמכונה המקומית אל  
המרוחקת), mput (העתק-put כמה קבצים בבת אחת), bye (התנתק). כדי להשיג אוטומציה  
בסקריפט, ייתכן ותרצה להשתמש בncftpput וncftpget, לדוגמא:

```
ncftpput -u my_user+name -p my_password -a remote.host.domain  
remote_dir *local.html
```

נראה של ncftp יש בעיה כשהמחשב נמצא מאחורי פיירוול--עליך לקנפג את הקובץ  
/home/usr\_name/.ncftp/firewall. כאלטרנטיבה, אתה יכול להשתמש ב"lftp" כדי להשיג  
את אותה המטרה, לדוגמא:

```
lftp -e "mput -a *local.html" -u my_user_name,my_password  
ftp://remote.host.domain
```

כדי לשמור על mirrors של ספריות ftp, אתה יכול להשתמש בfmirror.

```
wget -m --no-parent http://sunsite.dk/linux-newbie
```

מעתיק קבצים מאתרי ווב. הדוגמא שלמעלה משתמשת באופציה -m (=mirror) כדי להשיג  
סט שלם של קבצים מהאתר המסטר של מדריך זה. האופציה "--no-parent" מגבילה את  
העתקת הקבצים לאלו שניתנו בספרייה ובתת הספריות.

minicom  
תוכנה ל"terminal emulation" של חיבור סיריאלי. נראה ועובד כמו "procomm" או  
"telix". שימושית לבדיקה ודיבוג של החיבור הסיריאלי שלך.

אז  
מקבל קבצים כשמתמשים בפרוטוקולים, Ymodem, Zmodem או Xmodem.  
Xmodem דורש שם קובץ. השתמש ב --help אז למידע נוסף. מי בכלל משתמש  
בפרוטוקולים האלה היום?

"אני משתמש בזמודם באופן קבוע. יש לי שני מחשבים שמריצים לינוקס (SuSE), מחשב נייד ונייח.  
למחשב הנייח אין גישה לחיבור לאינטרנט. לכן, אני מעביר קבצים שהורדתי מהרשת ממחשב אחד לשני על ידי  
שימוש ב null-modem cable, בעזרת minicom והפרוטוקול Zmodem. בדרך זו אני אפילו יכול לחבר את  
הנייד שלי מהעבודה שמריץ win2000 אל מכונת הלינוקס שלי על ידי שימוש ב Reflexion (תוכנית אמולציית  
טרמינל של win32)"  
(from berry Vos, B.Vos@getronics.com, 2001 01 28)

## 5.10 דחיסה ופריסה של קבצים

```
tar -zxvf filename.tar.gz
```

(tape archiver=) פרוס קובץ tarball דחוס (\*.tar.gz או \*.tgz) שהורדת מהאינטרנט.

```
tar -xvf filename.tar
```

פרוס קובץ tar לא דחוס (\*.tar)

```
tar czvpf /var/backups/mybackup.tar.gz /home  
cd / : tar xzvpf /var/backups/mybackups.tar.gz ' * /myfile.rtf '
```

(כ root) צור גיבוי של /home אל קובץ דחוס. הפקודה השניה מראה כיצד לשחזר קובץ  
מגיבוי.

```
gunzip filename.gz
```

פרוס קובץ זיפ (\*.gz או \*.z). השתמש בגzip (גם zip או compress) אם אתה מעונין  
לדחוס קבצים אל פורמט זה.

```
zcat filename.gz | more
```

(zip cat=) הצג את התוכן של קובץ דחוס. זמינים גם כלים אחרים לטיפול בקבצים דחוסים  
ללא פריסה מוקדמת: zless, zmore, zgrep...

```
bunzip2 filename.bz2
```

(big unzip=) פרוס קובץ (\*.bz2) שנדחס על ידי הכלי bzip2. משמש לקבצים גדולים.

```
unzip filename.zip
```

פרוס קובץ (\*.zip) שנדחס על ידי כלי דחיסה בעל תאימות ל PKZIP לדוס.

```
zip filename.zip filename1 filename2
```

דחוס שני קבצים - filename1 filename2 - אל תוך ארכיב zip שנקרא filename.zip.

```
unarj e filename.arj
```

פרוס את התוכן של ארכיב \*.arj.

```
lha e filename.lha
```

פרוס את התוכן של ארכיב \*.lharc.

`uudecode -o outfile filename`  
פענח קובץ אשר מקודד בעזרת `uuencode`. קבצים שמקודדים על די `uuencode` משמשים בדרך כלל להעברת קבצים שאינם טקסט בדואר אלקטרוני (`uuencode` מבצע טרנספורמציה של כל קובץ אל קובץ ASCII).

`cat filename | mimecode -o filename`  
`cat filename.mime | mimecode -u -o filename`  
(2 פקודות). קודד ולאחר מכן פענח חזרה קובץ מ/אל סטנדרט מבוסס דואר אינטרנט (mail-oriented internet standard) של העברת מידע בן 7 ביט שנקרא "mime". בהפצות ישנות יותר, הפקודה שלא תעבוד (`mimecode`) נקראת `mmencode`. בדרך כלל, לא תצטרך לטרוח עם הפקודות האלו, ה"דוור" (`mailer`) שלך אמור לבצע את קידוד/פיענוח `mime` באופן שקוף.

`ar -x my_archive.a file1 file2`  
(`archiver=`) פרוס את הקבצים `file1` ו-`file2` מרכיב שנקרא `my_archive.a`. כלי הארכיב `ar` משמש בעיקר להחזקת ספריות.

`ark &`  
(בטרמינל X). אפליקצית ארכיב בעלת ממשק גרפי (Qt-based). יכול מאוד להיות שזה כל מה שאתה צריך כדי לארגן את הקבצים הדחוסים שלך. אלטרנטיבה נוספת היא `gnozip`.

## 5.11 שליטה על תהליכים

`ps`  
(`print status` או `process status`) הצג את התוכן של התהליכים שרצים כרגע ביחד עם מספרי הזיהוי שלהם (PID). השתמש ב `ps aux` כדי לראות את כל התהליכים שרצים כרגע במערכת שלך ( גם את אלו של משתמשים אחרים או אלו שללא טרמינל שליטה), כל אחד ביחד עם הבעלים שלו. השתמש ב"top" כדי להמשיך להציג את התהליכים שרצים כרגע.

`any_command &`  
הרץ כל פקודה ברקע (הסימן `&` מתפרש כ"הרץ את הפקודה המקדימה ברקע").  
ה `job_number` מוצג על גבי המסך ככה שאתה יכול להביא את הפקודה אל החזית (ראה למטה) במידה ואתה מעוניין בכך. אני משתמש ב"&" בדרך כלל כשאני מתחיל תוכנית בעלת ממשק גרפי מטרמינל X.

`jobs`  
הצג את התהליכים ששיכים לי, אשר נמצאים ברקע או כאלו שנעצרו והראה את מספר `job` שלהם.

`fg job_number`  
הבא תהליך שנעצר או תהליך שרץ ברקע חזרה לחזית.

`bg job_number`  
הנח תהליך ברקע, ממש כאילו הוא הותחל עם `&`. זה יתחיל מחדש תהליך רקע שנעצר. התהליך שנמצא כרגע בחזית יכול להיעצר בדרך כלל על ידי `<Ctrl>z`. אם עצרת או שלחת תהליכים לרקע, עליך להקליד `exit` פעמיים ברצף כדי לבצא לוג אוטו.

`batch`  
`at>updatedb<Ctrl>d`  
הרץ כל פקודה (בדרך כלל כזו שהולך לקחת לה יותר זמן להשלים) כשהעומס על המערכת נמוך. אני יכול לצאת מהמערכת, והתהליך ימשיך לרוץ. כשהפקודה תושלם, יישלח אלי מייל עם הפלט. בדוגמא למעלה ה "`at>`" מייצג שורת פקודה (prompt), הפקודה שתורץ היא `updatedb`, וה `<Ctrl><d>` משמיד את הקלט שלי אל `batch` (יכולתי להתחיל פקודות רבות להרצה, מופרדות על ידי `<Enter>`).

`at 17:00`  
מבצע פקודה בזמן מסויים. אתה תתושאל לגבי הפקודה (פקודות) שברצונך להריץ, עד שתלחץ `<Ctrl><d>`. הפקודות המקושרות הן `atq` (מציג את תור התהליכים שהותחלו עם `at`) וגם `atrm` (מסיר תהליך מהתור - "at queue").

`kill PID`  
כופה כיבוי על תהליך. לפני כן קבע, על ידי `ps`, מהו מספר התהליך (PID) שאותו אתה מעונין להרוג.

`killall program_name`  
הרוג תוכנית (תוכניות) על פי שם. למשל `killall pppd` יינתק את רשת החיוג שלך.

`nohup program_name`  
(`no hungup=`). הרץ תוכנית כך שהיא לא תושמד לאחר שאתה תצא מן המערכת. הפלט מופנה אל הקובץ `nohup.out` בספרית הבית שלך. אתה בטוח לא תרצה להריץ תוכנית אינטרקטיבית תחת `nohup`.

`xkill`  
(בטרמינל X) הרוג תוכנית מבוססת ממשק גרפי בעזרת העכבר. (הצבע עם סמן העכבר אל החלון של התהליך אותו מעונין להרוג והקלק).

