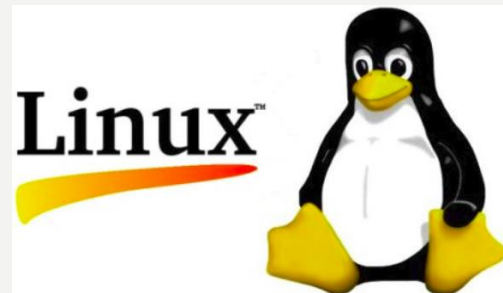


LINUX



HODAYA - AMAR (c)

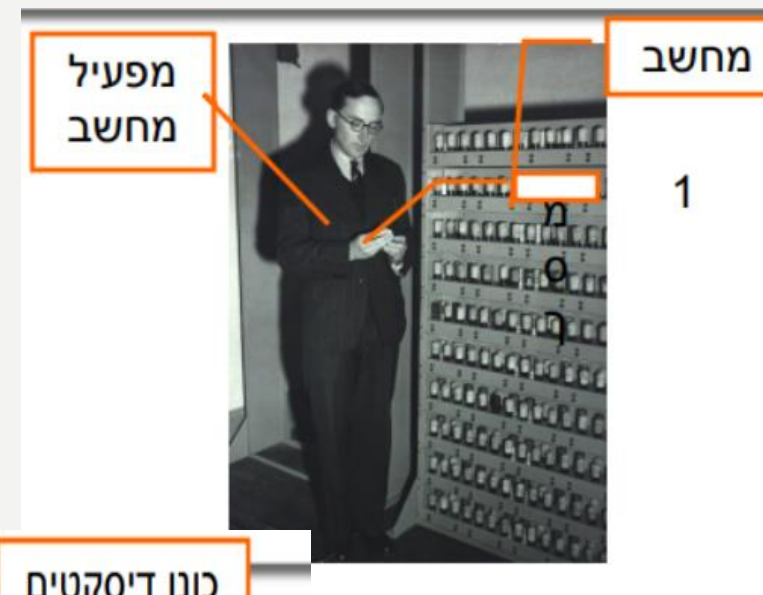
אין להשתמש \ להעתיק \ לצלם
ללא רשות מפורשת בכתב

0527605557

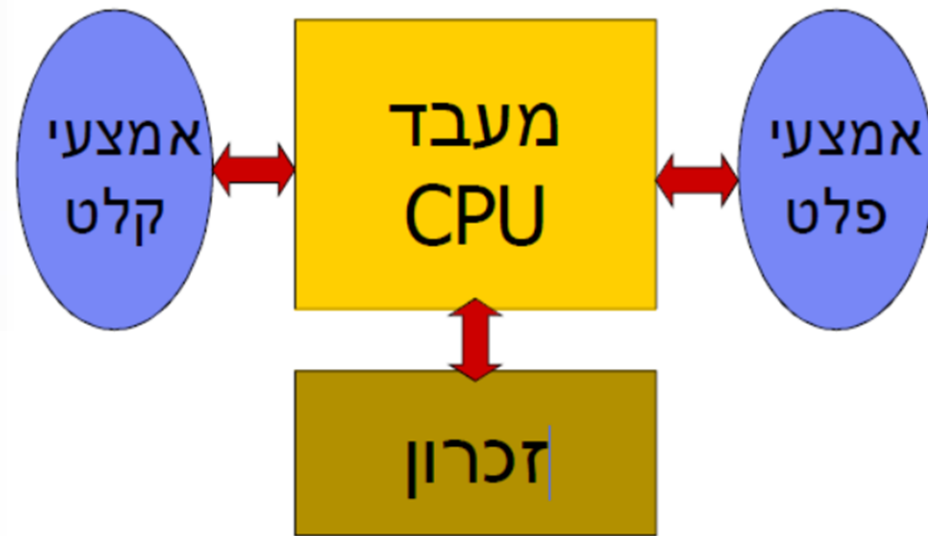
LINUX

- מבנה סכמטי של מחשב
 - חומרה
 - תוכנה
- מהו לינוקס Linux, unix, ios, android
 - Kernel space
 - User space
- יתרונות לינוקס
 - קוד פתוח
- מערכות הפעלה נפוצות של לינוקס, מערכות מסחריות, מערכות מיוחדות, ואנדרואיד

מבנה סכמטי של מחשב



מבנה סכמטי של מחשב



חומרה - מבנה המחשב - קלט פלט

אמצעי קלט



אמצעי פלט



חומרה - מבנה המחשב - רכיבים פנימיים

זיכרון ראשי



לוח אם



מעבד



דיסק קשיח



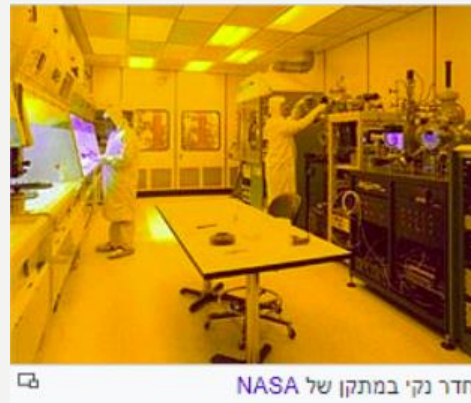
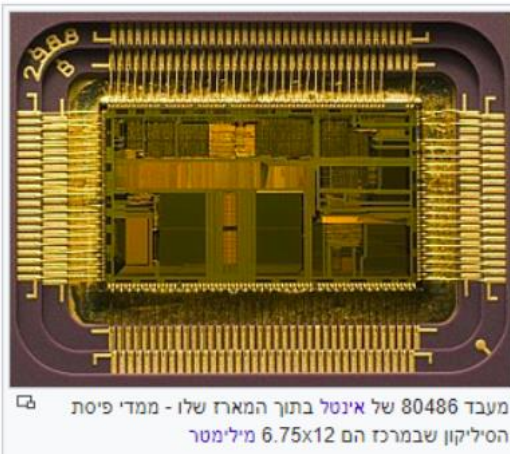
כרטיס מסך



מעבד

• מעבד הוא יחידת עיבוד מרכזית CPU - Central Processing Unit

- בהתאם לארכיטקטורת פון נוימן הפקודות אותן מקבל המעבד מאפשרות לו קריאת מידע מהזיכרון או מהתקנים שונים, ביצוע פעולות חשבוניות ולוגיות על מידע זה וכתובת תוצאות החישוב בחזרה לזיכרון או לחלופין שליחתו להתקנים חיצוניים. הפעולות הלוגיות מאפשרות בקרת זרימה וחזרה על פקודות ככל שנדרש.
- הפקודות הן בסיסיות ביותר ובנויות, כל אחת, מרצף קצר של ביטים. רצף זה קרוי שפת מכונה.
- מבחינה מספרית, הרוב המכריע של המעבדים המיוצרים כיום, משמשים כיחידות בקרה של מוצרי צריכה. חלק ממוצרים אלו הם ציוד הקפי של מחשבים אישיים (כגון: צגים, מדפסות, כרטיסי רשת) ואחרים אינם (מכוניות, מכונות כביסה, טלפונים סלולריים).



פעולות ומבנה המעבד

- לצורך ביצוע פקודה בודדת המעבדים משתמשים באחד מהסטים של הפעולות הבאות (תלוי במעבד):
 - חישוב הכתובת בה נמצאת הפקודה הבאה
 - קריאת הפקודה הבאה לביצוע מהזיכרון
 - פיענוח הפקודה: בשלב זה המעבד מפענח מהי הפקודה שצריכה להתבצע, איזו יחידת ביצוע להפעיל ואילו משתנים מעורבים.
 - חישוב: בשלב זה קוראים ערכים מהזיכרון אם צריך, ובהתאם לסוג הפעולה מופעלת יחידת החישוב המתאימה של המעבד לביצוע הפעולות האריתמטיות והלוגיות הרלוונטיות.
 - כתיבה חזרה לזיכרון: הערך המחושב נכתב למקום אחסון התוצאה.
- יחידות הקיימות בכל מעבד
 - אוגרים (Registers או "רגיסטרים"): כל אוגר הוא יחידה אחת של זיכרון פנימי מהיר ביותר הנמצא לרוב בתוך יחידת העיבוד המרכזית של מחשב אשר מאפשר אחסון ערכים, בדרך כלל זמנית, עבור פעולות בסיסיות שונות מסט הפקודות של המעבד (חיבור, חיסור, והשוואה). יש מעבדים בהם האוגרים הם ייעודיים, כלומר פעולות מסוימות מוגבלות לאוגר או אוגרים מסוימים, ולעומתם מעבדים אחרים.
 - יחידת בקרה: יחידה האחראית על שלבי הקריאה והפיענוח של פקודות.
 - יחידה אריתמטית-לוגית (ALU) חידה זו מבצעת פעולות במספרים שלמים (חיבור, חיסור, השוואה) ופעולות לוגיות שונות (כדוגמת שער לוגי, NAND, NOR, NOT, כדומה). היחידה נחשבת ללב המעבד, ויעילותה היא החשובה ביותר בביצועי המעבד.

