מעבדי-תמלילים: טיפשיים ולא יעילים

אלין קוטרל*

3 במאי 2005

תוכן עניינים

1 הטענה 1
2 מסמכים מודפסים 2
2.1 כתיבה אל מול סידור 2
2.2 החוליים של WYSIWYG 3
2.3 מבנה המסמך 3
2.4 עורכי-טקסט 3
2.5 כמה מעלות טובות ל-ASCII . . 4
2.6 תוכנת-הסדר 5
2.7 לשלב הכל ביחד 5
3 הפצה דיגיטלית 6
3.1 מסמכים פשוטים 6
3.2 מסמכים מורכבים 6
4 הסתייגות 7
5 תעמולה, תעמולה 7
6 לעיון נוסף 8

הטיותיה השונות של המילה "סידור" מתייחסות ל-typesetting (סדר, sdaʁ), שלב בהכנת טקסט להדפסה שבו הטקסט לובש צורה (טיפו)גראפית, בהתאם למבנה הלוגי שלו. המילה "סדר" (typesetter) נהגית כ-sa'daʁ, ולא כ-'sedeʁ (order).

את המאמר המקורי, באנגלית, ניתן לקרוא בפורמטים HTML, PDF ו-TEX בהתאמה בכתובות:

- http://ricardo.ecn.wfu.edu/~cottrell/wp.html
- http://www.cs.huji.ac.il/~sandman/files/posts/17/wp.pdf
- http://ricardo.ecn.wfu.edu/~cottrell/wp.tex

*מאנגלית: יהודה רונן.

1 הטענה

מעבדי-תמלילים הם כלי טיפשי ומגושם, שלא מתאים להכנת טקסט כתוב. זוהי הטענה שאצדד בה במאמר זה. במבט ראשון, היא נראית מוזרה. אם אני מתנגד למעבדי-תמלילים, מה אני מציע כחלופה: לכתוב בכתב-יד, או אולי לחזור להשתמש במכונות-כתיבה? לא. למרות שניתן למצוא צדדים לחיוב בכתיבה ביד או במכונת-כתיבה, אני מניח שמי שקורא את המאמר הזה עושה את מרבית כתיבתו בעזרת מחשב, כמוני. הטענה שלי היא שיש דרכים עדיפות על שימוש במעבד-תמלילים להכנת טקסט במחשב.

בכוונה תחילה בחרתי ניסוח בוטה לטענה שלעיל, אך עלי להבהיר את עצמי: כשאני כותב ש**מעבדי-תמלילים** הם טיפשיים, אני לא מתכוון להגיד ש**את** או **אתה** טיפשים אם אתם משתמשים במעבד-תמלילים. אני מבקר את הטכנולוגיה, כזו שיצרני-התוכנה הגדולים הקפידו לקדם, והפכו אותה לתקן **דה פקטו** בתחום. מי שלא היה במקרה במקום הנכון בזמן הנכון מן הסתם לא מודע לחלופות. את אותן חלופות לא מקדמים יצרני-תוכנה, ומסיבה טובה: כפי שנראה, הן זמינות בחינם.

אתחיל דווקא מהסוף, מהמוצר המוגמר. ישנן שתי דרכים עיקריות להפצת טקסט מוכן, המעוצב כך שיתאים להעברת מידע ורעיונות לאנשים אחרים:

1. באמצעות עותק מודפס.
2. באמצעים דיגיטליים, כגון דואר אלקטרוני, דפי אינטרנט, וכל פורמט אחר שניתן להצגה על מסך-מחשב.

כמובן שיכולה להיות חפיפה בין שתי הדרכים. לדוגמה, מסמך להדפסה יכול להשלח באופן דיגיטלי, בתקווה שמי שיקבל את הקובץ ידפיס אותו. עם זאת, אתייחס לכל אחת מהדרכים בנפרד.

2 מסמכים מודפסים

נאמר שאתם רוצים להקליד מסמך במחשב שלכם, ולהדפיס אותו כראוי במדפסת. מן הסתם לא תרצו שההדפסה תתרחש בזמן־אמת (כלומר, שמה שאתם מקלידים יודפס בו־ברגע). תרצו להקליד את המסמך ולשמור אותו דיגיטלית באמצעי־אחסון כלשהו, לאחזר את המסמך ולערוך בו שינויים במידת־הצורך, ורק כשיגיע הזמן המתאים, לשלוח אותו להדפסה. ודאי שמעבד־תמלילים — כגון Word של חברת Microsoft, שהוא הנפוץ ביותר — הוא הדרך ה"טבעית" לעשות זאת, לא? למעשה, זו רק אחת הדרכים, ואפילו לא הטובה שבהן. למה לא?