`kpm`  
(בטרמינל X) מנהל תהליכים של KDE.

`lpc`  
(`root`) בדוק ונהל מדפסת (מדפסות). הקלד "?" כדי לראות רשימה של פקודות אפשריות.

`lprm job_number`  
הסר את עבודת ההדפסה `job_number` מהתור.

`nice program_name`  
הרץ את `program_name` תוך כיווןן העדיפות שלה. מכיון שהעדיפות אינה מצויינת בדוגמא זו, היא תוגדל ב10 (התהליך ירוץ לאט יותר) מערך ברירת המחדל (בדרך כלל 0). ככל שהמספר קטן יותר (הערך של `niceness` אל משתמשים אחרים במערכת), העדיפות גבוהה יותר. ערך העדיפות יכול להיות בטווח שבין 20- עד ל 19. רק `root` יכול לקבוע ערכים שליליים. השתמש ב`top` כדי להציג את העדיפויות של התהליכים הרצים.

`renice -18 PID`  
(`root`) שנה את העדיפות של תהליכים רצים למינוס 18. משתמשים רגילים יכולים לכוון



תהליכים שנמצאים בבעלותם בלבד, ולציין רק ערך גבוהה יותר מהערך הנוכחי (לגרום להם לרוץ לאט יותר). ניתן גם לבצע `u peter -10 +renice` כדי לגרום למשתמש peter להשתמש בפחות עוצמת מעבד ככה שמתמשימים אחרים לא יסבלו כשהוא יריץ משימות שדורשות כח עיבוד רב.

`<Ctrl>c, <Ctrl>z, <Ctrl>s, <Ctrl>q`  
גם הם שייכים לפרק זה אבל תוארו לפני כן. בקצרה הם מתפרשים כ: עצור את הפקודה הנוכחית, שלח את הפקודה הנוכחית אל הרקע, הפסק העברת מידע, השב העברת מידע.

`ls -l`  
(list the open files). אם אני root, כל הקבצים יוצגו. אני יכול להגביל את עצמי לקבצים שנפתחו על ידי תהליכים אשר בבעלות הקונסולה הראשונה, אם אני בוחר ב `ls -l /dev/tty1`. כדי להציג קבצי רשת בלבד (חיוני לבדיקת אבטחה), הייתי עושה משהו כמו `ls -li` (root).

`watch -n 60 my_command`  
בצע את `my_command` שוב ושוב בהפוגות של 60 שניות (ברירת המחדל היא 2 שניות).

## 5.12 פקודות ניהול בסיסיות

`su`  
(substitute user id) צא מנוקדת הנחה שהזהות היא `superuser` (=root) (אתה תתבקש להקליד סיסמא). הקלד "exit" כדי לחזור אל הלוגאין הקודם שלך. השתדל לא לעבוד על המכונה שלך כroot. החשבון root נועד לאדמיניסטרציה והפקודה `su` נועדה להקל על הגישה שלך אל חשבון הניהול כשאתה זקוק לו. אתה יכול להשתמש ב `su` כדי להניח כל זהות משתמש אחרת. `su barbara` יהפוך אותי ל"barbara" (דרושה סיסמא, אלא אם כן אני הוא `superuser`).

`alias ls="ls --color=tty"`  
צור שם נוסף לפקודה "ls" כדי לשפר את הפורמט שלה עם צבע. בפקודה זו, השם הנוסף נקרא גם כן "ls" והאופציה "color" מתעוררת רק כשהפלט נעשה אל טרמינל (לא אל קבצים). הנח את `alias` בתוך הקובץ `/etc/bashrc` אם אתה מעוניין שהוא יהיה נגיש תמיד לכל המשתמשים במערכת. שמות נוספים (`aliases`) הינם דרך שימושית לבצע קוסמיזציה למערכת שלך. הקלד "alias" לבד כדי לצפות ברשימה של `aliases` בחשבון שלך. השתמש ב `unalias alias_name` כדי להסיר `alias`.

`cat /var/log/httpd/access_log`  
הראה מי התחבר אל שרת הווב שלך (`apache`) מאז הפעם האחרונה שקובץ הרישום ביצע רוטציה (בדרך כלל פעם ביום, כש `cron` רץ). קובץ הרישום הקודם הוא `access_log.1`, וזה שלפניו הוא `access_log.2` וכו'.

`cat /var/log/secure`  
(כ root) בחן את רישום המערכת החשוב. זה באמת רעיון טוב לעשות את זה מדי פעם אם יש לך גישה לאינטרנט.

`ftpwho`  
(כ root) הצג את מי שמחובר כרגע לשרת `ftp` שלך.

printtool  
(כ root בטרימינל X) כלי קביעת התצורה למדפסת (מדפסות) שלך. ההגדרות נשלחות אל הקובץ /etc/printcap ואל (באופן די תמוהה) הקובץ /var/spool/lpd.

setup  
(כ root) קבע את הגדרות העכבר, כרטיס הקול, מקלדת, X-windows, ושירותי המערכת. ישנם המון כלי קונפיגורציה ספציפיים להפצות שונות, setup הוא ברירת המחדל בREDHAT 7.0. מנדרייק 7.0 מציעה הרבה כלי Drakconf.

linuxconf  
(כ root, בטרימינל טקסט או בX). מאפשר לך לגשת ולשנות מאות הגדרות רשת. עוצמתי מאוד--אל תשנה יותר מדי דברים בבת אחת, והיה זהיר עם שינוי הגדרות שאינך מבין. כלי הגדרת תצורת הרשת של REDHAT, netconf, הוא תת חלק של linuxconf, לכן לעיתים הוא נח יותר לשימוש.

mouseconf  
(כ root). כלי פשוט להגדרת התצורה של העכבר שלך (לאחר ההתקנה הראשונית). מנדרייק כוללת כלי אלטרנטיבי בשם mousedrak.

kudzu  
(כ root). קובע ומגדיר באופן אוטומטי את החומרה שלך. אם אתה נתקל בבעיות מיסתוריות עם העכבר שלך (או כל חומרה סיריאלית אחרת), ייתכן ותרצה לבטל את kudzu, ככה שהוא לא ירוץ עם איתחול המערכת (kudzu בילגן לי את המערכת ככה שאפילו בעכבר לא יכולתי להשתמש). אתה יכול להריץ אותו באופן ידני כשאתה זקוק לו.

timeconfig  
(כ root) קבע את איזור הזמן במערכת שלך. שעון החומרה במערכת שלי (זה שבביוס) שומר את הזמן ב UTC (Coordinated Universal Time), מה שנקרא פעם GMT או (Greenwich Mean Time). בדרך זו, אני נמנע מבעיות אפשריות שקשורות להחלפת איזורי זמן עקב זמני חיסכון של אור יום, העברת קבצים סביב הגלובוס דרך הרשת, או תזוזה פיזית. במחשבים שמשים כשרתים נהוג לשמור את הזמן בUTC כדי למנוע מהזמן "ללכת אחורה" (מה שעלול לעורר מספר בעיות). חותמות זמן על קבצים נשמרות תמיד בUTC ומוצגות בזמן המקומי על ידי שימוש במידע של איזור הזמן. לדוגמא, אפליקציות רבות (קומפיילרים, מסדי נתונים) מסתמכות על היכולת להבדיל בין קובץ חדש לבין קובץ ישן יותר על ידי השוואת חותמות הזמן של הקבצים. זה חשוב לשמור על איזור הזמן מדוייק. הסיבה היחידה בגללה יכולתי לבחור בשמירה על זמן BIOS בזמן המקומי היא כדי להמנע מבעיות כשאני משתמש באיתחול כפול על גבי אותו מחשב, וגם כשמערכות הפעלה אחרות (MS ווינדוס?) לא יודעות כיצד לטפל בUTC. לכן, אני מאפשר לשרת הלינוקס שלי להיות מודע לנושא על ידי סימון הקופסא "Hardware clock to GMT", ככה שלינוקס יכולה לחשב אחורה את הUTC כשהיא צריכה.

setclock  
קבע את שעון החומרה שלך מהזמן המערכת הנוכחי של לינוקס. השתמש בפקודה "date" לפני כן כדי להגדיר את זמן המערכת של לינוקס. יכולתי לשנות את התאריך והשעה ל:  
2000-12-31 23:57 על ידי שימוש בפקודה זו:

```
date 123123572000
```

ואז לכתוב את הזמן אל שעון החומרה על ידי:

```
setclock
```

`dateconfig&`  
(בטרמינל X, כ `root` או שתישאל לסימא של `root`). כלי ממשק גרפי מצויין כדי להגדיר את שעון החומרה, את שעון מערכת ההפעלה שלי ואת איזור הזמן, ובנוסף להגיד לביוס שלי לשמור את הזמן בUTC. אין לי צורך בשתי הפקודות הקודמות.

`xvidtune`  
(בטרמינל X). כוון את ההגדרות של תצוגת המסך שלך לכל הרזולוציות כמו גם השמדת פסים שחורים, החלף את התצוגה ימינה/שמאלה/למטה וכו'. (לפני כן השתמש במכוונים שעל המסך שלך כדי להתאים את תצוגת מצב הטקסט בצורה תקינה אל המסך שלך). לאחר מכן השתמש ב `xvidtune` כדי לכוון את תדירויות המסך שלך לכל רזולוציה ככה שהיא תתאים היטב אל המסך שלך. כדי לגרום לשינויים להפוך לקבועים, הצג את התדירויות על המסך שלך ואז העבר אותן אל קובץ ההגדרה `-etc/X11/XF86config`. במסכים חדשים יותר, ייתכן ותעדיף לכוון את המסך שלך על ידי ההגדרות שמובנות אל תוך המסך--`xvdtune` נועד למסכים ישנים שנעדרים את היכולת לזכור את ההגדרות של עצמם.