ריבוי משימות

- ריבוי משימות (Multitasking) הוא היכולת של מחשב לבצע מספר משימות במקביל.
- במקרה של מחשב עם מעבד יחיד יכולה לרוץ רק משימה אחת בכל פרק זמן נתון, אולם אפשר לקבל הדמיה של הרצת מספר משימות במקביל על ידי מעבר מהיר בין משימות שונות
- המעבר בין הרצת משימות שונות נקרא החלפת הקשר, מכיוון שבו עוברים מהסביבה (ההקשר) של משימה אחת לזו של משימה אחרת. למשימות קוראים (במערכות שונות) תהליכים, תהליכונים, או משימות

תזמון משימות

- במדעי המחשב, אלגוריתם תזמון Scheduling הוא אלגוריתם המממש מדיניות תזמון מסוימת עבור סדרן התוכניות. מדיניות זו קובעת את דרך חלוקת המשאבים, למשל המעבד, בין תהליכים ותהליכונים. הצורך באלגוריתם תזמון נובע מהדרישה לריבוי משימות שקיימת ברוב המערכות המודרניות.
- מטרתו העיקרית של אלגוריתם תזמון הן ניצול מקסימלי של המעבדים ושמירה על זמן תגובה נמוך של המערכת לאירועים שונים. דרישות נוספות מהאלגוריתם יכולות להתבסס על אופי המערכת בה הוא מתקיים. לדוגמה:
 - במערכות זמן אמת, הדרישה לזמן תגובה נמוך הוא הכרחי ביותר עבור חלק מהתהליכים. כך למשל, במערכת הטייס האוטומטי, במידה והמערכת מגלה ירידה במהירות הטיסה, נרצה שהמערכת תגביר מהירות באופן מיידי.
 - במערכות שנמצאות במחשב רשת מרכזי, בו התהליכים יכולים להיות משתמשי קצה שונים, ההוגנות (חלוקת זמן שווה וזמן המתנה שווה בין התהליכים) תהיה דרישה משמעותית.
- כמו כן, ניתן לבצע אופטימיזציות ספציפיות לחומרה ובכך לשפר את ניצול המעבדים. (ע"י אלגוריתמי תזמון שמבצעים את האופטימיזציות)

זיכרון

- זיכרון גישה אקראית (Random Access Memory - RAM) הוא שם כללי למספר רב של סוגי זיכרון מחשב, המתאפיינים כולם ביכולת המעבד לגשת ישירות לכל תא בזיכרון לפי כתובתו, לכתוב בו ולקרוא ממנו. ההתייחסות הנפוצה לזיכרון מחשב היא למעשה התייחסות לזיכרון הגישה האקראית הראשי שלו.
- סוג הזיכרון הנמצא בשימוש נפוץ ביותר בימינו הוא DRAM. זיכרון מסוג זה הוא זיכרון נדיף, כלומר מאבד את תוכנו עם ניתוק הזיכרון ממקור האנרגיה שלו. בנוסף, DRAM דורש רענון מספר פעמים בשנייה כדי לשמור את תוכנו.

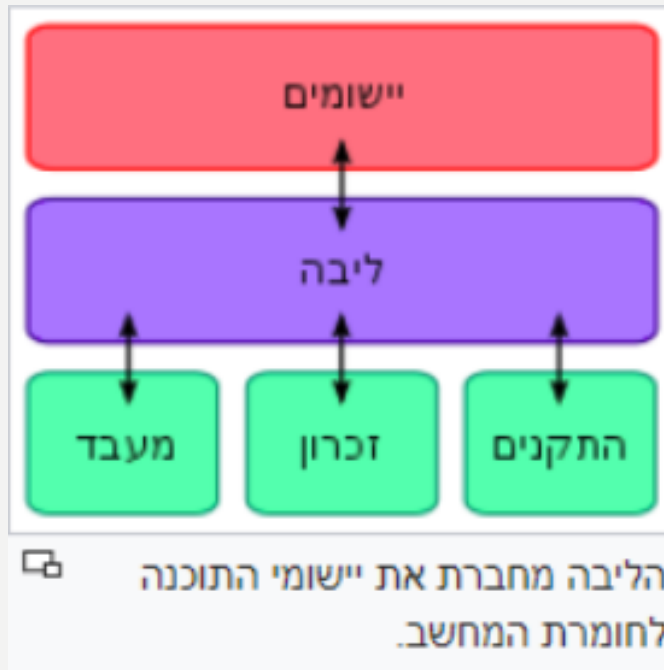


כרטיסי DDR בנפח 512 מגהבייט ובתדר 400 מגהרץ

OVERVIEW

– Kernel space (גרעין, ליבה - the core of the operating system)
• מתאם בין החומרה (מעבד, זיכרון, וכו') לבין התוכנה (מערכת הפעלה, התוכנות של המחשב)

– User Space (מרחב המשתמש)
• יישומים, התכונות המותקנות על המחשב, אפליקציות



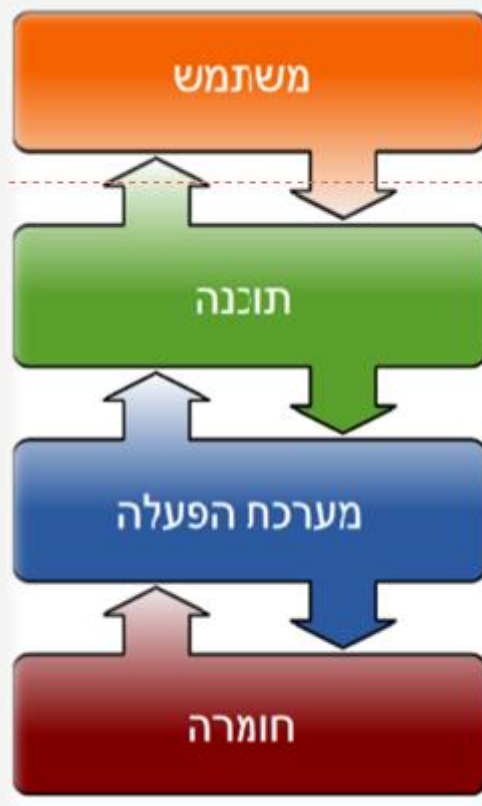
- היא הרכיב המרכזי של מרבית מערכות ההפעלה
- זהו הגשר שבין תוכניות המחשב לבין עיבוד הנתונים עצמו שמבוצע ברמת החומרה. אחד התפקידים העיקריים של הליבה הוא ניהול משאבי המערכת (התקשורת שבין רכיבי החומרה והתוכנה). בדרך כלל, בתור הרכיב הבסיסי של מערכת הפעלה, הליבה יכולה לספק את שכבת האבסטרקציה ברמה הנמוכה ביותר עבור המשאבים (בייחוד עבור מעבדים והתקני קלט-פלט) שהתוכניות צריכות לשלוט עליהם על מנת לבצע את תפקידן.
- המשימות של מערכות הפעלה מבוצעות באופן שונה על ידי ליבות שונות, בהתאם לעיצוב ולמימוש
- ליבת מערכת ההפעלה היא התוכנית היחידה אשר מוכנה לריצה בכל זמן שהוא. בנוסף, הליבה היא התוכנה היחידה שיכולה לבצע אוסף פקודות בצורה אטומית (ללא שום הפרעה או הפסקה מגורם כלשהו). כל מערכת הפעלה חייבת ליבה כדי לפעול, אך הליבה אינה בהכרח ייחודית למערכת ההפעלה - מערכות הפעלה שונות יכולות להשתמש בליבה זעה. לדוגמה, על ליבת לינוקס ניתן להריץ הפצת לינוקס שולחנית אך גם אפשר להריץ את מערכת ההפעלה אנדרואיד למכשירים ניידים.

תפקידיה הבסיסיים של ליבה

תפקידה הראשי של הליבה הוא לנהל את משאבי המחשב ולאפשר לתוכניות אחרות לרוץ ולהשתמש במשאבים אלה.