2.1 כתיבה אל מול סידור

כשמכינים בעזרת מעבד־תמלילים מסמך להדפסה יש כורך לאחד שתי משימות השונות **קונספטואלית** אחת מהשניה. אם רוצים שהזמן ינוצל בצורה היעילה ביותר, ושהתוצאה הסופית תהיה אופטימלית, צריך להפריד הפרדה **מעשית** בין שתי המשימות האלה, שהן:

1. כתיבת הטקסט עצמו. אני מתכוון לעצם בחירת המילים שמשמשות להבעת הרעיון, ולארגון המבנה הלוגי של הטקסט. זה האחרון משתנה בתלות בטבעו של המסמך, וכולל תחומים כגון חלוקה לפסקאות, סעיפים ופרקים, ההחלטה אם אמירה מסויימת תכתב בהערת־שוליים או בגוף הטקסט, הדגשה של חלקים שונים בטקסט, הפרדה בין מובאות לבין הדברים שהמחבר עצמו כותב, וכן הלאה.

2. סידור המסמך (typesetting). הכוונה לעניינים כגון בחירת משפחת־הפונטים שבה יודפס הטקסט והדרך שבה החלקים השונים במבנה הלוגי יוצגו בפועל. האם להציג כותרות של סעיפים בפונט מעובה? לישר את הכותרות לימין או למרכז? האם לישר את הטקסט דו־כיוונית? להציג הערות בתחתית העמוד או בסוף הטקסט? האם לעמד את הטקסט בעמודה אחת או בשתיים? וכן הלאה.

מחבר הטקסט צריך להתמקד אך ורק במשימה הראשונה מבין השתיים. זהו עיסוקו של המחבר. אדם סמית מנה בזמנו את היתרונות הגדולים שנובעים מחלוקה של העבודה. חלקו של המחבר ביצירת התוצר המוגמר הוא חיבור הטקסט עצמו ובניין המבנה הלוגי שלו. הסידור בפועל הוא עיסוקו של הסדר (typesetter). בעידן שלפני המחשב חלוקת־העבודה הזו בוצעה כראוי בתהליך המסורתי להכנת ספרים ומאמרים. המחבר כתב את הטקסט, וסימן את המבנה הלוגי שלו באמצעות סימונים שונים. הסדר פירש את הסימונים של המחבר, מימש אותם בעיצוב טיפוגראפי מסויים, ותרגם את מה שקיבל מהמחבר למסמך מודפס. כדי להבין עד כמה איחוד של פעולות הכתיבה והסידור הוא מגוחך, נסו לדמיין את ג'יין אוסטין תוהה באיזה פונט להשתמש בכותרות־הפרקים ב־Pride and Prejudice ("גאווה ודעה קדומה"). ג'יין אוסטין היא סופרת נהדרת, אבל היא לא סדרת.

יתכן שבנקודה הזו תחלקו על מה שנאמר. כתביה של ג'יין אוסטן יצאו לאור; לסדרים מקצועיים היה עניין בהדפסתם. לנו, לעומת זאת, לא התמזל המזל — אם נרצה לכתוב מאמר, נצטרך לעשות זאת בעצמנו (ובנוסף, נרצה שהעבודה תעשה במהירות רבה בהרבה מזו שבשיטת הסידור המסורתית). זה נכון רק חלקית. אמנם עלינו לבצע את העבודה בעצמנו, על המחשב, אבל אנחנו יכולים להעזר בו רבות. במיוחד יש לציין את תוכנת־הסידור. **תוכנה זו (או אוסף של תוכנות) תעשה בעבורנו, בחינם ובשברי־שניה, את העבודה שהסדרים האנושים בצעו עבור שייקספיר, ג'יין אוסטין, סיר וולטר סקוט וכל השאר.** כל שעלינו לעשות הוא לספק לתוכנה טקסט ובו סימונים שונים בשפה מוסכמת מראש, כמו שעשו בשיטה המסורתית.

אם כן, אני מציע שנחלק את הפקתו של טקסט מודפס במחשב לשני שלבים: בשלב הראשון מקליד המחבר את הטקסט ומחליט על המבנה הלוגי שבו הוא יסודר, כשהוא מסמן את המבנה בשפת־סימון פשוטה. את השלב הזה מבצעים בעזרת **עורך־טקסט** (אין לבלבל עם מעבד־תמלילים, את ההבדל אסביר בהמשך). בשלב השני מעבירים את הטקסט המסומן לתוכנת־סידור, שלאחר פרק זמן קצר מאוד מפיקה פלט מעומד יפה.

2.2 החוליים של WYSIWYG

שתי המלאכות שהוזכרו קודם מאוחדת במעבדי-תמלילים מודרניים מסוג WYSIWYG (ראשי-תיבות של What You See Is What You Get, כלומר: מה שאתה רואה זה מה שתקבל). **בו בזמן** שמקלידים את הטקסט הוא מוצג על-גבי המסך, מעוצב בתבנית טיפוגראפית שאמורה להיות דומה ככל הניתן לתוצר המודפס (למרות שמסיבות שונות התוצאה לא תמיד זהה). למעשה, הטקסט מסודר (typeset) בזמן ההקלדה. במבט ראשון, זה נראה מאוד נוח; אך בבדיקה מעמיקה יותר מדובר בקללה של ממש. הבעיה מורכבת משלושה אספקטים.