`kvideogen`  
(בטרמינל X). ייצר "modelines" לרזולוציה מותאמת אישית של המסך שלך. לאחר שיצרת את טקסט ההגדרה (ה "modelines"), אתה יכול להעתיק ולהדביק אותו אל תוך קובץ ההגדרה של `X-windows - etc/X11/XF86Config` (או `/etc/X11/XF86Config-4`) אם אתה משתמש בשרת X גירסה 4.x). ראה גם את קיצור הדרך `<Ctrl><Alt><+>`

`SVGATextMode 80x25x9`  
`SVGATextMode 80x29x9`  
(כ `root`) שנה את הרזולוציה בטרמינל טקסט. בדוגמא למעלה (שורה שניה) שיניתי את מסך הטקסט ל80 עמודות x 29 שורות עם תוים בגודל 9 פיקסלים. השורה הראשונה מגדירה רזולוציה שתמיד עובדת, ככה שאם הפקודה השניה לא תעבוד במערכת שלי, אני יכול להקליק על חץ למעלה פעמיים ו `<Enter>` כדי להשיב לעצמי את השליטה על המסך שלי. המצבים השונים תלויים כרטיס המסך שלך ובתדירויות הסינכרון של המסך שלך--הייתי צריך לערוך (כ `root`) את הקובץ `/etc/TextConfig` ולסמן (להסיר סימון) את השורות המתאימות כדי לאפשר ל `SVGATextMode` לדעת במה המערכת שלי תומכת.

`SuperProbe`  
(כ `root`). כלי לקביעת סוג כרטיס המסך וכמות הזיכרון שלו.

`cat /var/log/XFree86.0.log`  
קובץ רישום ל X שיכול להועיל בנסיון לקבוע מה לא תקין בתצורת ה X שלך. ה"0" בשם הקובץ מתפרש כ"display 0"--שנה את שם הקובץ בהתאמה אם אתה זקוק לרישום לגבי תצוגה "1", "2" וכו'.

`lspci`  
הצג מידע לגבי לוח האם שלך ואילו כרטיסים יושבים בחריצי ההרחבה של `pci`. למחשבים הישנים שלי יש חריצי ISA (או EISA), ולא `pci`.

`lsdev`  
מציג מידע לגבי החומרה שלך (`IO ports`, `IRQ`, `DMA`).

`lsdf | more`  
הצג רשימה של קבצים פתוחים במערכת שלך.

kerneklcfg

(כ root בטרמינל X). ממשק גרפי להוספה הסרה של מודולי קרנל (modules). מודול זה משהו כמו מנהל התקן--חלק מהקרנל של לינוקס שמספק תמיכה לחלק מסויים של חומרה או לחלק של פונקציונליות מסויימת. אתה יכול לעשות את אותו הדבר משורת הפקודה על ידי שימוש בפקודה insmod.

lsmod

(=list modules). הצג רשימה של מודולי קרנל שטעונים כרגע. מודול זה משהו כמו מנהל התקן--הוא מספק לקרנל של מערכת ההפעלה תמיכה לחלק מסויים של חומרה או תכונה.

modprobe -l | more

הצג את כל המודולים שזמינים לקרנל שלך. המודולים הזמינים נקבעים על ידי האופן בו הקרנל של הלינוקס שלך קומפל. כמעט כל מודול/פיצ'ר שקיים ניתן לקימפול תחת לינוקס כ"hard wired" (אולי מהיר יותר, אבל לא ניתן להסרה), כ"module" (אולי קצת איטי, אבל ניתן לטעינה/הסרה לפי דרישה), או כ"stock" (ללא תמיכה לפיצ'ר זה כלל). המודולים בהם הקרנל שלך תומך (שאיתם הוא קומפל) הינם קבצים תחת הספרייה /lib/modules (ותת הספריות שלה) לכן צפייה בספרייה זו יכולה לתת לך רמז במידה והלכת לאיבוד. אם הקרנל שלך לא תומך במודול שלו אתה זקוק, ייתכן ותצטרך לקמפל מחדש את הקרנל כשהמודול הזה מאופשר (זה די נדיר מכיון שה"stock" של הקרנל של לינוקס שמגיע עם רדהאט או מנדרייק בא מקומפל מראש עם כמעט כל המודולים המצויים והלא נסיוניים. בכל מקרה, אם יש לך חומרה ממש חדשה...)

modprobe sb

טען את המודול של soundblaster (sb). השתמש בפקודה הקודמת כדי למצוא מודולי קרנל אחרים שניתנים לטעינה.

insmod parport

insmod ppa

(כ root) הכנס מודולים אל תוך הקרנל (מודול הוא באופן גס המקבילה של מנהל התקן מדוס). בדרך כלל, אני משתמש ב"modprobe" (ראה את הפקודה הקודמת) כדי להכניס מודולים. דוגמא זו מראה כיצד להכניס מודולים לתמיכה בחיבור מקבילי חיצוני לזיפ של 100-MB. בשביל כונן זיפ חיצוני של 250-MB, אני משתמש במודול immm במקום ppa.

rmmod module\_name

(כ root, לא הכרחי). הסר את המודול module\_name מהקרנל.

dpm -a

(כ root) בנה את טבלת התלות (dependency) של המודול אל הקרנל. לא הכרחי אלא אם כן שינית את /etc/modoules ואינך מעוניין לאתחל את המחשב מחדש.

seterial /dev/cua0 port 0x03f8 irq 4

(כ root) הגדר פורט סיריאלי לתצורה לא סטנדרטית. הדוגמא כאן מראה את ההגדרה הסטנדרטית לחיבור הסיריאלי הראשון (cua או ttyS0). הגדרות pc הסטנדרטיות לחיבור הסיריאלי השני (cua1 או tty2) הינן: הכתובת של פורט i/o 3 irq, 0x02f8. פורט החיבור הסיריאלי השלישי (cua2 או ttyS2): 4 irq, 0x03e8. פורט החיבור הסיריאלי הרביעי (cua2 או ttyS3): 3 irq, 0x02e8. הוסף את ההגדרה שלך אל /etc/rc.d/rc.local אם אתה מעוניין שהיא תיקבע בזמן האיתחול. הסתכל ב-man setserial לסקירה טובה על הנושא.

tunelp  
(כ root, בשימוש לעיתים נדירות) כוונן את החיבורים הפרללים שלך.

```
/sbin/chkconfig --level 123456 kudzu off
```

(כ root) כלי לבדיקה/איפשור/ביטול שירותי מערכת שיאותחלו אוטומטית תחת runlevels שונים. בדרך כלל, אני משתמש רק בכלי ntsysv של REDAHT אם אני צריך לאפשר/לבטל שירות runlevel הנוכחי, אבל chkconfig אכן מקנה לי גמישות נוספת. כלי חליפי הוא tksysv (מבוסס X). הדוגמא למעלה מראה איך לבטל את השירות kudzu ככה שהוא לא יתחיל בכל runlevel (הוא מבלגן את תצורת העכבר באחד מהמחשבים שלי). כדי להציג את כל השירותים אשר מופעלים/מופסקים תחת כל runlevels, אני משתמש ב:

```
chkconfig --list | more
```

כדי לבדוק את המצב הנוכחי של שירותים, אני יכול להשתמש ב:

```
service --status-all
```

כדי להתחיל שירות עכשיו, אני יכול להשתמש במשהו כמו (מתחיל שרת ftp):

```
service wu-ftpd start
```

כדי להתחיל מחדש את samba (לאחר ששינתי את הקונפיגורציה שלה), אני יכול להשתמש ב:

```
service smb restart
```

```
symlinks -r -c /usr
```

(כ root) בדוק ותקן לינקים סימבוליים במערכת שלי. התחל ב / והתקדם דרך כל תתי-התיקיות (האופציה -r = "rescue") ושנה לינקים מיושנים/מבולגנים חזרה ליחסיים, מחק לינקים מתנדנדים, וקצר לינקים ארוכים (האופציות -c). במידה ומערכת הקבצים שלי מתפשטת על גבי מחיצות דיסק שונות, עליי להריץ את הפקודה הזו שוב לכל אחת מהן (symlinks -r -c /usr).

```
cd /usr/src/linux-2.4.7-10
make xconfig
```

(כ root בטרמינל X). ממשק גרפי נחמד לקינפוג אופציות הקרנל כהכנה לקימפול קרנל שלך שעבר קוסטמיזציה. (שם הספרייה בדוגמא זו מכיל את הגירסא של קרנל הלינוקס ככה שאתה יכול לשנות את שם הספרייה אם שם הקרנל שלך שונה מ-2.4.7-10 שבדוגמא זו. אתה זקוק למפענח "Tk" כדי להריץ את "make xconfig", וקרנל המקור מותקן.) האלטרנטיבות ל"make xconfig" הינן: "make config", (מריץ סקריפטים ששואלים אותך שאלות במצב טקסט) וגם "make menuconfig" (מריץ כלי קונפיגורציה מבוסס מצב טקסט). נסה את: less /usr/doc/HOWTO/Kernel-HOWTO למידע נוסף. לאחר שהגדרתי את האופציות לקרנל החדש בעזרת "make xconfig", אני יכול להמשיך בקימפול הקרנל החדש על ידי הזנת הפקודות הבאות:

```
make clean
```