בין משאבי המחשב נכללים:

- יחידת העיבוד המרכזית (המעבד): זהו החלק המרכזי ביותר במערכת מחשב, האחראי לריצה או ביצוע של תוכניות על גביו. בכל רגע נתון, באחריות הליבה לקבוע עבור איזו מהתוכניות הרבות שרצות יש להקצות את המעבד או המעבדים (שכל אחד מהם בדרך כלל יכול להריץ רק תוכנית אחת בו זמנית).
- זיכרון המחשב: הזיכרון משמש הן לאחסון של הוראות של תוכניות והן לאחסון של נתונים. בדרך כלל גם על ההוראות וגם על הנתונים להיות מאוחסנים בזיכרון לצורך ביצועה של התוכנית. בדרך כלל תוכניות רבות יבקשו גישה לזיכרון, ולעיתים קרובות ידרשו יותר זיכרון מהזיכרון הזמין שיש במחשב. באחריות הליבה לקבוע באיזה חלק מהזיכרון כל תהליך רשאי להשתמש, ולקבוע מה יש לעשות כאשר אין מספיק זיכרון זמין.
- התקני קלט/פלט (I/O) שמחוברים אל המחשב: כגון מקלדת, עכבר, כונני דיסקים, מדפסות, צגים וכו'. הליבה מעבירה בקשות לביצוע קלט/פלט מהתוכניות אל ההתקן המתאים (או לחלקים מההתקן, במקרה של קבצים על דיסק או חלונות על גבי צג), ומספקת דרכים נוחות לשימוש בהתקן



- זהו התכונה הראשונה שעולה עם הדלקת המחשב והיא זו המאפשרת לו לפעול
- מערכת ההפעלה מגשרת בין המשתמש, החומרה ויישומי התוכנה
- כל מחשב חייב מערכת הפעלה, ממחשבי על ועד לטלפונים סלולריים



- מערכת הפעלה מספקת שירותים שונים למשתמש במחשב וגם למתכנת מי שכותב את קוד התוכנה לדוגמא:

– מבחינת המשתמש במחשב:

- עבודה נוחה עם קבצים, התקני קלט\פלט
- הרצת תוכנית במקביל
- תוכניות עזר קטנות: מחשבון, דפדפן, פנקס רשימות וכו'

– מבחינת המתכנת:

- ביצוע פלט\קלט
- עבודה נוחה עם קבצים
- הרצת תוכניות במקביל
- חלוקת משאבים, למשל זיכרון

LINUX

קצת היסטוריה

- בכדי להבין את לינוקס כראוי, אנחנו צריכים לחזור אחורה בזמן בערך 30 שנה...
- דמיינו מחשבים כבתים גדולים, אפילו איצטדיון. כאשר גודל המחשבים היוו בעיה ממשית, היה דבר אחר שהפך את זה ליותר גרוע: לכל מחשב היתה מערכת הפעלה שונה, תוכנות היו תמיד מתאימות לשרת מטרה מסוימת, ותוכנה שנועדה למערכת הפעלה אחת לא עובדת על מערכת הפעלה אחרת. אפשרות לעבוד עם מערכת אחת לא אומר אוטומטי שאתה יכול לעבוד עם אחרת. זה היה קשה גם למשתמשים וגם למנהלי מערכות (system administrators)
- מחשבים היו מאוד יקרים והיו צריכים להוציא עוד כסף לאחר הקניה המקורית רק בכדי שהמשתמשים יבינו כיצד הם עובדים. המחיר הכולל היה עצום.
- טכנולוגית העולם לא היה כל כך מתקדמת, לכן הם היו צריכים לחיות עם הגודל של המחשבים עוד עשור. ב-1969, קבוצה של מתכנתים במעבדות בל התחילו לעבוד על פתרון לבעיית התוכנה, בכדי לטפל בבעיית התאימות של התוכנה. הם פיתחו מערכת הפעלה חדשה אשר היתה
- 1. פשוטה ואלגנטית.
- 2. כתובה בשפת סי C במקום באסמבלי.
- 3. יכולה למחזר קוד
- המפתחים של מעבדות בל קראו לפרוייקט שלהם "יוניקס" (UNIX)
- יוניקס אשר יקרא קרנל (- kernel גרעין, ליבה). הקרנל הוא רק חלק קטן מהקוד שצריך להתאים לכל מערכת מוגדרת ומהווה את הבסיס של מערכת יוניקס. מערכת ההפעלה וכל שאר הפונקציות נבנו מסביב לקרנל בשפת תכנות עילית C. השפה פותחה במיוחד בכדי ליצור את יוניקס. על ידי שימוש בטכניקות חדשות היה קל לפתח מערכת הפעלה שיכולה לעבוד על הרבה סוגים של חומרה.
- מוכרי התוכנות התאימו את עצמם לזה במהרה, מאחר והם יכלו למכור פי עשרה כמעט ללא מאמץ. יוניקס עשתה מהלך גדול שעזר ליצור תאימות בין מערכות שונות.
- במשך שני העשורים הבאים הפיתוח של יוניקס המשיך. התוספו יכולות נוספות ועוד ספקי חומרה ותוכנה הוסיפו תמיכה ליוניקס למוצריהם.
- יוניקס נוסדה בהתחלה רק לסביבות גדולות עם מחשב-גדול, היה צריך לעבוד באוניברסיטה, בממשלה או בעבור חברה גדולה בכדי שתוכל להשתמש במערכת יוניקס.
- אבל מחשבים קטנים התפתחו, ובסוף שנות ה-80, לאנשים רבים היה מחשבים ביתיים. באותו זמן, היו כמה גרסאות של יוניקס שהיו זמינות לארכיטקטורות PC אך אף-אחת מהם לא היתה חופשית.

LINUX

- מטרתו של לינוס טורבאלדס היתה ליצור מערכת הפעלה חופשית אשר תהווה תחליף מלא ליוניקס המקורי עם הסטנדרט של יוניקס.
- הוא קרא לפרויקט שלו לינוקס משילוב של שמו הפרטי לינוס עם שילוב שם המערכת יוניקס.
- בימים ההם plug-and-play עדין לא הומצא, אבל אנשים רבים התענינו שיהיה להם יוניקס משלהם, כך שהיה זה רק מכשול קטן. דריוורים נהפכו לזמינים לכל סוגי החומרה, בקהילה המתפתחת בקצב מהיר. כמעט מיד שחומרה חדשה נהייתה זמינה, מישהו קנה אותה ושלח אותה למבחן של לינוקס. כאשר המערכת החלה נדרשת בהדרגה, משחררת עוד קוד פתוח לטווח גדול יותר של חומרה. כאשר מפתחי הקוד (coders) לא עצרו ב: PC כל חומרה שהם מצאו היתה שימושית ללינוקס.
- שנתיים לאחר הפרסום של לינוס טורבאלדס, היו בלינוקס כ-12,000 משתמשים. הפרויקט, שהפך להיות פופולרי אצל בעלי-התחביב, גדל באופן תמידי, כל התכונות של יוניקס נוספו במשך השנתיים הבאות, אשר בעקבותיהם נהפכה לינוקס למערכת הבוגרת אשר היא היום. לינוקס היא שכפול של יוניקס, מתאימה לשימוש בתחנות עבודה כמו לשרתים לטווח בינוני וגבוה.

קוד פתוח

- הרעיון מאחורי הקוד-הפתוח הוא מאוד פשוט: כאשר מתכנתים יכולים לקרוא, להפיץ ולשנות את הקוד, הקוד יתפתח. אנשים יכולים להתאים את הקוד, לטפל בו, לנפות שגיאות (debug it) והם יכולים לעשות זאת במהירות אותה תוכנה תהיה יותר גמישה ובאיכות טובה יותר מאשר תוכנה שפותחה על ידי שימוש באפיק רגיל, מאחר ואנשים בחנו אותה ביותר מצבים שונים מאשר מפתחים בתוכנה בעלת קוד סגור יכולים לדמיין.
- מיזם הקוד-הפתוח החל להבהיר זאת לעולם המסחרי, ובאופן איטי מאוד, מוכרי תוכנות-מסחריות החלו להבין את הנקודה, כאשר אקדמאים, ואנשי מקצוע שוכנעו כבר לפני 20 שנה, שזה הדרך ללכת בה, מוכרי תוכנות-מסחריות צריכים ישומים כמו האינטרנט בכדי שהם יבינו שהם יכולים להרוויח מקוד פתוח. כעת לינוקס התפתחה מהתקופה בה המערכת היתה אך ורק מערכת אקדמית, שימושית רק לאנשי מקצוע עם רקע טכני.



