

בין תוכנה חופשית לכלכלה חופשית: כסף בעולם הקוד הפתוח

גלעד בן-יוסף (benyossef.com)

גלעד בן-יוסף, בעל תואר B.A. במדעי המחשב מהמכללה האקדמית של ת"א יפו, ממייסדי קודפידנס בע"מ (codefidence.com), ודמות מוכרת בקהילת הקוד הפתוח המקומית, מחבר ותורם לכמה פרויקטי קוד פתוח. הוא ממייסדי עמותת "המקור", והיוזם והמארגן הראשי של כנס אוגוסט פינגווין – כנס קהילת הקוד הפתוח בישראל. מאמר זה מבוסס בחלקו על מאמר קודם בשם "קוד פתוח, מדריך למשתמש" שפורסם במגזין היי-טק.

אימרה ידועה היא שמי שלא היה סוציאליסט בנעוריו, אין לו לב, ומי שלא התפקח מהסוציאליזם בבגרותו, אין לו ראש. הרעיון של תוכנה חופשית, שממנו התפתחה תנועת הקוד הפתוח, מדגיש את מושג ה"חופש", אבל ממעט לומר מדוע מושג החופש חשוב כל-כך. ללא הסבר של חשיבות חופש התוכנה, ובמיוחד חשיבותו הכלכלית, ניתן לחשוב בטעות שתוכנה חופשית נופלת באותו פח כמו סוציאליזם: רעיון שנשמע טוב על הנייר, אבל אינו מעשי משום שהוא אינו מסוגל ליצור מערכת כלכלית ברת קיימא.

במאמר זה אנסה להסביר מדוע אין הדבר כך ולתאר את המודל הכלכלי העומד מאחורי תנועת הקוד הפתוח. באופן מפתיע, המודל הזה מוכר לחברה האנושית מזה זמן מה ואף יש לו שם: קוראים לו קפיטליזם, או כלכלת השוק החופשי. כדי להבין את הקשר בין השניים, נתחיל בתיאור ההגיון העומד מאחורי האינטראקציה הכלכלית העומדת ביסוד פיתוח תוכנה חופשית ופתוחת קוד, ונמשיך בשאלה מיליון הדולר: איך עושים מזה כסף.

לפני שנתחיל, ראוי לציין שהאמור לעיל איננו חומר עיוני אקדמאי בלבד. מזה למעלה משנה אני ושותפי עובדים בחברה שהקמנו, בשם קודפידנס בע"מ, והמבוססת על העקרונות המפורטים כאן וכל פרנסתינו עליה. גם איננו היחידים בעולם (ואף לא בארץ) המתפרנסים מהתחום. נכון, יש להניח שלעולם לא נתחרה בהכנסה השנתית של ביל גייטס, אבל בעידן שבו הזכות לעבוד איננה מובנת מאליה, אני וחברי מרגישים בטוחים יותר מבחינה פיננסית מהממוצע, תודות לתוכנה חופשית וקוד פתוח. נשמע מעניין?

קוד פתוח: מדריך למשתמש

'קוד פתוח' הינה דרך לניהול פרויקטי פיתוח תוכנה בצורה פתוחה ומבוזרת, המאפשרת שיתוף פעולה של גורמים רבים ליצירת המוצר הסופי. גורמים אלו אינם חייבים להיות מאותו ארגון או חברה. כדי להמחיש את דרך הפיתוח המקבילית והמבוזרת שעומדת בבסיס רעיון הקוד הפתוח, כדאי לחשוב על ההקבלה בין פיתוח תוכנה מקבילי להרצה של תוכנה מקבילית. כדי שנוכל להריץ תוכנה באופן מקבילי ולהשיג זמני חישוב מיטביים יש צורך במספר מרכיבים:

ראשית, יש לנו צורך בבעיה הניתנת למקבול, דהיינו חישוב הניתן לפרק למספר רב של חישובים קטנים יותר, ובעלי תלות מוגבלת אחד בשני. לדוגמה, עיבוד תמונה הינה בעיה הניתנת למקבול בקלות רבה, מפני שאנו יכולים לפרק את התמונה למקטעים קטנים, ולעבוד על כל אחד מהם בנפרד. בנוסף, דרושים לנו כמובן מחשבים רבים לביצוע העבודה.