(אופציונלי; זה מנקה קבצי אובייקט ישנים, עלול להאריך את הקימפול, ועלול למנוע בעיות במצבים שונים)

```
make dep
make bzImage
```

לפקודה האחרונה ייקח קצת זמן לסיים את פעולתה (אולי 10 דקות או שעתיים, תלוי בחומרה שלך). היא מפיקה את הקובץ `arch/386/boot/bzImage`, שהוא למעשה קרנל הלינוקס החדש שלך. לאחר מכן:

```
make modules
make modules_install
```

כעת המודולים החדשים מותקנים ב `./lib/modules/kernelname`

אל תשנה את שם ספריית המודול אם ברצונך להריץ כמה קרנלים--הקרנל חייב להיות מסוגל למצוא את המודולים ה"תואמים" שלו. אם אני מעוניין לשנות את שם הקרנל שלי, עליי לערוך את `makefile` הראשי של הקרנל (`/usr/src/linux-2.2.14/makefile`) ולשנות את השורות שמופיעות בחלק העליון. אצלי (RH 7.2) הן:

```
VERSION = 2
PATCHLEVEL = 4
SUBLEVEL = 7
EXTRAVERSION = -10custom
```

ניתן להציג את שם הקרנל לקרנל שרץ כרגע על ידי שימוש ב `uname -r` שלי (ברירת המחדל של RH 7.2) הוא "2.4.7-10custom".

כעת אני יכול להתקין את הקרנל החדש. ההתקנה מערבת העתקת הקרנל החדש (כשמשנים את שמו) אל תוך ספריית ה `/boot`:

```
cp arch/386/boot/bzImage /boot/vmlinuz-2.4.7-10custom
cp System.map /boot/System.map-2.4.7-10custom
```

ויוצתרים שינויים ל `/etc/lilo.conf` או ל `/boot/grub/grub.conf` ככה שאני אוכל לבחור איזה קרנל לאתחל בזמן האיתחול (את החדש או הישן). ממולץ מאוד שתשמור את הקרנל הישן שלך כאופציית איתחול (למקרה שהקרנל החדש יסרב לאתחל). אם אתה משתמש ב `initrd` (initial ram disk) לאיתחול בעל שני שלבים, תצטרך ליצור אימיג' נוסף עם מודולים שבשימוש הקרנל בזמן באיתחול:

```
mkinitrd /boot/initrd-2.4.7-10custom.img 2.4.7-custom
```

הסתכל בזה לפרטים לגבי `kernel patching`:

```
patch -E -p1 < /home/download/the_patch_to_apply
```

ייתכן ויהיה זה מועיל לקרוא גם את: `/usr/doc/HOWTO/Kernel-HOWTO` ואולי גם את `man depmod`.

קינפוג, קימפול והתקנה של קרנל חדש זה ענין די פשוט אבל יכול להוביל לבעיות. קימפול הקרנל יכול הינו גם דרך טובה לבדיקת החומרה שלך, מכיון שהוא מערב כמות נכבדת של מיחשוב. אם החומרה שלך "רעועה", יכול להיות שתקבל את הודעת השגיאה "signal 11" (לכן קרא את `/usr/doc/FAQ/txt/GCC-SIG11-FAQ`).

ldconfig  
(כ root) צור מחדש את ה bindings ו cache לטוען של הספריות הדינמיות ("ld"). ייתכן ותרצה להריץ את ldconfig לאחר התקנה של ספריות מקושרות דינמית במערכת שלך. (הפקודה מורצת מחדש בכל פעם שאתה מאתחל את המחשב, ככה שאם איתחלת אותו אין צורך להריץ את הפקודה באופן ידני.)

```
mknod /dev/fd0 b 2 0
```

(כ root) צור קובץ התקן באופן ידני. דוגמא זו מראה כיצד ליצור קובץ התקן שמקושר אל כונן הפלופי הראשון שלך ויכול להועיל אם מחקת אותו בטעות. האופציות הן: p=FIFO device, c=character mode device, b=block mode device, u=unbuffered character mode device. שתי הספרות מציינות את מספר ההתקן המינורי והמז'ורי. בדרך כלל לא הייתי יודע אילו פרמטרים knod דורש. לכן, כדי ליצור את ההתקנים, הייתי קורא קודם כל קורא את man MAKEDEV כדי לפענח את שמו של ההתקן ולאחר מכן מריץ את הסקריפט /dev/MAKEDEV שמודע לגבי התקני לינוקס לפי שם--ראה את הפקודה הבאה. במידה ועמוד החמ לא עוזר, אני יכול לפנות אל התייעוד האולטימטיבי שכלול בקוד המקור של הקרנל:

```
less /usr/src/linux/Documentation/devices.txt.
```

```
cd /dev  
./MAKEDEV audio
```

(כ root). שחזר את התקן ה"audio" שבדיק קילקלתי איך שהוא. בנוסף ראה את הפקודה הקודמת.

### 5.13 כלי ניהול דיסק

```
fdisk /dev/hda
```

(כ root, "fixed disk", כ root). כלי החלוקה למחיצות של לינוקס (לדוס יש כלי בעל שם זהה). בדוגמא שלמעלה, ציינתי שהייתי מעוניין לחלק למחיצות את הדיסק הקשיח הראשון בממשק ה IDE הראשון, לכן, hda. אם אני הייתי אתה, הייתי מגבה נתונים חשובים לפני שימוש ב fdisk על כל מחיצה. אינני מכיר מישהו שאוהב את fdisk (במהדורת לינוקס או במהדורת דוס)--אני מעדיף להשתמש ב cfdisk, ראה את הפקודה הבאה.

```
cfdisk /dev/hda
```

(כ root) כלי לחלוקה למחיצות, מבוסס תפריט. נח יותר לשימוש מאשר fdisk. (ראה את הפקודה הקודמת). כוננים פיזיים יכולים להכיל מחיצות ראשיות (4 לכל כונן מקסימום), ומחיצות לוגיות (ללא הגבלה על הכמות). מחיצה ראשית יכולה להיות ניתנת לאיתחול. מחיצות לוגיות חייבות להימצא בתוך מחיצות "extended": מחיצות "extended" לא שימושיות כשלעצמן, הן רק מיכל למחיצות הלוגיות. כשמחלקים דיסק למחיצות, אני בדרך כלל: (1) יוצר מחיצה ראשית (primary), (2) הופך את המחיצה הראשית לברת איתחול, (3) יוצר מחיצה מורחבת, (4) יוצר מחיצה (מחיצות) לוגית בתוך המחיצה המורחבת.

```
sfdisk -l -x | more
```

(כ root) הצג את רשימת המחיצות (כולל מחיצות מורחבות) לכל הכוננים שבמערכת שלי.

```
parted /dev/hda
```

כלי לניהול מחיצות ללינוקס (ext2) ודוס (FAT & FAT32). הוא משמש ליצירה, השמדה,

הזזה, העתקה, כיווץ, והרחבה של מחיצות. עליך באמת לאהוב לגבות את הנתונים שלך  
בזהירות לקרוא את info parted לפני שאתה משתמש בו.

```
fdformat /dev/fd0H1440
mkfs -c -t ext2 /dev/fd0
```

(floppy disk format=) שתי פקודות, כ root) בצע פורמט low-level של פלופי שבכונן  
הפלופי הראשון (/dev/fd0), high density (1440 kB). לאחר מכן צור מערכת קבצים של  
לינוקס (-t ext2).

בדוק/סמן בלוקים פגומים (-c). יצירת מערכת הקבצים היא המקבילה לפורמט high level.  
בנוסף אני יכול לפרמט פלופי destinies אחרים (גם כן לא סטנדרטיים); נסה את  
<Tab>/dev/fd0/.ls. אני יכול לפרמט גם אל densities הדיפולטי (באופן נורמלי k 1440)  
על ידי שימוש ב fdformat /dev/fd0.

```
badblocks /devfd01440 1440
```

(כ root) בדוק בלוקים פגומים בפלופי high-density והצג את התוצאות על גבי המסך.  
הפרמטר "1440" מציין את מספר הבלוקים שיש לבדוק. פקודה זו לא משנה את הפלופי.  
badblocks יכול לשמש גם לבדיקת פני השטח של דיסק קשיח אבל עלי לעשות unmount  
למערכת הקבצים לפני כן אם ברצוני בבדיקה מלאה של קריאה כתיבה:

mount

[על מנת לבדוק איזה התקן מכיל את מחיצת הדיסק שבו אני מעוניין לבדוק בלוקים פגומים]

```
unmount /dev/hda8
```

[בצע unmount למחיצה הנבחרת]

```
badblocks -n /dev/hda8
```

[בדוק מצב קריאה כתיבה באופן לא הרסני את המחיצה הנבחרת ככה שנתונים לא יימחקו]

```
mount /dev/hda8
```

[המר את המחיצה חזרה מכיון שלא הודפס מידע לגבי בלוקים פגומים]