אדיאולוגיית קוד פתוח

מושג הקוד הפתוח

- קיימת תפיסת עולם הטוענת כי התוכנות שייכות להמונים, ועל כן כל החפצים בכך יכולים להיכנס לקוד ולערוך בו שינויים. זאת, בניגוד לקוד סגור, אשר מזוהה, על פי רוב, עם יצרני התוכנה הגדולים. לאדם הקונה תוכנות בקוד סגור, לרוב אין אפשרות פשוטה לצפות בפרטי התהליכים שמרכיבים את התוכנות שרכש, מכיוון שהם מוסתרים מפני המשתמש באמצעות הידור (אף שאין זה מחויב, ייתכן קוד ניתן לצפיה אך אסור לשינויים מצד הרישיון ובשל כך הוא אינו קוד פתוח).
- התפתחות אידאולוגיית הקוד הפתוח התהוותה במקביל למהפכה ממשית בעולם התוכנה. בעשורים האחרונים יותר ויותר תוכנות מפותחות מראש כקוד פתוח, ולא מעט בתי תוכנה מפתחים את מוצריהם בקוד פתוח. כך, למשל, משוחררת מערכת ההפעלה אנדרואיד ודפדפן כרום של גוגל, כ-40% מאתרי האינטרנט בעולם מבוססים על פלטפורמות בקוד פתוח. במשך השנים הפכה מהפכת הקוד הפתוח לגורם השראה מעבר לעולם התוכנה, והיא מתחילה לחלחל גם להתנהלות הניהולית בארגונים

- אז מה זה לינוקס?
- יש אנשים שחושבים שזה מערכת הפעלה כמו מאק ונידוס אז לא 😊
- לינקוס זה לא מערכת הפעלה זה קרנל שיש גם ווינדוס הוא נמצא בכל מערכת הפעלה
- הקרנל מתאם בין החומרה (מעבד, זיכרון) לבין התוכנה (שזה מערכת הפעלה והתוכנות של המחשב)
- אובונטו היא מערכת הפעלה מבוססת לינקוס כלומר משתמשות בקרנל של לינוקוס
- יש למעלה ממאה של מערכות הפעלה של לינוקס ואנחנו נתמקד באוטובטו מערכת הפעלה הכי פופורלית יש לה הרבה תמיכה ותמיכה רחבה במנהלי התקן שיש בשוק.

HOW TO INSTALL?

- אז איפה הקוד של לינוקס נמצא? איך משיגים אותו? מתקנים אותו על המחשב בצורה חנימית?
- ההתקנה של לינוקס היא ממש ממש שונה מהתקנה של תוכנה
- הקוד של לינוקס נמצא repository זה בעצם big data base או storage של כל הקוד הפתוח שאנשים מפתחים ומעלים אותו לשם. (אנשים שרוצים להתקין תוכנה לא צריך ללכת לאינטרנט ולחפש פשוט אפשר להוריד את זה מתוך ה storage - והוא גם כן ידע להוריד את כל החבילות שהוא צריך)
- אז איך מורידים את לינוקס קרנל נפתח את Firefox נכתוב בגוגל Linux kernel archive <https://www.kernel.org/>
- נגיע לגירסא האחרונה של לינוקס Latest stable version



METHOD #1: SEARCH AND INSTALL THE LATEST STABLE VERSION

The highest possible version is 4.10.0-27. To install it just type the following `apt-get` command/`apt` command:

```
$ sudo apt install linux-image-4.10.0-27-generic
```

OR

```
$ sudo apt-get install linux-image-4.10.0-27-generic
```

Sample outputs:

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-4.4.0-81 linux-headers-4.4.0-81-generic linux-image-4.4.0-81-g
Use 'apt autoremove' to remove them.
Suggested packages:
  fdutils linux-tools linux-headers-4.10.0-27-generic
The following NEW packages will be installed:
  linux-image-4.10.0-27-generic
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/20.1 MB of archives.
After this operation, 70.0 MB of additional disk space will be used.
Selecting previously unselected package linux-image-4.10.0-27-generic.
(Reading database ... 89746 files and directories currently installed.)
Preparing to unpack .../linux-image-4.10.0-27-generic_4.10.0-27.30~16.04.2_amd64
Done.
Unpacking linux-image-4.10.0-27-generic (4.10.0-27.30~16.04.2) ...
Setting up linux-image-4.10.0-27-generic (4.10.0-27.30~16.04.2) ...
Running depmod.
update-initramfs: deferring update (hook will be called later)
Examining /etc/kernel/postinst.d.
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 4.10.0-27-generic
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 4.10.0-27-generic
update-initramfs: Generating /boot/initrd.img-4.10.0-27-generic
W: mdadm: /etc/mdadm/mdadm.conf defines no arrays.
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 4.10.0-27-gener
run-parts: executing /etc/kernel/postinst.d/update-notifier 4.10.0-27-generic
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 4.10.0-27-generic
Searching for GRUB installation directory ... found: /boot/grub
Searching for default file ... found: /boot/grub/default
Testing for an existing GRUB menu.lst file ... found: /boot/grub/menu.lst
Searching for splash image ... none found, skipping ...
Found kernel: /vmlinuz-4.4.0-83-generic
Found kernel: /vmlinuz-4.4.0-81-generic
Found kernel: /vmlinuz-4.10.0-27-generic
Found kernel: /vmlinuz-4.4.0-83-generic
Found kernel: /vmlinuz-4.4.0-81-generic
Replacing config file /run/grub/menu.lst with new version
Updating /boot/grub/menu.lst ... done
```

You must reboot the Linux server, run:

```
$ rm *.deb
$ sudo reboot
```

Verify new version:

```
$ uname -mrs
Linux 4.10.0-27-generic x86_64
```

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 16.04.2 LTS
Release:        16.04
Codename:       xenial
```

To find out your current version of Linux kernel, run:

```
$ uname -mrs
Linux 4.4.0-83-generic x86_64
```

To find out the latest version of Linux kernel type:

```
$ apt-cache search linux-generic
```

Sample outputs:

```
linux-generic - Complete Generic Linux kernel and headers
linux-generic-lts-utopic - Complete Generic Linux kernel and headers (dur
linux-generic-lts-vivid - Complete Generic Linux kernel and headers (dum
linux-generic-lts-wily - Complete Generic Linux kernel and headers (dummy
linux-generic-lts-xenial - Complete Generic Linux kernel and headers (dur
linux-image-4.4.0-21-generic - Linux kernel image for version 4.4.0 on 64
linux-image-extra-4.4.0-21-generic - Linux kernel extra modules for vers
linux-generic-hwe-16.04 - Complete Generic Linux kernel and headers
linux-generic-hwe-16.04-edge - Complete Generic Linux kernel and headers
linux-image-4.10.0-14-generic - Linux kernel image for version 4.10.0 on
linux-image-4.10.0-19-generic - Linux kernel image for version 4.10.0 on
linux-image-4.10.0-20-generic - Linux kernel image for version 4.10.0 on
linux-image-4.10.0-21-generic - Linux kernel image for version 4.10.0 on
linux-image-4.10.0-22-generic - Linux kernel image for version 4.10.0 on
linux-image-4.10.0-24-generic - Linux kernel image for version 4.10.0 on
linux-image-4.10.0-26-generic - Linux kernel image for version 4.10.0 on
linux-image-4.10.0-27-generic - Linux kernel image for version 4.10.0 on
linux-image-4.4.0-22-generic - Linux kernel image for version 4.4.0 on 64
linux-image-4.4.0-24-generic - Linux kernel image for version 4.4.0 on 64
linux-image-4.4.0-28-generic - Linux kernel image for version 4.4.0 on 64
linux-image-4.4.0-31-generic - Linux kernel image for version 4.4.0 on 64
```

METHOD #1: SEARCH AND INSTALL THE LATEST STABLE VERSION

- ls_release -a
- uname -mrs (find current version of linux kernel)
- apt-cache search linux-generic (find lsv)
- sudo apt-get install linux-image-4.x.x-generic (get optional)
- rm *.dbe (reboot required)
- sudo reboot
- uname -mrs (verify new version)

METHOD #2: INSTALL MAINLINE KERNEL VERSION

Use the wget command to grab the files:

```
$ wget http://kernel.ubuntu.com/~kernel-ppa/mainline/v4.12.3/linux-headers-4.12.3-041203-generic_4.12.3-041203.201707210343_amd64.deb
$ wget http://kernel.ubuntu.com/~kernel-ppa/mainline/v4.12.3/linux-image-4.12.3-041203-generic_4.12.3-041203.201707210343_amd64.deb
```

To verify the mainline build binaries, run:

```
$ wget http://kernel.ubuntu.com/~kernel-ppa/mainline/v4.12.3/CHECKSUMS
$ sha256sum --ignore-missing -c CHECKSUMS
```

Sample outputs:

```
linux-headers-4.12.3-041203-generic_4.12.3-041203.201707210343_amd64.deb
linux-image-4.12.3-041203-generic_4.12.3-041203.201707210343_amd64.deb:
sha256sum: WARNING: 23 lines are improperly formatted
```

You must see OK for all downloaded .deb files. Next, you need to install .deb files using dpkg command:

```
$ sudo dpkg -i *.deb
```

You must reboot the Linux server, run:

```
$ sudo reboot
```

Verify new version:

```
$ uname -mrs
Linux 4.12.3-041203-generic x86_64
```

kernel.ubuntu.com/~kernel-ppa/mainline/v4.12.3/

v4.12.3 mainline build

These binary packages represent builds of the mainline or stable Linux kernel tree at the commit below:

v4.12.3 (0f803aa5b4617e77841e2d2e6fe9c5e096d3ad55)

To obtain the source from which they are built fetch the commit below:

git://git.launchpad.net/~ubuntu-kernel-test/ubuntu/+source/linux/+git/mainline-crack v4.12.3

and apply the following patches on top in the order below:

0001-base-packaging.patch
0002-debian-changelog.patch
0003-configs-based-on-Ubuntu-4.12.0-7.8.patch

This is summarized in the [SOURCES](#) file included with these binaries.