לבסוף, אנו זקוקים לאמצעי תקשורת יעיל בין המחשבים. גם כאשר הבעיה ניתנת למקבול, עדיין יש צורך בתקשורת טובה בין כל רכיבי התוכנה השונים הרצים במקביל לתיאום ביצוע החישוב: הרי לא היינו רוצים שקטע בתמונה יעובד פעמיים בשל חוסר תיאום בין רכיבי התוכנה.

באופן מפתיע, הרכיב האחרון הוא במקרים רבים הבעייתי ביותר בין שלושת הדרישות. אם הרווח שאנו משיגים ממקבול הבעיה שלנו קטן מהתקורה, או overhead בלע"ז, שאנו נאלצים לשלם עבור תיאום חלקי התוכנית השונים הרצים במקביל, התועלת שבמקבול נתונה בספק. כפי שנראה מיד, כך הוא הדבר גם בפיתוח מקבילי של תוכנה.

ובכן, כיצד מתייחסים שלושת המרכיבים הנ"ל לפיתוח תוכנה מקבילי?

השיעור הראשון שלומד כל מתכנת, הוא שכדי לשלוט בסיבוכיות פיתוחית של תוכנה, יש לבנות תוכנה באופן מודולורי ככל האפשר, כקבוצה של "קופסאות שחורות" המתקשרות ביניהם בממשק מוסכם. מתיאור זה נשמע כמעט ברור מאליו שפיתוח תוכנה הינה בעיה הניתנת למקבול ואכן כך הוא. גם מחשבים וחשוב מכך מתכנתים, אינם נמצאים במחסור בעולם.

נותר לנו איפה לפתור רק את בעיית התקשורת, או התיאום, בין מרכיבי המערכת השונים, הלא הם המתכנתים. מסתבר שדווקא כאן אנו נתקלים במחסום קשה לביקוע. את המחסום ניסח לראשונה באופן פורמלי פרדריק פ. ברוקס ג'וניר, המנהל שהיה אחראי על צוות הפיתוח של מערכת ההפעלה OS/360 של IBM, בספרו המאלף "The Mythical Men Month". הוא קבע, כתוצאה מניסיונו בניהולו של המאמץ הפיתוחי אדיר הממדים הזה, את מה שנודע מאז כ- 'חוק ברוקס': "הוספת מתכנתים לפרויקט פיתוח תוכנה מאחר תגרום לאיחור גדול יותר".

הסיבה לחוק ברוקס היא הרלוונטיות לעניינו. ברוקס מצא כי בעוד התועלת הנובעת מהוספת מתכנתים לפרויקט פיתוח תוכנה עולה בטור הנדסי כתלות במספר המפתחים העובדים על הפרויקט, הרי שהתקורה של התקשורת בין המפתחים הדרושות לתיאום העבודה עולה בצורה מעריכית. לכאורה, חוק ברוקס מנבא כי כל נסיון לנהל פרויקט פיתוח תוכנה מבוזר יקרוס תחת משקל התקשורת בין המתכנתים. כיצד איפה מצליחים פרויקטי פיתוח אדירי-ממדים כלינוקס המונה מאות אלפי משתתפים בכל העולם במאות גופים, חברות וארגונים, לייצר מערכת הפעלה מיטבית כל-כך?

הפתרון לסוגיה נעוץ בהבחנה, כפי שנוסחה בפני בבהירות יוצאת דופן על ידי פיטר אוסטרין, דירקטור פיתוח בחברת "סיליקון גרפיקס", כי "מפתחי תוכנה עוסקים בתקשורת: 5% תקשורת בין אדם למכונה ו-95% תקשורת בין אדם לאדם". אם נצליח איפה להוריד את המקדם של עלות התקורה של תקשורת בין אישית בין המפתחים בפרויקט לרמות נמוכות מאוד, נמוכות עד כדי כך שלמרות הגידול המעריכי, העלות הכוללת של התקורה הזאת, תהיה נמוכה מהתועלת הצומחת לנו ממספר המפתחים הרב העובד על הפרויקט, נצליח לגזור ערך ממקבול הפיתוח.