במידה ונמצאו בלוקים פגומים, הם ניתנים לסימון על גבי הדיסק הקשיח ככה שהם לא יהיו  
בשימוש:

```
e2fsck -c /dev/hda8
```

```
fsck -t ext2 /dev/hda2
```

(file system check=, כ root) בדוק ותקן מערכת קבצים, לאחר כיבוי "לא נקי" כתוצאה  
מכשל באספקת החשמל. הדוגמא שלמעלה מבצעת את הבדיקה על גבי המחיצה hda2,  
מערכת קבצים מסוג ext2. אתה בהחלט תרצה לעשות unmount למחיצות או לאתחל את  
לינוקס אל "single mode" כדי לבצע את זה (הקלד "linux sungle" בזמן איתחול LILO או  
השתמש ב init 1 כ root כדי להיכנס אל מצב sungle user). אם נמצאו שגיאות במהלך  
בדיקת מערכת הקבצים, אני מקבל את ברירת המחדל לתיקון.



```
tune2fs -j /dev/hda2
```

(כ root, רק לקרנל שתומך בext3--RH7.2) כוון את פרמטר בר הכיוון של מערכת קבצים ext2. הדוגמא שלמעלה מראה כיצד להוסיף ג'ורנל למחיצת דיסק (hda2 בדוגמא זו), ובאופן אפקטיבי להמיר את מערכת הקבצים לext3 (journaling). כדי להשלים את המעבר עליך לערוך את הקובץ /etc/fstab ולשנות את מערכת הקבצים מext2 לext3, אחרת יכול מאוד להיות שתיתקל בבעיות--ext2 לא תמיר מערכת קבצים "מג'ורנלת" כבויה! כדי לבדוק מה הוא סוג מערכת הקבצים שלך השתמש בmount (ללא כל ארגומנטים) או בcat /etc/mntab. אם אתה זקוק למידע נוסף לגבי הקמת ext3 נסה ב:

<http://www.symonds.net/~rajesh/howto/ext3/ext3-5.html>

אפשרויות אחרות ל tune2fs מאפשרות לי להוסיף תג שם למחיצה, לכוון את מספר ההמרות שאחריהן מתבצעת בדיקה למערכת הקבצים (maximal mount count), או להפעיל בדיקות מערכת קבצים מבוססות זמן במקום (נפוץ פחות).

```
dd if=/dev/fd0H1440 of=floppy_image
dd if=floppy_image of=/dev/fd0H1440
```

(שתי פקודות, "dd=date duplicator" צור תעתיק של פלופי אל הקובץ שנקרא "floppy\_image" שבספריה הנוכחית. לאחר מכן העתק את floppy\_image (קובץ) אל פלופי אחר. עובד כמו "DICKCOPY" של דוס.

```
mkbootdisk --device /devfd0 2.4.2-3
```

צור פלופי חירום לאיתחול. בדרך כלל תישאל אם אתה מעוניין ליצור פלופי לאיתחול בזמן התקנת המערכת. הפקודה שלמעלה מראה כיצד לעשות זאת לאחר ההתקנה, על גבי כונן הפלופי הראשון (/dev/fd0). שם הקרנל שלך (יש בו צורך בפקודה, כאן 2.4.2-3) יכול להיקבע על ידי הרצת `uname -a` או `ls /lib/modules`.

## 5.14 ניהול חשבונות משתמשים והרשאות קבצים

```
useradd user_name
passwd user_name
```

(כ root) צור חשבון חדש. `useradd barabara`, אל תשכח לקבוע סיסמא למשתמש החדש כצעד הבא. ספריית הבית של המשתמש (שנוצר) היא `/home/user_name`. אתה יכול להשתמש גם בפקודה המקבילה `adduser user_name`.

```
ls -l /home/peter
useradd peter -u 503 -g 502
```

(כ root). צור חשבון שמתאים לספריה קיימת (אולי מהתקנה קודמת). אם מזהה המשתמש (UID) ומזהה הקבוצה (GID) היו שניהם 503, אני יוצר חשבון בעל שם, UID ו GID מתאימים. זה עוזר להימנע מהבלאגן שקשור בשינוי הבעלות על קבצי משתמש לאחר שידרוג מערכת.

```
userdel user_name
```

הסר חשבון (אתה חייב להיות root). ספריית הבית של המשתמש והדואר שעדיין לא הועבר אליו זקוקים לטיפול נפרד (באופן ידני מכיון שעליך להחליט מה לעשות עם הקבצים). ישנו גם `groupdel` כדי למחוק קבוצות.

`groupadd group_name`  
(`root`) צור קבוצה חדשה במערכת שלך. לא הכרחי במערכות ביתיות, אבל יכול להיות שימושי על גבי מכונה ביתית עם מספר משתמשים מועט.

לדוגמא, אני יכול ליצור את הקבוצה "friends", על ידי שימוש ב:  
`groupadd friends`  
ולאחר מכן לערוך את הקובץ `/etc/group`, ולהוסיף את שמי ואת השמות של חבריי אל השורה שמצגיה את הקבוצה, ככה שהשורה הסופית תיראה בערך כך:  
`friends:x:502:stan,pete, marie`  
לאחר מכן, אני יכול לשנות את ההרשאות לקובץ נבחר כך שהוא יהיה שייך לי ולקבוצה "friends".  
`chgrp friends my_file`  
כעת לחברי הקבוצה הרשומים יש גישה מיוחדת אל הקובץ, מה שאין לשאר העולם, כמו הרשאות קריאה וכתובה למשל:  
`chmod g=rw, o= my_file`  
האלטרנטיבה תהיה לתת הרשאת כתיבה לכולם, וזו בהחלט לא אלטרנטיבה מועילה, אפילו במחשב ביתי.

`groups`  
הצג את הקבוצות שאליהן המשתמש הנוכחי שייך. או שיכולתי להשתמש ב`groups john` על מנת לברר לאילו קבוצות המשתמש `john` שייך.

`usermod`  
`groupmod`  
(`root`) שני כלים של שורת הפקודה שמשמשים לשינוי חשבונות משתמשים וקבוצות ללא העריכה הידינית של הקבצים & `/etc/passwd /etc/shadow /etc/group` ורוב הזמן לא הכרחי.  
`/etc/gshadow`

`userconf`  
(`root`) כלי ניהול משתמשים מבוססי תפריט (מדיניות סיסמא, שינוי קבוצה, הוספת משתמשים וכו') חלק מהחבילה `linuxconf`, אבל ניתן להרצה בנפרד.

`passwd`  
שנה את הסיסמא של החשבון הנוכחי שלך. אם אתה `root`, אתה יכול לשנות סיסמא לכל משתמש על ידי:  
`passwd user_name`

`chfn`  
(`"change full name"`). שנה מידע לגבייך (שם מלא, מספר משרד, מספר טלפון, וכו'). מידע זה מוצג כשהפקודה `finger` מורצת אל מול שם הלוגאין שלך.

`chage -M 100 login_name`  
(`"change age"`). קבע את תאריך תפוגת הסיסמא ל-100 ימים למשתמש בשם `login_name`.

`quota username`  
`setquota username`  
`quotaon /dev/hda`  
`quotaoff /dev/hda`  
אוסף של פקודות המשמשות לניהול מכסות דיסק למשתמש. בדרך כלל לא נמצא בשימוש במחשב ביתי. "disk quota" מתפרש כהגבלות על שימוש בשטח דיסק פר משתמש. הפקודות מציגות (בהתאמה) את מכסת המשתמש, הגדרת המכסה למשתמש, הפעלה של מערכת המכסות למערכת קבצים נתונה (`/dev/hda` בדוגמא שלמעלה), כבה את מערכת המכסות. הפצות לינוקס "אופייניות" שראיתי מגדירות כברירת מחדל: ללא הגבלה לכל המשתמשים, ומערכת המכסות לא עובדת בכל מערכות הקבצים.

kuser  
(כ root, בטרמינל X) נהל משתמשים וקבוצות בעזרת ממשק גרפי. נחמד וכנראה מכסה את רוב מה שבאופן נורמלי תזדקק לו כדי לנהל חשבונות משתמשים.

chmod perm filename  
(change mode=) שנה את הרשאות הגישה לקובץ לקבצים שבבעלותך (אלא אם כן אתה root מה שמאפשר לך לשנות כל קובץ). אתה יכול לגרום לקובץ להיות נגיש בשלושה מצבים: קריאה (r), כתיבה (w), ביצוע (x) לשלוש רמות של משתמשים: בעלים (u), חברים בקבוצה שבבעלותה הקובץ (g), אחרים במערכת (o). בדוק את הרשאות הגישה הנוכחיות על ידי:

```
ls -l filename
```

במידה והקובץ נגיש לכל המשתמשים בכל הרמות הוא ייראה:

```
rw-rw-rw-
```

השלישייה הראשונה מראה את הרשאות הקובץ לבעלים של הקובץ, השניה לקבוצה שבבעלותה הקובץ, והשלישית לאחרים ("שאר העולם"). הרשאה שלילית מוצגת כ"-". כשמגדירים הרשאות, משתמשים בסימנים אלו: "u" (=משתמש או הבעלים של הקובץ), "g" (=קבוצה שבבעלותה הקובץ), "o" (=אחרים), "a" (=כולם, בעלים קבוצה ואחרים), "=" (=קבע את ההרשאות ל..), "+" (=הוסף את ההרשאה), "-" (=הסר את ההרשאה), "r" (=הרשאה לקריאת הקובץ), "w" (=הרשאת כתיבה, למעשה הרשאה לשנות את הקובץ), "x" (=הרשאה לבצע את הקובץ).