Build for amd64 succeeded (see [BUILD.LOG.amd64](#)):
linux-headers-4.12.3-041203_4.12.3-041203.201707210343_all.deb
linux-headers-4.12.3-041203-generic_4.12.3-041203.201707210343_amd64.deb
linux-headers-4.12.3-041203-lowlatency_4.12.3-041203.201707210343_amd64.deb
linux-image-4.12.3-041203-generic_4.12.3-041203.201707210343_amd64.deb
linux-image-4.12.3-041203-lowlatency_4.12.3-041203.201707210343_amd64.deb

Build for i386 succeeded (see [BUILD.LOG.i386](#)):
linux-headers-4.12.3-041203_4.12.3-041203.201707210343_all.deb
linux-headers-4.12.3-041203-generic_4.12.3-041203.201707210343_i386.deb
linux-headers-4.12.3-041203-lowlatency_4.12.3-041203.201707210343_i386.deb
linux-image-4.12.3-041203-generic_4.12.3-041203.201707210343_i386.deb
linux-image-4.12.3-041203-lowlatency_4.12.3-041203.201707210343_i386.deb

Build for armhf succeeded (see [BUILD.LOG.armhf](#)):
linux-headers-4.12.3-041203_4.12.3-041203.201707210343_all.deb
linux-headers-4.12.3-041203-generic_4.12.3-041203.201707210343_armhf.deb
linux-headers-4.12.3-041203-generic-lpae_4.12.3-041203.201707210343_armhf.deb
linux-image-4.12.3-041203-generic_4.12.3-041203.201707210343_armhf.deb
linux-image-4.12.3-041203-generic-lpae_4.12.3-041203.201707210343_armhf.deb

Build for arm64 succeeded (see [BUILD.LOG.arm64](#)):
linux-headers-4.12.3-041203_4.12.3-041203.201707210343_all.deb
linux-headers-4.12.3-041203-generic_4.12.3-041203.201707210343_arm64.deb
linux-image-4.12.3-041203-generic_4.12.3-041203.201707210343_arm64.deb

Build for ppc64el succeeded (see [BUILD.LOG.ppc64el](#)):
linux-headers-4.12.3-041203_4.12.3-041203.201707210343_all.deb
linux-headers-4.12.3-041203-generic_4.12.3-041203.201707210343_ppc64el.deb
linux-image-4.12.3-041203-generic_4.12.3-041203.201707210343_ppc64el.deb

Build for s390x succeeded (see [BUILD.LOG.s390x](#)):
linux-headers-4.12.3-041203_4.12.3-041203.201707210343_all.deb
linux-headers-4.12.3-041203-generic_4.12.3-041203.201707210343_s390x.deb
linux-image-4.12.3-041203-generic_4.12.3-041203.201707210343_s390x.deb

For further information see: <http://wiki.ubuntu.com/Kernel/MainlineBuilds>

	Name	Last modified	Size	Description
Parent Directory				-
	0001-base-packaging.patch	2017-07-21 07:44	14M	
	0002-debian-changelog.patch	2017-07-21 07:44	31K	
	0003-configs-based-on-Ubuntu-4.12.0-7.8.patch	2017-07-21 07:44	12K	
	BUILD.LOG	2017-07-21 08:54	12M	
	BUILD.LOG.amd64	2017-07-21 08:54	2.9M	

METHOD #2: INSTALL MAINLINE KERNEL VERSION

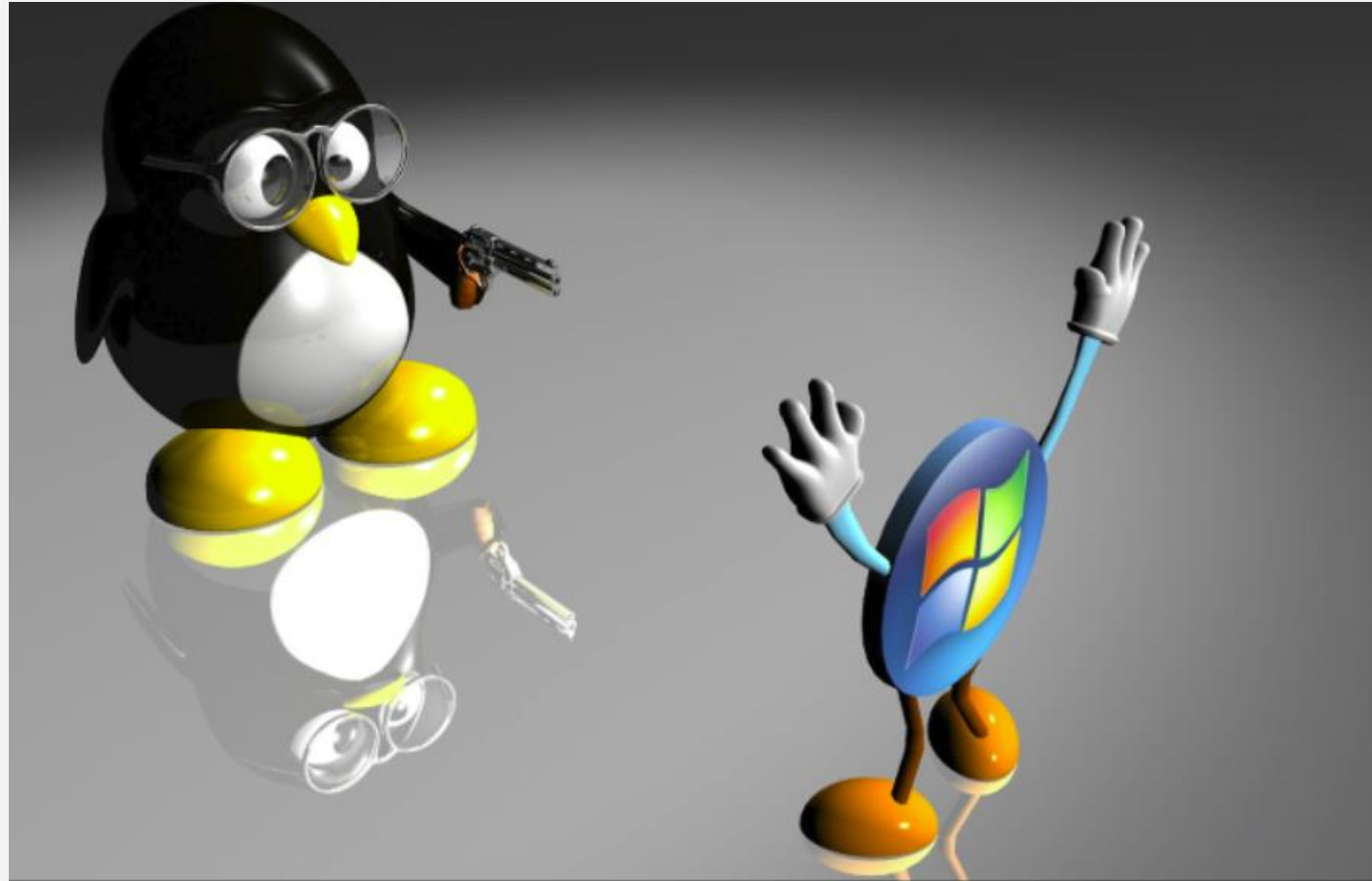
- search at kernelUbuntu.com
- wget <http://kernel.ubuntu.com/~kernel-ppa/mainline/vx.x/linux-headers-x.x.x-generic x.x.x....deb> -
- sudo dpkg -i *.deb
- sudo reboot
- uname -mrs

UBUNTU

פתרונות מחשוב

Comp^yoter

LINUX VS WIN



WHY LINUX IS BETTER?