אז איך מורידים עלויות של תקשורת לרמות מינימליות בצורה סקלבילית (ניתנת להרחבה)? רשת האינטרנט מורה לנו את התשובה – אם תקשורת משמעה העברת מידע בין שתי צדדים, הרי שעלינו לדאוג לקיומו של פרוטוקול

המסדיר את התקשורת האמורה. על הפרוטוקול להיות כללי וברור ככל הניתן ועליו להבטיח נגישות שווה לערוץ התקשורת לכל הצדדים לה, ממש כמו שפרוטוקול התקשורת של רשת האינטרנט tcp/ip ממלא את המטרות האמורות על רשת האינטרנט.

המימוש של פרוטוקול התקשורת בין מפתחי תוכנה הוא עקרונות התוכנה החופשית כפי שנוסחו לראשונה על ידי ריצ'רד סטולמן, המייסד של פרויקט GNU והמנסה הראשי של רשיון השימוש הכללי הפומבי ה-GPL. מילותיו הראשונות של רשיון שימוש זה הינן: "הרשיונות למרבית התוכנות מכוונים לשלול את חירותך לחלוק בהן ולשנות אותן. בניגוד לכך, הרשיון הציבורי הכללי של GNU מיועד להבטיח את חירותך לחלוק תוכנה חופשית ולשנות אותה, כדי לוודא שהתוכנה היא חופשית לכל משתמשיה".

תנאי הרשיון הברורים והשיווייוניים מבטיחים כי כל אחד רשאי לעשות שימוש בתוכנה, לשנות, להפיץ ואף למכור אותה אבל לא לטעון לבעלות קניינית עליה, הם פרוטוקול המסדיר את התקשורת ברמה השמינית, המדומה, של מודל שבע שכבות התקשורת של OSI, הלא היא שכבת הפוליטיקה או האינטראקציה האנושית, ממש כמו שפרוטוקול ה-tcp/ip עושה כן עבור שבע השכבות ה"מסורתיות" של מודל זה.

על ידי שימוש ב-GPL או ברשיונות תוכנה חופשית אחרים בעלי עקרונות דומים, מסדירים השותפים המפתחים יחדיו תוכנה קוד פתוח את האינטראקציה ביניהם. ללא רשיונות אלו, כל שיתוף פעולה בין מפתחים מארגונים שונים היה דורש הסדרים חוקיים, חוזים, הסכמי סודיות, פגישות בין מנהלים ועורכי דין ועלות התקורה היתה מרקיעה שחקים והופכת כל שיתוף פעולה גדול ממדים לבלתי אפשרי.

תודות ל-GPL ועקרונות החופש והשיווייוניות שהוא מבטיח למשתתפים במשחק, יכולים להיות כל המשתתפים בטוחים בהגינותו של המשחק ללא צורך בהבטחות נוספות ומאפשר ליצור ארגון אד-הוק, או במילים אחרות קהילה של מפתחים ומשתמשים.

מפני שהקוד פתוח ומפני שההסדרים החברתיים משפטיים תומכים בכך, קהילה כזאת יכולה לעסוק בפיתוח משותף של תוכנה ולקיים מעקב מתמיד על איכותה ובטיחותה תוך שימוש באותו עקרונ של peer review העומדים ביסוד השיטה המדעית. או כמו שניסח זאת לינוס טורבלדס, בחוק לינוס, שהוא כעין

משלים לחוק ברוקס: "Given enough eyeballs, all bugs are shallow" – "בהינתן מספיק עיניים, כל הבעיות בתוכנה הינן שטחיות".

איפה הכסף?

נקודה אחת שעולה מיד למשמע התיאור הכביכול אוטופי הנ"ל היא – איפה הכסף? חוסר הבנה נפוץ כורך קוד פתוח עם "תוכנה חנימית" ומבלבל בין העובדה שאין צורך לשמש דמי רשיון עבור ההפצה והשימוש בתוכנות קוד פתוח, לבין העובדה הפשוטה שפיתוח תוכנה, תחזוקתה והתמיכה בה עולים כסף.