לדוגמא, פקודה זו תוסיף את ההרשאה לקרוא את הקובץ junk לכל (=משתמש+קבוצה+אחרים):

```
chmod a+r junk
```

פקודה זו תסיר את ההרשאה לבצע את הקובץ junk מאחרים:

```
chmod o+x junk
```

נסה כאן למידע נוסף.

אתה יכול להגדיר את הרשאות ברירת המחדל לקבצים חדשים שיצרת על ידי שימוש בפקודה umask (הסתכל בman umask).

```
chown new_ownership filename
```

```
chgrp new_groupname filename
```

שנה את הבעלים והקבוצה של הקובץ. עליך להשתמש בפקודות אלו לאחר שהעתקת קובץ לשימוש על ידי מישהו אחר. רק הבעלים של הקובץ יכול למחוק אותו.

```
lsattr files
```

הצג את האטריביוטים של קובץ (קבצים). לא נפוץ בשימוש מכיון שרוב האטריביוטים המעניינים עדין לא יושמו. ניתן לשנות את האטריביוטים על ידי הפקודה chattr. האטריביוטים הם: A (=אל תעדכן זמן כשהקובץ עובר שינוי), S (=עידכונים סינכרוניים), a (=הוסף רק אפשריים לקובץ זה), c (=קובץ דחוס ברמת הקרנל, לא מיושם עדין), i (=קובץ שאינו ניתן לשינוי), d (=no dump), s (=מחיקה מאובטחת), u (=אינו ניתן למחיקה, לא מיושם עדין). שימוש מענין יהיה יצירת קובץ שאינו ניתן למחיקה אפילו על ידי root (עד שהוא/היא מנקה את האטריביוט).

```
sudo /sbin/shutdown -h now
```

(כמשתמש רגיל, אני אתבקש להקליד את סיממת המשתמש שלי) מריץ את הפקודה "shutdown" (או פקודה אחרת שאליה ניתנה לך הרשאת הרצה על ידי מנהל המערכת שלך). בעזרת sudo, האדמיניסטרטור יכול לתת למשתמשים נבחרים את הזכויות להריץ פקודות נבחרות, ללא שייצטרך למסור את הסיסמא של root. הקובץ /etc/sudoers חייב להכיל משהו כמו:

```
my_login_name my_host_computer_name = /sbin/shutdown
```

```
pwck
```

```
grpck
```

(כ root, שתי פקודות). וודא את התקינות והשלמות של קבצי הסיסמא והקבוצה.

```
pwconv
```

```
grpconv
```

(כ root) קרוב לוודאי שלא תיזקק לפקודות אלו. הן ממירות סיסמאות מהסגנון הישן וקבצים של קבוצות אל קבצי "shadow" מאובטחים יותר.

## 5.15 התקנת תוכנה

```
rpm -ivh package_name-version.platform.rpm
```

(כ root) התקן את החבילה (האופציה "i", חייבת להיות האות הראשונה לאחר המקף), דבר אלי הרבה (האופציה "v" verbose) והדפס "hashes" כדי להראות את התקדמות ההתקנה (האופציה "h") rpm זה "Redhat Package Manager".

```
rpm -Uvh package_name-version.platform.rpm
```

(כ root) שדרג (האופציה "U", עליה להיות האות הראשונה לאחר המקף) חבילה, היה וורבלי (האופציה "v") והצג סולמיות ("h").

```
rpm -ivh --force --nodep package_name-version.platform.rpm
```

(כ root) התקן את החבילה והתעלם מכל קונפליקט אפשרי כמו גם מבעיות dependency.

```
rpm -e package_name
```

(כ root) הסר (האופציה "e" מחק) את החבילה package\_name. בבקשה שים לב להיעדרות של package\_name-version.platform.rpm מסוף שם החבילה (שם החבילה זהה לשם של קובץ rpm). שממנו הותקנה החבילה, רק ללא המקף, גירסה, פלטפורמה ו"rpm".

```
rpm -qpi package_name-version.platform.rpm
```

בצא שאילתה (האופציה "q", חייבת להיות האות הראשונה לאחר המקף) לגבי חבילה שעדין לא מותקנת (האופציה "p") באופן כזה שיוצג המידע (האופציה "i") אותו החבילה מכילה.

```
rpm -qpl package_name-version.platform.rpm
```

בצא שאילתה (האופציה "q", חייבת להיות האות הראשונה לאחר המקף) לגבי חבילה שעדין לא מותקנת (האופציה "p") באופן שבו תוצג רשימה (האופציה "l") של כל הקבצים שהחבילה מכילה.

```
rpm -qf a_file
```

מצא את השם של חבילה מותקנת אליה שייך או היה שייך בעבר הקובץ "a\_file". שימושי

למקרה שמחקתי בטעות קובץ וכעת עלי למצוא את החבילה המתאימה ולהתקין אותה מחדש.

```
rpm -qi package_name
```

בצע שאילתה לחבילה שכבר מותקנת כך שהיא תציג מידע לגבי עצמה. שים לב בבקשה ש `-version.platform.rpm` נעדר מהסוף של שם החבילה.

```
rpm -qai | more
```

בצע שאילתה לגבי כל החבילות שמותקנות אצלי במערכת ככה שהן יציגו את המידע שלהן. במערכת הפשוטה שלי, יש לי כ-600 חבילות מותקנות, לכן קרוב לוודאי שחייב להיות לי הרבה זמן פנוי כדי לקרוא את כל המידע שלהן. כדי לספור את החבילות שלך נסה:

```
rpm -qa | grep -c ' '
```

כדי למצוא חבילה מסויימת, נסה:

```
rpm -qa | grep -i the_string_to_find
```

(האופציה `-i` גורמת לgrep להתעלם ממצב התווים, לכן אותיות גדולות וקטנות יותאמו).

```
rpm -Va
```

וודא (האופציה "V") את כל החבילות שמותקנות במערכת שלי. זה מציג קבצים ששוננו מאז ההתקנה. לפניך המקרא לפלט:

.	הבדיקה הצליחה
c	קובץ זה הינו קובץ קונפיגורציה
5	בדיקת MD5 checksum נכשלה
S	גודל הקובץ הינו שונה
L	לינק סימבולי השתנה
T	זמן המודיפיקציה של הקובץ השתנה
D	קובץ התקן שונה
U	המשתמש שבבעלותו הקובץ, השתנה
G	קבוצה שבבעלותה הקובץ, השתנתה
M	מצב הקובץ (הרשאות ו/או סוג הקובץ) שונה

```
kpackage  
gnorpm  
glint
```

(בטרמינל X, כ root אם אתה רוצה להיות מסוגל להתקין חבילות, 3 פקודות) ממשקים גרפיים למנהל החבילות של REDHAT. "glint" מגיע עם RH5.2 ונראה מיושן כעת. gnorpm הוא הממשק הגרפי ה"רישמי" של REDHAT להתקנת חבילות, גירסאות ישנות יותר בדרך כלל איטיות ומבלבלות אבל הגירסא החדשה יותר (מגיעה עם RH7.0) משופרת הרבה יותר. kpackage היא התוכנית ה"רישמית של KDE והיתה די טובה לאורך כל הדרך. השתמש בכל אחת מהתוכניות כדי לראות אילו חבילות תוכנה מותקנות אצלך במערכת ואילו חבילות שעדיין לא הותקנו זמינות על גבי cdn של RH, כדי להציג מידע לגבי החבילות, וכדי להתקין אותן אם אתה רוצה בכך (ההתקנה חייבת להתבצע ל ידי root).

## 5.16 גישה לכוננים/מחיצות

mount

הסתכל כאן לפרטים על המרת כוננים. דוגמאות מוצגות בפקודות הבאות.

```
mount -t auto /dev/fd0 /mnt/floppy
```

(כ root) המר את הפלופי. הספרייה /mnt/floppy חייבת להיות קיימת, להיות ריקה ושלא תהיה הספרייה הנוכחית שלך.

```
mount -t auto /dev/cdrom /mnt/cdrom
```

(כ root) המר את ה-cd. ייתכן ויהיה עליך ליצור/לשנות את הקובץ /dev/cdrom תלוי היכן ה-CDROM שלך. הספרייה /mnt/cdrom חייבת להיות קיימת, להיות ריקה ולא להיות הספרייה הנוכחית שלך.

```
mount /mnt/floppy
```

(כמשתמש או כ root) המר פלופי כמשתמש. הקובץ /etc/fstab חייב להיות מוגדר לעשות את זה. הספרייה /mnt/floppy חייבת לא להיות הספרייה הנוכחית שלך.

```
mount .mnt/cdrom
```

(כמשתמש או כ root) המר cd כמשתמש. הקובץ /etc/fstab חייב להיות מוגדר לעשות את זה. הספרייה /mnt/cdrom לא יכולה להיות ספריית העבודה הנוכחית שלך.

```
umount /mnt/floppy
```

בטל את ההמרה של הפלופי. הספרייה /mnt/floppy חייבת לא להיות ספריית העבודה הנוכחית שלך (או של מישהו אחר). תלוי בתצורה שלך, ייתכן ולא תוכל להסיר/לבטל המרה מכונן שהומר על ידי מישהו אחר.