בחר כיצד אתה רוצה ששולחן העבודה שלך יראה.



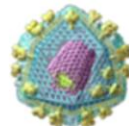
שמירה על קשר עין עם מזג האוויר



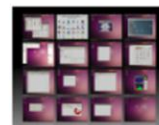
בשביל מה להעתיק תכנה באופן לא חוקי אם אפשר לקבל אותה בחינם?



תשכחו מוירוסים



יותר מדי חלונות? סביבות עבודה כבר ניסית?



האם המערכת שלך סובלת מאי-יציבות?



קפצו אל הדור הבא של שולחנות העבודה.



לינוקס מגינה על המחשב שלך



צריכים תכנה חדשה? אל תטרחו לחפש ברשת, לינוקס תדאג לכם.



דיווח באגים



עזרו לארצות זרות, ולשלכם



מדוע חלונות איטית יותר מיום ליום?



חיים חדשים למחשבך הישן



תמיכה לא מוגבלת ובחינם להנאתך



עדכן את כל התוכנות שלך בקליק אחד



השתמשו במסג'ר AIM, אייסיקיו וג'אבר בתוכנה אחת



חירות!



שחק במאות משחקים, בחינם!



תרמו את חלקכם למען איכות הסביבה



עזבו אתכם ממנהלי התקנים

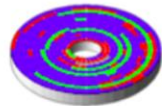


WHY LINUX IS BETTER?

למה לשלם 900 שקל על מערכת הפעלה



מרגישים שהחיים הדיגיטליים שלכם שבורים לרסיסים?



לא עוד תכניות זבל.



אין בלגן בתפריט ההתחלה שלך



אין פרצות סתרים בתוכנה שלך.



קבל גן מוסיקה מדהים

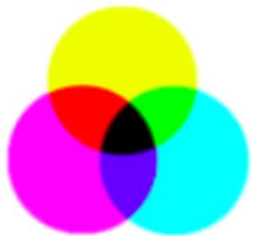


אחרי שכבר התקנת את המערכת, למה עליך בכל זאת להתקין עוד דברים?



WHY WIN IS BETTER?

ספרים ותעשיית ההוצאה לאור



משחקים



חומרה



תוכנה קניינית



WHY LINUX IS BETTER?

- אחרי התקנת WIN יש צורך בהתקנת תוכנות רבות לאחר ההתקנה
- מתקין את אופיס באופן אוטומטי תואם (מסמך וורד ב WIN נפתח בלינוקס יש תאימות מלאה)
- בלינוקס נגן וידאי צריבה מייל תוכנות תקשורת פסיבוק מגיעות כבר עם מערכת הפעלה
- חירות – ליצור לעצמם כל מה שרוצים לעשות קוסטמציזה לעשות הכול לשנות צבע ומראה
- חינם – למה לשלם כמה מאות עבור חינום אם WIN באמת יותר טוב זה לא רק כסף מקבלים יותר טוב ובחנם למה אנשים משתמשים WIN כי אנשים מפחדים משינויים
- מגינה על המחשב שלך- לא מפתחים וירוסים ללינוקס ולא ניתן לעשות שינויים במערכת בלי לבקש סיסמא אז אי אפשרי יש יותר הבטחה, הרבה חברות משתמשים בלינוקס בגלל הבטחה, וגם השרתים ברוב העולם מבוססי לינוקס
- התקנה תוכנה חדשה למשל אפשר להוריד מהחנות שהיא מתקשרת עם האינטרנט
- – למשל התקנה VLC ויש כפתור להתקנה ומבקש סיסמא – לא ניתן לעשות שום דבר בלי לבקש סיסמא , אין מה לפחד מילדים שיתקינו דברים או WIN שיכול להורס את המערכת בלי סיסמא
- למה להפר את החוק – אם אפשר לקבל בחינם כמה אנשים יכולים להגיד שהמערכת הפעלה שלהם חוקית??
- הלינוקס דריבר מגיע עם כל הדריברים בפנים מבונה בתוך הקרנל
- איטיית ב WIN שמצריך אחיו דיסק
- מערכת סובלת מאי יציבות – עדין אחרי שמתקנים צריך לעשות הפעלה חדשה לא יציבה לא יכולה לפעול שבועים בלי הפעלה מחדש בלי קרסה, לינוקס יכולה לפעול שנים
- תשכחו וירוסים
- האם אתם מריגשים שהחיים שלכם שבורים לרססים – WIN מצריך איחוי דיסק בשביל שמערכת הפעלה תשמור על המהירות שלה
- קפוץ לדור הבא של שולחן עבודה- ניתן לעשות אפקטים מרהיבים
- צריך תוכנה חדשה – לא צריך לחפש
- איך מתקנים תוכנה Ubuntu software center כלומר מרכז התוכנה
- רוב העולם משתמש ב WIN לכן אקרים מפתחים וירוסים יותר ל WIN

WHY LINUX IS BETTER?

- יש כמה אנשים בעולם שמחזיקים את הקוד של לינוקס, כל זמן מסוים יוצאת גריסא חדשה KERNEL ויש כמעט את כל התוכנות חינם, הכול כמובן מתוחזק (חוכמת ההמונים)
- לינוקס הוא הרבה יותר security דבר ראשון בגלל זה open source - כמו שאמרנו זה חוכמת ההמונים מלא אנשים עובדים עליו מתעניינים בו וזה מוסיף לו מנגנוני הבטחה וכמובן יותר הגנה וירוסים, בוינדוס הוריסוים יכולים לחדור לתוכנות ולהשבית את המחשב בשונה מלינוקס
- עוד אחד היתרונות הוא יכול להתאים את עצמו לשימושים רבים להיות SERVER להיות FIRE WALL בשונה וינדוס שצריך לקנות win server גירסא מיוחדת
- לינוקס הוא אוניברסאלי יותר, כמשהו אומר שהוא משתמש בלינוקס אז יש לו הרבה מאוד גרסאות (בעצם אתה צריך לשאול אותו באיזה אתה משתמש)

הפצות לינוקס



WHY LINUX IS BETTER?

- דבר נוסף זה התאמה - אפשר להתאים הכול אפילו את הכתב של קוד. דבר זה בעצם מבוסס על הרעיון שזה קוד פתוח, יש לך את הקוד תעשה בו כרצונך
- אמינות – crash הרבה פחות מצוי, כמובן שזה גם יכול לקרוא (כשאתה מפתחים קוד ב kernel space ואז גורמים ל kernel panic)
- – אנחנו מדברים על משתמשים שלא נוגעים ב kernel מרצים תוכנות רגילות ב user space אז כמובן כמעט לא מגיעים לקריסות של המחשב
- בונידוס אם לא עושים restart אחרי שמתקנים תוכנה הוא לא מתנהג כמו שצריך בשונה בלינוקס לא צריך לעשות את זה, זה פשוט עובד!
- פרטיות – וינדוס אוספים הרבה אינפורמציה עלינו (כאשר מתקנים וינדוס הוא שואל הרבה שאלות בעצם כאשר שמים לב המטרה היא לאסוף כמה שיותר אינפורמציה) לעומת לינוקס שזה קוד פתוח ואין מנגנון של איסוף אינפורמציה.

WHY LINUX IS BETTER?

- יתרונות WIN

- התאמה – לא כל יצרני התכונות מתאימות אותם לכל מערכות הפעלה לכן יש תוכנות שאין אותם בלינוסק, adobe או תוכנות ודיאו שאין אותם בלינוקס יש הלטרנטיבות דומות אבל הם לא אותו דבר ולא נותנות אותם ביצועים

- התקנות בוינדוס יותר קלות, מבחינת משתמש ומבחינת תלויות dependency

- אז למי מתאים לינוקס?