מקור הבעיה היא שהשיטה הקניינית לפיתוח תוכנה משתמשת בחוקי זכויות היוצרים והסודות המסחריים להטלת מס בחסות המדינה על הפצת תוכנות קנייניות כדרך למימון פיתוחן ואחזקתן. למעשה, בעזרת ההגבלות המלאכותיות הללו, הופכת תוכנה למוצר שהמודל הכלכלי העומד בבסיסה זהה למודל הכלכלי של פיתוח מוצרים בני קיימא אחרים.

אבל תוכנה היא מוצר מוזר מאוד. בדומה למוצרים אחרים יש לו עלות פיתוח ראשונית. בניגוד למוצרים אחרים, מרגע שקיימת תוכנה מוגמרת ניתן להפיק עותקים אין ספור ממנה בעלות אפסית. כמו כן, בניגוד למוצרים לא מוחשיים אחרים שלהם תכונה דומה, כמו שיר או סיפור, קנייה של רשיון שימוש אינה חושפת בפניך את הידע הטמון במוצר אלא מאפשרת שימוש מוגבל בו בלבד. זאת בשל הדואליות של קוד מקור מול בינארי היחודית לעולם התוכנה. ספר מכיל את כל הידע אודות הספר, אבל קובץ הרצה של תוכנה חסר את הידע בצורה שימושית הניתנת ללמידה – קוד המקור של התוכנה. גם השימושיות של תוכנה שונה ממוצרי הון אינטלקטואלי אחרים: ספר או שיר אינם "שימושיים" במובן שקשה לדבר על התאמה של ספר ללקוח מסויים או מציאת טעויות בשיר כפי שקורה לעיתים קרובות בתוכנה.

אבל האם קיימת התייחסות כלכלית אחרת לתוכנה שתפיק תוצאות טובות יותר? ובכן תשובה אחת טמונה בהקבלה לתחום אחר שגם בו הנושא המרכזי הינו שימושי אבל בלתי מוחשי – חוק ומשפט.

תוכנה כשירות

עורכי דין אינם מייצרים מוצרים. אמנם חוזה שנערך על ידי עורך דין אחד היא יצירה שניתן להחיל עליה את חוקי זכויות היוצרים, אבל המנהג המקובל איננו

עושה שימוש ביכולת הזאת. אנו רגילים לקחת חוזה שהוכן על ידי עורך-דין אחד לעורך-דין אחר, כדי שזה יתאימו לצרכינו. טיעונים של עורכי דין במסגרת התדיינות משפטית אינם קנייניים – ההיפך, המושג של "תקדים משפטי" עומד בבסיס המערכת המשפטית שלנו, ומרגע שעו"ד הגה טיעון משפטי מוצלח כל עו"ד אחר רשאי לעשות בו שימוש. למעשה, המערכת הכלכלית העומדת בבסיס מקצוע עריכת הדין בנויה על התפיסה של עריכת דין כשירות ולא כמוצר. למרות שאיננו נוהגים להפעיל הגבלות על הפצה של התוצרים של עורכי דין, עדיין אנו משלמים להם עבור הפקתם, עבור תמיכה משפטית וייצוג משפטי.

הדוגמה של עורכי הדין מאפשרת לנו להבין את מודל עסקי אחד של תוכנת קוד פתוח: תוכנה כשירות ולא כמוצר. התשלום עבור פיתוח תוכנה אינו קשור להפצתה, ממש כמו טיעון משפטי של עורך דין. פיתוח תוכנה, תמיכה בתוכנה, מתן אחריות ותחזוקה עולים כסף, ועל שירותים אלו צפוי המשתמש לשלם. העובדה שהתשלום נעשה ישירות עבור השירות הרלוונטי ולא מגולם כרשיון שימוש בתוכנה הקשור להפצתה אינו אומר שכסף לא מחליף ידיים – רק שהתשתית הכלכלית להחלפת התשלום אינה תלויה ביצירת מגבלה מלאכותית על השימוש, הלמידה וההפצה של התוכנה.