## 5.17 כלי ניהול רשת

netconf

(כ root) כלי מבוסס תפריט להגדרת הרשת שלך.

```
ping machine_name
```

בדוק אם אתה יכול ליצור קשר עם מכונה אחרת (תן את שם המכונה או את כתובת הקו שלה). לחץ על <Ctrl>C (ללא <Ctrl>c, הפקודה תמשיך לרוץ). כמו כל הפקודות של לינוקס, ל ping יש אופציות, כולל ה "ping of death" attack, כנראה שאתה יכול לשלוח ping לכמה שרתים ככה שהם מתים--נסה את האופציות -s | -f.

```
route -n
```

הראה את טבלת הניתוב של הקרנל.

```
host host_to_find
```

```
nslookup host_to_find
```

```
dig ip_to_find
```

(3 פקודות, השתמש בכל אחת מהן) שלח שאילתה לשרת dns שלך לגבי שם האינטרנט (או כתובת ip) host\_to\_find. בדרך זו אתה יכול לבדוק אם שרת dns שלך עובד. אתה יכול גם למצוא את שם מארח כשאתה יודע רק את כתובת הקו שלו.

traceroute host\_to\_find  
בדוק איך ההודעה שלך מנותבת אל host\_to\_trace (שיכול להיות גם שם מארח וגם כתובת IP).

mtr host\_to\_trace  
(כ root) כלי עוצמתי ונחמד שמשלב את הפונקציונליות של הפינג הישן ושל traceroute.  
(RH7.0)

nmblookup -A ip\_address  
המצב של מכונת MS windows שמחוברת לרשת (עם שם netbios). פקודה זו היא המקבילה של הפקודה nbtstat מווינדוס.

ipchains -p forward DENY  
ipchains -A forward -s xxx.xxx.xxx.0/24 -d 0.0.0.0/0 -j MASK  
(שתי פקודות).  
הגדר מדיניות ip forwarding על מנת ליישם masquerading בפיירוול. (לא מאובטח מדי אבל פשוט). המטרה: כל המחשבים מהרשת הביתית שלך ייראו לעולם החיצון כמחשב אחד עסוק מאוד, תוכל למשל, לצפות באינטרנט מכל המחשבים בת אחת.  
החלף את הא בספרות של כתובת ה IP (class C) שהענקת לרשת הביתית שלך.

ipchains -L  
הצג את כל חוקי הפירוול. השתמש כדי לבדוק אם הפירוול שלך עובד.

iptables -L  
קרגל 2.4.x משתמש ב"iptables". הפקודה למעלה מציגה את חוקי הפירוול.

firewall -config  
(כ root, בטרמינל X). ממשק גרפי לבניית פיירוול מותאם אישית.

ifconfig  
(כ root) הצג מידע לגבי ממשקי הרשת שפעילים כרגע. (אטרנט, kppp, וכו'). מתאם הרשת הראשון שלך אמור להופיע כeth0, השני כeth1 וכו'. הקקק הראשון על גבי מודם יופיע כkppp0 השני כkppp1 וכו'. "lo" הוא ממשק ה"loopback only" שצריך להיות פעיל תמיד. השתמש באופציות (הסתכל ב --help ifconfig) כדי להגדיר את הממשקים שלך.

ifup interface\_name  
/sbin/ifup על מנת להרית כמשתמש) מפעיל ממשק רשת:

```
ifup eth0
ifup kppp0
ifup kppp1
```

משתמשים יכולים לכבות או להפעיל ממשק kppp רק כשניתנת להם הרשות בהגדרת הקקק (על ידי netconf). כדי להפעיל ממשק kppp (חיבור בעזרת חיוג), אני בדרך כלל משתמש בקקק שזמין תחת תפריט ה"K" ב KDE (או על ידי הקלדת kppp בטרמינל X).

/etc/rc.d/init.d/network restart  
הפעל מחדש את הרשת על ידי שימוש בסקריפט האיתחול הרגיל (זה שנמצא בשימוש בעת

איתחול המערכת). שימושי אם בדיוק ערכת שינויים לקונפיגורציית הרשת שלך. כל שירות אחר שרשום ב init.d ניתן להפעלה, כיבוי, או הפעלה מחדש בדרך די דומה (קרא לסקריפט עם האופציות (stop, start, restart).

```
ifdown interface_name
```

כדי להריץ כמשתמש). כבה ממשק רשת:

```
ifdown ppp0
```

```
netstat | more
```

הצג המון מידע (יותר מדי?) לגבי מצב הרשת שלך.

```
/usr/sbin/mtr --gtk
```

(כ root, X windows, אם אתה מעונין בממשק מבוסס gtk). כלי לדיאגנוסטיקת רשת שמשלב את היכולות של ping & traceroute.

```
nmap ip_number
```

ממפה את הפורטים על המכונה עם ה ip\_number. זה באמת שימושי כדי להשיג אבטחה בקונפיגורציית הרשת שלך כשאתה יכול להציג פורטים פתוחים. nmap יכול גם לבצע "fingerprinting" של מערכת ההפעלה. באופן נורמלי אנשים (וספקי האינטרנט שלהם) לא אוהבים שהפורטים של המחשבים שלהם נסרקים (הם רואים את זה כגישוש לפני התקפה) לכן הם עלולים להתלונן אם הם יגלו --למד כיצד להשתמש ב nmap על המחשבים ברשת הביתית שלך, אחרת בקרוב מאוד תשמע מספקי האינטרנט שלך (התלונות יופנו אליהם). איך אני יודע את זה בכלל?

```
ethereal
```

(כ root, בטרמינל X) מנתח רשת--הסתכל בתנועת הרשת שעוברת דרך המחשב שלך. שימוש ב ethereal יכול להיות לא אתי במצבים מסוימים, שימוש לא מורשה במקומם העבודה יכול להוות סיבה הוגנת לפיטורים.

```
tcpdump -i ppp0 -a -x
```

(כ root) הצג את כל תנועת הרשת שעוברת דרך הממשק הטלפוני הראשון (ppp0) בascii וכ hexadecimals. קרוב לודאי פלט רב מדי. tcpdump הוא יותר כלי גס ויכול להועיל לבנית כלים יותר מותאמים אישית להאזנה/רישום למה שאתה צריך.

## 5.18 פקודות שקשורות למוזיקה

```
cdplay play 1
```

נגן את הרצועה הראשונה מתקליטור מוזיקה (audio). השתמש ב cdplay כדי לנגן את כל ה cd. השתמש ב cdplay stop כשמספיק לך.

```
eject
```

ארגן לעצמך מיתקן להחזקת כוסות קפה (:). (שלוף את מגש כונן התקליטורים). ברירת המחדל של פקודה זו מתייחסת אל כונן התקליטורים אבל אתה יכול להשתמש בה לשליפה של מדיה ניידת אחרת על ידי ציון נקות ההמרה או ההתקן, אני יכול לשלוף החוצה את דיסקט zip מכונן zip פרללי (חיצוני) על ידי: eject /dev/sda4.



```
play my_file.wav
נגן קובץ wav
```

```
rec my_file.wav
הקלט קובץ wav ממיקרופון.
```

```
mpg123 my_file.mp3
נגן קובץ mp3.
```

```
mpg123 -w my_file.wav my_file.mp3
צור קובץ wave מקובץ mp3. שימושי אם אתה מעונין לכתוב תקליטור אודיו רגיל מmp3--
עליך להמיר תחילה את קבצי הmp3 אל הפורמט wav. *. . אל תתפלא אם ההמרה איטית--
פריסה (decompressing) של קבצי mp3 הינה פעולת עיבוד אינטנסיבית מאוד.
```

```
xmms
(בטרמינל X) נגן mp3 בעל משק גרפי נחמד.
```

```
freeamp
(בטרמינל X) עוד נגן mp3 בעל ממשק גרפי.
```

```
lame input_file output_file
מקודד mp3.
```

```
cdparanoia -B "1-"
(צורב תקליטורים) קרא את התוכן של תקליטור אודיו ושמור אותו אל תוך קבצי wav
בספריות הנוכחיות, רצועה אחת לכל קובץ wav. הפירוש של "1-" הוא "מרצועה 1 עד
לאחרונה". -B כופה עליו לשים כל רצועה בקובץ נפרד.
```

```
playmidi my_file.mid
נגן קובץ midi .midi -r my_file.mid יציג אפקטים במצב טקסט על גבי המסך
במצב טקסט.
```

```
sox sudio_file another_format_audio_file
("SOund eXchacnge") המר מכמעט כל פורמט של קובץ אודיו אל אחר (אבל לא mp3).
הסתכל בman sox לרשימה של פורמטי קבצים נתמכים (הרבה). sox מאפשר לך גם
להוסיף אפקטים מיוחדים לקבצי האודיו שלך.
```

```
kscd
(בטרמינל X) נגן תקליטורים.
```

```
kmidi
(בטרמינל X) נגן midi.
```

```
kmid
(בטרמינל X) נגן midi/קריוקי.
```

```
kmix
(בטרמינל X) מיקסר צלילים.
```

studio  
(בטרמינל X) סטודיו לסאונד--ערוך קבצי קול, הוסף אפקטים, וכו'.