- אז נענה ככה לאנשים ממוצעים יותר יהיה מתאים ונוח ונידוס אם הם יכולים לקבל את אותם ביצועים שהם צריכים בממשק קל ונוח, אבל לאנשים שידענים מה הם רוצים יתאים לינוקס

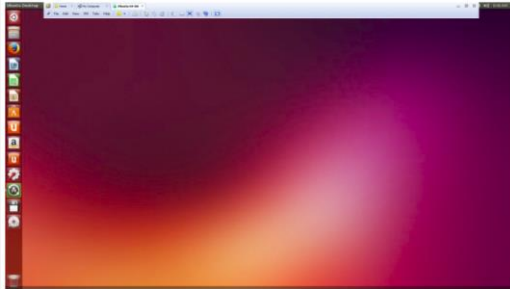
- אנחנו נתמקד ב UBUTU שהיא המערכת הפעלה הכי פופלרית של לינוס הכי פורפלרית יש לה הרבה תמיכה בכל מנהלי התקן שיש היום בשוק



הפצות נפוצות של לינוקס

• Ubuntu

- היא הפצה של לינוקס והמפורסמת בהם גם בין משתמשי קצה וגם בשרתים בגלל גרסת ה Desktop שלו והגרסה המאוד ידידותית למשתמש הוא הפך במרוצת הזמן לפופולרי ביותר אובונטו מבוססת על Debian



- היא משחררת גרסאות כל חצי שנה בגרסאות LTS -Long Term Support מבטאים אותו אובונטו לאובנטו יש

- גרסת Desktop
- גרסת Server שהיא CLI
- גרסת ה Live CD שהיום לא קיים יותר

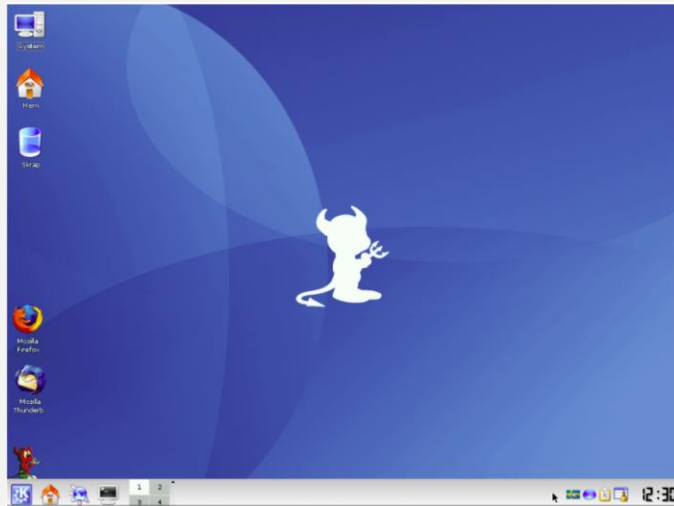
המניפיסטו של קבוצת המתכנתים של אובונטו היא שתוכנה צריכה להיות חינם ללא עלות

- קליק ימני ואפשרויות יציג את מאפייני התיקיה עליה לחצתם, אז מה שונה? השוני הוא רב בכלל הפעולה הפנימית של המערכת מה שאתם לא רואים אבל מבחינת המראה הויזואלי הגרפי זה דומה בגרסאות החדשות של אובונטו הקדישו יותר זמן ופיתוח על עיצוב וידידותיות למשתמש

FREEBSD

FreeBSD •

- זה קיצור של Free Berkeley's Distribution Software (יש עוד כמה פירושים...) מערכת הפעלה זו נכתבה לפני 20 שנה וכמובן מבוססת Unix המערכת כל כך הצליחה אז שהיא הפכה לחלקים שלמים מתוך מערכות הפעלה של אפל OSX וחברת המתגים והנתבים Juniper
- השימושים של FreeBSD הם משרתי אינטרנט, שרתי דואר, שרתי DNS שימוש כראוטרים ו DHCP ואפילו Firewall
- FreeBSD נקראת XWindows System והיא גרסת הדקסטופ



FEDORA

Fedora •

- היא מערכת הפעלה אחת מהפצות לינוקס ובין הפופולריות ביותר, הפצה זו היא בבעלות REDHAT
- היא העדכנית ביותר ומתעדכנת מהר מאוד בשאר להפצות האחרות, התקנות בה הם לא דבר שבשגרה ודרוש ידע קל להפעיל פיצ'רים מסויימים בה
- פדורה היא מערכת הפעלה בגרסת Desktop בלבד ואין לה גרסת Serer
- לשולחן העבודה של פדורה קוראים KDE השם החדש ליתר דיוק הוא KDE Plasma Workspace 4.7 אפשרויות כמו וירטואליזציה, ענן וגם לפדורה גירסה חיה של ההפצה Fedora Live Desktop
- התוכנות הנפוצות הקיימות גם על אובונטו וגם על פדורה

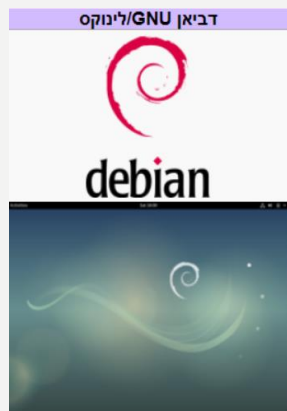


- * Chorme & Firefox
- * GIMP סוג של פוטושופ
- * VLC נגן וידאו
- * Brasero disc burner
- * Thunderbird דוא"ל דומה לאוטלוק

Debian •

– דביאן גנו/לינוקס היא הפצת לינוקס קהילתית המפותחת על ידי קבוצה גדולה של מתנדבים מרחבי העולם. בנוסף לגרעין מערכת ההפעלה, כוללת המערכת אלפי חבילות תוכנה חופשיות נוספות, מיעוט התוכנות משולבות בבסיס המערכת ורובן ניתנות להתקנה לפי בחירת וצורך המשתמש מאתרי מקורות של דביאן. דביאן מתאפיינת בהיצמדותה לעקרונות יוניקס והתוכנה החופשית, בשיתוף הפעולה בפיתוח התוכנות ובתהליך בקרת האיכות שלה, ובאמנה החברתית העומדת במרכז ומתווה את מטרותיה.

– דביאן נוסדה על ידי איאן מרדוק בשנת 1993, והיא אחת מהפצות הלינוקס הפעילות הוותיקות ביותר. דביאן היא הפצה פופולרית ומשפיעה מאוד, הפצות לינוקס רבות מתבססות על המבנה היסודי של דביאן, ובהן: אובונטו, מינט ו- Tails. דביאן יכולה לשמש הן כמערכת הפעלה לשרת והן כמערכת הפעלה למחשב אישי סטנדרטי. היא תומכת במגוון ארכיטקטורות חומרה.



MINT & CENTOS



• Linux Mint

- לינוקס-מינט היא גרסה של לינוקס אשר (לרוב) נוחה יותר למשתמשי חלונות מאשר אובונטו. היא גרסה יותר "שמרנית", במובן הזה שיש פחות עידכוני תוכנה (ולכן פחות הזדמנויות לתוכנה לקרוס). מינט מגיעה מוכנה "הישר מהקופסא", עם רוב הדברים שצריך כדי פשוט להשתמש בה (עם מינימום בעיות של חומרה לא מזוהה, קבצי וידיאו לא מזוהים, וכדומה). ובגלל שמינט מבוסס על דביאן ואובונטו, ניתן להתקין עליו עשרות אלפי תוכנות אחרות שנבנו למערכות הללו
- היא הפצת לינוקס המבוססת על ההפצה אובונטו (עם מספר חבילות שנלקחות ישירות מדביאן). יש גם גרסה בשם לינוקס מינט מהדורת דביאן אשר מבוססת על דביאן. לינוקס מינט היא אחת מהפצות הלינוקס הפופולריות ביותר מבין ההפצות השולחניות, בין השאר משום שהיא ידידותית מאוד למשתמש וכוללת בתוכה מספר רב של תוספי מולטימדיה ורכיבי מערכת אחרים, אשר על המשתמש להוריד ולהתקין בעצמו בהפצות אחרות.
- תומכת במספר ממשקים של שולחנות עבודה, גרסה חדשה של ההפצה יוצאת מידי שישה חודשים.



• Centos

- לצורך המחשה, החברה Red Hat פיתחה מערכת הפעלה מבוססת Linux לשרתים, אך השימוש בה בתשלום. לעומתה, החברה Centos מציעה גם היא מערכת זהה אך בחינם ועדיין בעולם משתמשים במערכת ההפעלה Red Hat

הפצות מסחריות של לינוקס



RedHat •

- Red Hat Enterprise Linux בקיצור RHEL היא הפצת לינוקס לשרתים ובין המפורסמות בעולם ובמיוחד בעולם השרתים ההפצה היא בתשלום, למה בתשלום? בגלל התמיכה שלהם למערכ
- היא מבוססת על Fedora ואף ההפצה Fedora שייכת ומומנת על ידם היא מכילה את רוב הפיצ'רים שצריך בעולם השרתים ולכן הוכרזה שמערכת ההפעלה הלינוקסאית הטובה ביותר בתחום הארגוני.
- רוב החידושים המבוצעים בפדורה מגיעים בסופו של דבר לאחר בדיקה של הקהילה ושיפור של אותו מנגנון אל RHEL ובכך עומדת RHEL בקצב ההתפתחות של הטכנולוגיה, היתרון של RHEL הגדול מכולם הוא התמיכה לארגונים אף ארגון לא היה רוצה להקים שרתים מבוססי Web ופתאום שיש בעיה וכל שעה שעוברת הם מפסידים ואין להם עם מי לדבר... ולכן RHEL הפכה להפצה הגדולה ביותר לארגונים בשנת 2008 רכשה Redhat חברת ישראלית בשם קומראנט בגלל מערכת הווירטואליזציה שלהם הנקראת KVM כיום המפיצה הבלעדית של Redhat היא חברת מטריקס בישראל