התייחסות לתוכנה כאל שירות ולא מוצר יוצרת מצב חדש ומעניין בשוק התוכנה: מאחר ואין הגבלה על הפצה ולמידה של תוכנה הרי שכל בעל ידע רלוונטי יכול לספק את השירותים השונים הדרושים למשתמשים, ללא תלות בשאלה האם הוא בעל הזכויות המקורי בתוכנה, אם לאו. כתוצאה ישירה מכך, מתפתח שוק חופשי ותוסס של אספקת שירותי תוכנה: פיתוח, אחריות, ואחזקה שיש בו תחרות אמיתית בין ספקי השירותים. זאת לעומת השוק המנוון של התוכנה הקניינית. בשוק זה התחרות קיימת עד הרגע של בחירת המוצר, ומאותו רגע הלקוח כבול לספק התוכנה שלו היכולת הבלעדית לספק לו גרסאות עתידיות, אחריות ותחזוקה.

המשמעות המעשית של האמור לעיל, היא כי פיתוח של תוכנת קוד פתוח נשלט למעשה על ידי עקרונות כלכלת השוק החופשי, כפי שניסח אותם אדם סמית בספרו "עושר האומות". הקצאת המשאבים בתוך פרויקט קוד פתוח מתרחשת על פי רצון המשתמשים – כאשר חברת "סיליקון גריפיקס" היתה זקוקה לתמיכה בריבוי מעבדים כבד בלינוקס, היא שילמה למהנדסים שידאגו לתמיכה זאת. כאשר חברת "מונטה ויסטה" היתה זקוקה לתמיכה ביצועי זמן אמת בלינוקס היא שילמה למהנדסים אחרים שידאגו לזאת. באופן דומה, חברת

"אורקל" דאגה שלינוקס תהיה מערכת ההפעלה המיטבית להרצת מאגרי מידע גדולים, וחברת "נובל" דואגת כיום להשמש לינוקס כמערכת לשולחן המשתמש הביתי ומשרדי.

היכן שיש חברות המוכנות לשלם כסף עבור מומחיות, תחזוקה, אבטחת איכות וכדומה, יש כמובן גם חברות המוכנות למכור שירותים כאלו. המפורסמת ביותר אולי היא חברת RedHat המספקת שירותי תמיכה ב"הפצת" לינוקס שהיא משווקת כמותג ארוז ומוכר, שהצליחה עד כדי הנפקה בורסאית. זה גרם לה להיות לשנויה במחלוקת. קיימות מספר חברות נוספות העושות את כספן במכירת הרגשת הבטחון שיש לתוכנה "אבא" שתומך בלקוח, כגון "נובל", "מנדרייק", ו"פרוגני" בעולם, ו"טק-מערכות", "mksoft", "ליזרד", "לינגנו" וכמובן "קודפידנס" (שכותב שורות אלו בין בעליה) בארץ (וסליחה ממי ששכחתי). חברות אלו מספקות שירותים של תמיכה והתאמה על פי דרישת הלקוחות שלהן וכך גם הן מנתבות את ההשקעה בתוכנה על פי הגדרת הלקוחות.

העקרון המצוי בפעולה כאן, הקרוי "היד הנעלמה", קובע כי המשתמש הוא המחליט מהן האפשרויות שיינתן להן משקל גדול יותר על פי צרכיו המדוייקים. הוא עושה זאת בכך שהוא משלם ישירות על זמן המהנדסים העוסקים בפיתוח. לכן הקצאת המשאבים היא המיטבית – לפי הצורך של המשתמשים כולם. בניגוד לקפיטליזם מסורתי, בכלכלת תוכנה חופשית ניתן להשפיע על תזוזת היד הנעלמה לא רק בכסף – אלא אף בהקדשת זמן לפרויקט, לכתיבת קוד, תיעודו, ובדיקתו. על-ידי כך מתאפשר גם ל"מאותגרים כלכלית" להשתתף במשחק. הנסיון שלנו מראה כי כלכלת השוק החופשי היא האלגוריתם הכלכלי היעיל ביותר להקצאת משאבים למערכות גדולות. תוכנות קוד פתוח רבות, כמו מערכת ההפעלה גנו-לינוקס, וכמו שרת האינטרנט "אפאצ'י" הם הטובות מסוגם.