extace  
(בטרמינל X) הצגה ויזואלית של צלילים.

festival --tts my\_file.txt  
אמור את התוכן של הקובץ my\_file.txt (ascii text). "festival" הוא סינטיסייזר לדיבור. כדי לומר משהו משורת הפקודה, עליך להפעיל את "festival" ולאחר מכן, בפרומפט "festival">, להקליד את הפקודה המתאימה, כמו בדוגמא הבאה (**מודגש** מייצג את הפרומפט):

```
festival  
festival>(SayText "good dog, really good dog")  
festival>(quit)
```

### 5.19 פקודות שקשורות לגרפיקה

kghostview my\_file.ps  
(בטרמינל X) הצג קובץ postscript על גבי המסך. אני יכול להשתמש ב ghostview בעל המראה המיושן יותר או gv כדי להשיג את אותו האפקט. אני גם יכול להדפיס את הקובץ מתוך התוכנה.

xpdf my\_file.txt -U 2  
(בטרמינל X) צפה בקובץ pdf.

enscript my\_file.txt -U 2  
המר קובץ טקסט לpostscript והדפס אותו במדפסת ברירת המחדל. יכולתי באותה מידה לשלוח את הפלט אל קובץ postscript:  
enscript my\_file.txt -U 2 -O my\_file.ps

האופציה -U 2 גורמת לenscript להדפיס שני עמודים לוגיים על גבי עמוד לוגי אחד מה שחוסך לי נייר, ויוצר תצורות הדפסה קומפקטיות ונוחות יותר. אתה יכול גם לבחור 4 עמודים לכל עמוד, מה שגורם לפלט להיות קריא פחות. enscript הוא כלי גמיש באמת, הסתכל ב man enscript כדי לבחור בין אופציות הפורמט הרבות.

ps2pdf my\_file.ps my\_file.pdf  
זור קובץ pdf (portable document format של adobe) מקובץ postscript.

mpage -2 my\_file.ps > new\_file.ps  
הדפס את קובץ postscript my\_file.ps. שלח את הפלט אל שני עמודים לוגיים על גבי עמוד לוגי אחד. שמור את הפלט אל הקובץ new\_file.ps.

ps2ps file.ps new\_file.ps  
psnup -nup 2 -pletter new\_file.ps new\_file2.ps  
דרך נוספת לגרום לקובץ postscript להכיל שני עמודים לוגיים על גבי עמוד לוגי אחד. ראשית, אני משתמש ב"postscript distiller" שנקרא ps2ps כדי להפוך את קובץ postscript לפשוט יותר ( כשהמחיר הוא הגדלת הגודל שלו). לאחר מכן אני משתמש

בכלי psnup כדי ליצור את new\_file2.ps אשר מכיל שני עמודים לוגיים בעמוד פיזי אחד. יכולתי גם להניח 4 או 8 דפים לוגיים על גבי עמוד פיזי אחד.

gimp

(בטרמינל X) נראה צנוע אבל למעשה זהו מעבד תמונה חזק מאוד. דורש קצת לימוד כדי להשתמש בו, אבל לחלוטין מצויין לאומנים, כמעט ואין משהו שאינך יכול לעשות בעזרת gimp. השתמש בכפתור הימני של העכבר שלך כדי לקבל תפריטים מקומיים, ולמד כיצד להשתמש בשכבות. שמור את עבודתך בפורמט הקבצים של gimp, \*.xcf (כדי לשמור על שכבות לעריכה עתידית) ורק לאחר מכן שטח אותו ושמור אותו כpng (או כל מה שלא יהיה) כדי להשתמש בו.  
"המפתח הוא ללמוד כיצד לבחור את הבחירה הנכונה."

gphoto

(בטרמינל X) עורך תמונות.

kpaint

(בטרמינל X) תוכנית קטנה לצביעת bitmap. (בסגנון של "paintbrush").

xfig

(בטרמינל X) תוכנית רישום קטנה. שימושי ליצירת שרטוטים פשוטים ודיאגרמות.

dia

(בטרמינל X) כלי לשרטוט דיאגרמות מרכיבים בנויים.

display my\_picture

(בטרמינל X) הצג תמונה לצפייה בלבד. בנוסף, אתה יכול גם להקליד & display ולבחור קובץ מהתפריט כדי לצפות בתמונה, לסובב אותה, לשנות את הצבעים שלה, להחיל אפקטים מסויימים וכו'. display היא חלק מהחבילה ImageMagick, ביחד עם כמה כלים אחרים שמתוארים למטה.

identify -verbose my\_picture

תן לי תאור של קובץ תמונה my\_file מבחינת: פורמט, סוג, רמה, גודל בפיקסלים, מספר הצבעים, גודל בבייטים, וכו'.

convert -geometry 60x80 my\_picture.gif out\_small\_picture.gif

המר את גודל התמונה ל60x80 פיקסלים. הסתכל כמה שורות למטה לקבלת דוגמא לגבי כיצד להשתמש בconvert כדי להמיר בין שני פורמטי קבצים גרפיים שונים.

animate -delay 6x5 pic1 pic2 pic3

הצג שתי תמונות או יותר ברצף. בדוגמא שלמעלה, הקבצים נקראים pic1, pic2, pic3, ההשהייה בין התמונות היא 0.06 שניות, וכל רצף התצוגה חוזר על עצמו ב5 שניות.

combine pic1 pic2 pic3 combined\_pic.miff

שלב שתי תמונות או יותר אל תמונה אחרת. אתה יכול למשל להניח לוגו על גבי כל תמונה.

montage -geometry 800x600+0+0 my\_picture montage.miff

צור תמונה אחת פרוסה מmy\_picture כך שסך הגודל שלה יהיה 800x600 פיקסלים, עם גבול של 0x0. הפלט הולך אל הקובץ montage.miff.

zgv my\_picture  
הצג תמונה לצפייה על גבי מסך vga (X לא הכרחי).

```
giftpnm my_file.gif > my_file.pnm  
pnmtopng my_file.pnm > my_file.png
```

המר תמונות gif אל קובץ מחק נייד, גס. ולאחר מכן המר את המחק אל קובץ png, שהוא בעצם סטנדרט חדש וטוב יותר לתמונות אינטרנט (טוב יותר מבחינה טכנית ובנוסף אין סכנה שתתבע על ידי הבעלים של הפטנטים של gif).

```
xwd -out my_capture_screen_file.xwd
```

(בטרמינל X) לכוד את התוכן של מסך X-windows אל תוך קובץ dump גרפי של "X windows" (\*.xwd). אתה יכול להמיר בשלב מאוחר יותר את הxwd אל הפורמט האהוב עליך על ידי הכלי convert. תחת KDE, אתה יכול להשתמש גם בקיצורי הדרך של המקלדת <Alt><PrintScreen> או <Ctrl><Alt><PrintScreen> כדי להעתיק את החלון הנוכחי או את כל שולחן העבודה אל תוך לוח העריכה (clipboard).

```
convert my_capture_screen_file.xwd my_capture_screen.jpg
```

המר את קובץ הxwd של X windows אל הפורמט \*.jpg. הכלי convert יכול להמיר גרפיקה מ/אל פורמטי קבצים רבים.

```
import -display 192.5.100.10:0 -windows root my_file.jpeg
```

לכוד את התוכן של המסך של root מתוך X-windows שרץ על השרת 192.5.100.10 תצוגה 0. קובץ הפלט הוא my\_file.jpeg (שנה את פורמט הקובץ על ידי כך שתיתן לו את הסיומת המתאימה). עליך להיות בעל ההרשאות המתאימות לכתיבה אל המסך כדי להיות מסוגל ללכוד את התוכן שלו (ההרשאה לכל יכולה להינתן על ידי הרצת + xhost בטרמינל X). הסתכל ב man import לאופציות.

```
ksnapshot
```

(בטרמינל X) כלי מבוסס ממשק גרפי ללכידת תוכן המסך.

```
xine frankenstein.avi
```

(בטרמינל X) צפה בסרט מהקובץ "frankenstein.avi". נראה טוב יותר מאשר בטלוויזיה.

## 5.20 משחקים קטנים

סביר מאוד להניח שרבה משחקים קטנים מותקנים במערכת שלך. לפניך רק דוגמא למה שמתקן על גבי תקליטור ההפצה הסטנדרטי שלי.

```
kpat
```

(בטרמינל X) משחק סבלנות בקלפים. sol (מהר) וגם pysol (איטי אבל עשיר) הן שתי אפשרויות. החביב עליי הוא: sol --variation=freecell&

```
xboing
```

(בטרמינל X). משחק pin-ball נחמד מאוד.

```
xboard
```

(בטרמינל X) שחמט. המחשב משחק יותר מדי טוב בשבילי:)

konquest &  
(בטרמינל X) התחרה בבנוך במסע לכיבוש הגלקסיה. משחק לוח מענין.

kmines  
(בטרמינל X) שולה מוקשים.

civserver  
civclient  
(בטרמינל X) הפעל שרת למשחק FreeCivilization (פקודה ראשונה). לאחר מכן,  
כשהשרת כבר רץ, הפעל את הקליינט (פקודה שניה). מישהו אחר הפעיל קליינט נוסף--  
ואתם משחקים. FreeCv הגיע עם התקליטורים של RH7.0 שיש לי.

fgfs  
"Flight Gear" סימולטור טיסה.