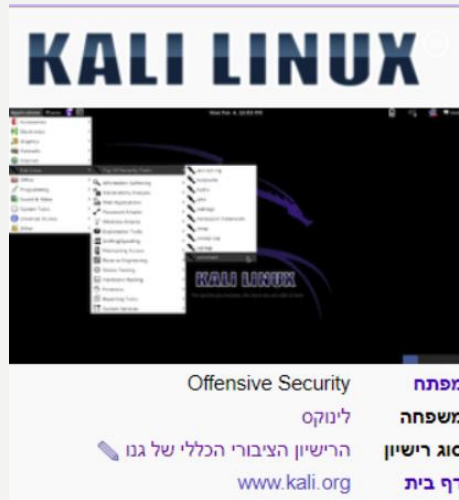
• Open SUSE

- openSUSE לשעבר SUSE Linux ו-SuSE Linux Professional
- היא הפצת לינוקס המפותחת על ידי חברת SUSE Linux וחברות אחרות.
 - במערכת נעשה שימוש נרחב ברחבי העולם.
 - התמקדות הפיתוח שלה הוא יצירת כלי קוד-פתוח שמישים עבור מפתחי תוכנות ומנהלי מערכות, תוך מתן שולחן עבודה נוח וסביבת שרת עשירה.

openSUSE נוצרה על ידי חברת Novell לאחר רכישת לינוקס סוזה עבור 210 מיליון דולר ב-4 בנובמבר 2003.

הפצות מיוחדות

• Kali



קאלי (נפוצה בצירוף קאלי לינוקס) נגזרת של דביאן, המיועדת לזיהוי פלילי דיגיטלי ולבדיקות חדירות ופופולרית בעיקר בקרב האקרים. היא מתוחזקת וממומנת על ידי חברת "אופנסיב סקיוריטי".

פותחה על ידי חוקר אבטחת המידע מתי אהרוני, ודבון קיירנס מ"אופנסיב סקיוריטי", על בסיס בקטראק, ההפצה הקודמת שלהם שגם התמקדה בהאקינג ובדיקות חדירות.

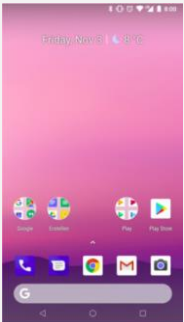
קאלי מופצת כשעליה מותקנים מראש תוכניות בדיקות חדירה רבות

קאלי יכולה להתחל מתוך דיסק קשיח, תקליטור אתחול, או החסן נייד.

ההפצה ניתנת להורדה באימג'ים בגרסת 32 ו-64 ביט, לשימוש על מחשבים PC וכן על כרומבוק של סמסונג

אנדרואיד

- אנדרואיד Android היא מערכת הפעלה המיועדת לסמארטפונים, טאבלטים, טלויזיות חכמות, שעונים חכמים ולמכונות ומבוססת על ליבת לינוקס. היא מופצת על ידי חברת גוגל בשיתוף פעולה עם Open Handset Alliance.
- מערכת הפעלה זו מעוצבת במיוחד לשימוש בטלפונים חכמים מבוססי מסך מגע ומחשבי לוח (טאבלטים). לקראת סוף שנת 2010, אנדרואיד הפכה למערכת ההפעלה הנפוצה ביותר בעולם לטלפונים חכמים. על פי IDC, ב-2012 היא החזיקה נתח של 70% מכלל שוק הטלפונים החכמים העולמי (ביחס לכ-20% למתחרה המרכזית iOS של חברת אפל) ובמהלך השנה נמכרו כמעט 500 מיליון מכשירים חדשים מבוססי אנדרואיד. סמארטפונים מפורסמים עם אנדרואיד
 - הם מכשירים בסדרת "גלקסי" של סמסונג ושל חברת LG, סוני או של חברות סיניות וגם חברת מוטורולה ו-Google Nexus
- אנדרואיד פותחה לראשונה בידי חברת הזנק בשם Android Inc שנוסדה ב-2003, ובשנת 2005 נרכשה על ידי Google הגרסה הראשונה של מערכת ההפעלה הוכרזה ב-5 בנובמבר 2007 עם הקמתו של Open Handset Alliance תאגיד של 86 חברות חומרה, תוכנה ותקשורת, שהוקדש למטרת קידום סטנדרטים פתוחים עבור מכשירים ניידים. המכשיר הראשון מבוסס אנדרואיד נמכר באוקטובר 2008. אנדרואיד נועדה בתחילה למצלמות חכמות.
- גוגל מפיצה את הקוד של אנדרואיד כקוד פתוח, תחת רישיון אפאצ'י. פרויקט הקוד הפתוח של אנדרואיד (Android Open Source Project) (AOSP) שמובל על ידי Google הוא הפרויקט האחראי לתחזוקה ולפיתוח של מערכת ההפעלה. בנוסף, לאנדרואיד יש קהילה גדולה של מפתחים אשר כותבים אפליקציות המגדילות את הפונקציונליות של המכשירים והן נכתבות בדרך כלל בשפה מותאמת אישית של Java את האפליקציות ניתן להוריד דרך Google Play או אתרי צד שלישי. נכון ליולי 2013, ישנן יותר ממיליון אפליקציות זמינות לאנדרואיד, וסך ההורדות מחנות האפליקציות Google Play מוערך בכ-50 מיליארד



IOS VS ANDROID

- נהלים ועובדים רבים מבצעים כיום את עבודתם בכל מקום באמצעות סמארטפונים. מגמת **BYOD** (Bring Your Own Device) תופסת תאוצה, בצד הצורך להחליט איזה סוגי **מכשירי סלולר** ניתן להכניס לתוך כתלי העסק כדי לאפשר למחזיקים בהם לעבוד מרחוק במערכות התפעול של העסק, החל מדוא"ל וכלה ביישומים עסקיים ייחודיים. כיום, הבחירה מצטמצמת בין מערכת ההפעלה מבית **גוגל** לזו מבית **אפל**. המערכות הוותיקות נמצאות בנסיגה ויציאה מהירה מהשוק הנייד, למשל אלו של בלקברי ונוקיה, שהיו נפוצות בעשור הקודם בקרב מנהלים ועובדים בארגונים ובעסקים.

ההבדלים בין מכשירים המריצים מערכת ההפעלה של אפל – iOS לבין אלה המריצים מערכת הפעלה של גוגל – Android מאוד ניכרים.

מערכת ההפעלה של אפל היא מערכת קניינית וסגורה. המשמעות של אמרה זו די ברורה: **רק אפל** מייצרת ומספקת מכשירים עם מערכת ההפעלה שלה. היישומים במערכת ההפעלה הזו הם רק אלו, שזכו לאישור של אפל ועמדו בקריטריונים נוקשים שאפל מכתיבה.

יש הטוענים, שבכך מקבלים המשתמשים מערכת הפעלה יציבה יותר ובטוחה יותר. לעומת זאת, **מערכת ההפעלה אנדרואיד** היא מערכת הפעלה פתוחה וחופשית. כל יצרן מכשירי סלולר וטאבלטים יכול לאמץ אותה. לכן, קיים בשוק מגוון רחב של יצרנים המספקים מכשירים ניידים עם מערכת ההפעלה אנדרואיד. בנוסף, כל אחד יכול לפתח יישומים לאנדרואיד, בלי לקבל על כך אישור מראש או בדיעבד מגוגל.

יש עוד הבדל אחד חשוב בין המערכות המתחרות הללו: **העלות**. לעסקים מדובר בנתון מאוד חשוב בתהליך קבלת ההחלטות. ננתח אותו מיד ואף נמליץ כיצד לקבל החלטות בנושא זה.

עפ"י חברת המחקר פורסטר, מכלל מיליארד מכשירי סמארטפון, שיהיו בעולם ב-2016, כ-350 מיליון סמארטפונים יהיו של עובדים המשתמשים בהם במקום העבודה לצרכי העבודה, ע"ג הרשת הארגונית. כמחציתם יהיו מכשירים, שהעובדים הביאו עימם מהבית BYOD לכן, חשוב לשמר את הרצון של העובדים להמשיך ולהשתמש ביישומים הפרטיים במכשירים שלהם, גם כשהמכשירים כפופים ונשלטים ע"י מדיניות האבטחה הארגונית.