אשאיר את המסקנה המתבקשת לגבי גורלם של ארגונים המסתמכים על שיטות ריכוזיות להקצאת משאבים לפיתוח פרויקטי תוכנה, והאנשים והארגונים המסתמכים על שרירותם של ארגונים כאלו, לקוראים הבקיאים בהיסטוריה הסובייטית....

תוכנה כמשלים לשירות או מוצר

הסתכלות אלטרנטיבית על ההיבט הכלכלי של תוכנת קוד פתוח היא האבחנה שלתוכנה יש ערך מכירה וערך שימוש. ערך מכירה הוא הסכום שניתן להשיג על מכירתה ואילו ערך שימוש הוא הערך שמפיק המשתמש משימושו בתוכנה. העובדה שערך המכירה הינו למעשה נגזרת של ערך השימוש, מביאה אותנו למסקנה הבאה:

תוכנת קוד פתוח מעלה את ערך השימוש במוצר ללקוח על ידי חיסולו של ערך המכירה של המוצר, בכך שהיא מאפשרת פיתוח מקבילי, בקרת איכות בשיטת peer review, וקיומו של שוק חופשי לשירותים הקשורים לתוכנה.

השלכה מעניינת אחת של העקרון המובא לעיל היא, שמרגע שמוצר קוד פתוח המשרת נישה מסויימת מגיע לנקודת בגרות מינימלית, הרי שבעוד שעריך השימוש העולה הוא של תוכנות הקוד הפתוח בנישה בלבד, ערך המכירה היורד הוא של כלל התוכנות והמוצרים בשוק המשרתים את אותה נישה, קנייניים ופתוחים באחד. שהרי, מרגע קיומה של חלופה פתוחה ש"לא עולה כסף" טובה מספיק, המחיר שיכולים יצרני מוצרי תוכנה קניינית לדרוש עבור המוצרים שלהם יורד ביחס ישר לבגרות והיקף הפתרון שניתן לבנות מתוכנת הקוד הפתוח באותה נישה.

דרך מעשית מאוד לעשות כסף מתוכנות חופשיות ופתוחות קוד איפה, היא ביכולת למכור שירות או מוצר התלוי בתוכנה חופשית לאספקה יעילה וזולה יותר שלו. כך מנוע החיפוש "גוגל" מופעל על ידי צבר, או cluster בלע"ז, מכונות הלינוקס המסחרי הגדול ביותר בעולם. כך, השרתים הנמצאים בחדרי השרתים של חברות לניתוח שווקים כלכליים מהגדולות בעולם כמו "מורגן סטנלי" מריצות לינוקס. החברות העוסקות במסחור של הנדסה גנטית משתמשות בכלים ומאגרי מידע הכתובים בשפת התכנות החופשית Perl ובכלים חופשיים כמו Blast. חברת טלפוניה האינטרנט NuFone, כמו חברות רבות אחרות בתחום, מבוססות על פלטפורמה של תוכנת הקוד הפתוח אסטרסק הרצה על לינוקס.

בשני עקרונות בסיסיים אלה, של תוכנה כשירות, ותוכנה כמאפשרת של מוצר או שירות זולים וטובים יותר, טמון המון כסף. הראשונים לדעת זאת הם בתי תוכנה קנייניים גדולים שמאז ומתמיד דאגו להשתמש במיטב של התוכנה החופשית כדי להפיק את מוצריהם ולהתבסס עליה (ומי שלא מאמין מוזמן

לבדוק את הודעת זכויות היוצרים של היישום ftp.exe בכל התקנה של חלונות (XP).

מה שאיפשר לחברות אלו לנצל את העושר של תוכנה חופשית כחלק מהתוכנה הקניינית שהם מוכרים לנו, הוא חוסר המודעות של הציבור ובעיקר עולם העסקים, לעולם הקוד הפתוח והתוכנה החופשית. אבל העולם השתנה. מאמצייהם של אנשים כמו ריצ'ארד סטולמן, ברוס פרנס, אריק ריימונד ולינוס טורבלדס העניקו לכולנו הזדמנות שווה ליהנות מהמשאב הגלובלי של ידע הטמון בתוכנה חופשית ופתוחת קוד. ליהנות. ולהרוויח.