1. דעתו של המחבר מוסחת מהעיקר, שהוא כתיבת הטקסט עצמו, לטובת עיצוב הטיפוגראפיה, מה גם שיתכן שלמחבר אין כלל כישורים ונסיון בעיצוב טיפוגראפי ("לשחק בפונטים ובעימוד השוליים" במקום להתרכז בתוכן).

2. אלגוריתם סידור הטקסט שממומש במעבדי-תמלילים WYSIWYG מקריב את האיכות לטובת המהירות הדרושה לסידור ועדכון הטקסט בזמן אמת. התוצר הסופי נחות משמעותית מזה שתוכנת סידור אמיתית היתה מפיקה.

3. כשמשתמשים במעבד-תמלילים, קיים פיתוי מתמיד לאבד את הפוקוס מבניית המבנה הלוגי של המסמך ולאחד אותו עם העיצוב הטיפוגראפי.

שתי הנקודות הראשונות מסבירות את עצמן בבהירות. ארחיב על הנקודה השלישית, שחשיבותה משתנה לפי סוג המסמך המדובר.

2.3 מבנה המסמך

נתבונן לדוגמה בכותרות לסעיפים. כשעוסקים בארגון המבנה הלוגי של המסמך, כל מה שחשוב הוא לסמן את הכותרת באופן כלשהו ככותרת. אפשר, לדוגמה, לכתוב "\section{הכותרת של הסעיף}". האופן שבו הכותרות ממומשות בפועל, מבחינה טיפוגראפית, הוא כבר עניין אחר. כשמשתמשים במעבד-תמלילים, לעומת זאת, מה שרואים זה (כל!) מה שמקבלים: יש צורך לבחור את המראה הטיפוגראפי של הכותרת כשיוצרים אותה.

נניח שבחרתם לעצב את הכותרות בפונט מעובה, הגדול במעט משאר הטקסט. איך תשיגו את המראה הזה? יש יותר מדרך אחת לעשות את זה, אבל רוב האנשים יבחרו בדרך האינטואיטיבית עבורם, לפי כל העקרון של ה-WYSIWYG: להקליד את הכיתוב של הכותרת, לסמן אותו, ללחוץ על האייקון של העיבוי, ולבחור את הגודל המתאים מהרשימה של הגדלים האפשריים. יופי - עכשיו הכותרת מעובה ומוגדלת.

טוב, אבל מה בעצם אומר שזו כותרת? אין במסמך שום דבר שמסמן את פיסת-הטקסט הזו ככותרת. נניח שמתישהו תרצו לשנות את העיצוב של הכותרות: למרכז אותן ולמספר אותן בספרות רומיות, או כל דבר אחר. הדבר הטבעי לבקש המחשב הוא "שנה את העיצוב של כל הכותרות לעיצוב כזה וכזה". הנקודה היא שאם פעלתם בדרך שלעיל, תצטרכו לעבור על כל המסמך ולשנות ידנית כל כותרת וכותרת.

אמנם **קיימת** דרך לסמן את התפקיד המבני של אלמנט מסויים במסמך בתוכנת Word, לדוגמה. **אפשר**, אם עושים את זה בזהירות, לכתוב מסמך כך שניתן יהיה להחליף בו את העיצוב של כל הכותרות בפקודה אחת. עם זאת, רק מעטים מהמשתמשים של Word באמת עושים שימוש שוטף בתכונה הזו. זה כלל לא מפתיע: הגישה שעומדת מאחורי ה-WYSIWYG לא מעודדת התמקדות במבנה של המסמך. אפשר בקלות — מדי בקלות — "לזייף" את המבנה בעזרת בחירה מקומית לעיצוב של הטקסט. כשמקלידים את הטקסט בעורך-טקסט, לעומת זאת, הצורך בסימון מסודר של המבנה ברור מאליו.

2.4 עורכי-טקסט

טוב, הגיע הזמן להסביר מה זה עורך-טקסט, ובמה הוא שונה ממעבד-תמלילים. עורך טקסט מודרני נראה, חיצונית, די דומה למעבד-תמלילים. יש לו תפריטים נגללים ואייקונים שמשמשים לפעולות כמו שמירה, טעינה, חיפוש מחרוזת בטקסט ובדיקת-איות. מה שאין בעורך-טקסט אלו הפונקציות שעוסקות בעיצוב וסידור המסמך. הטקסט שמוקלד מופיע בצורה בהירה על המסך, אבל לא מופיעה הצורה הסופית של הפלט המודפס.

מסמכים נשמרים בפורמט של **טקסט נקי** (plain text), בדרך-כלל ב-ASCII (ראשי-תיבות של American Standard Code for Information Interchange). ASCII מורכב מ-127 תווים (ולכן הוא לפעמים נקרא קידוד של "7-ביט": שבע ספרות בינאריות מספיקות כדי לייצג כל אחת מהאותיות בו, שהרי $2^7 = 128$). יש בו את הספרות 0–9, את האלף-בית הלטיני באותיות ראשיות וקטנות, סימני-הפיסוק הרגילים וכמה סימנים מיוחדים. ASCII הוא המכנה המשותף המינימאלי לתקשורת טקסטואלית בין מחשבים. לכן, אפשר לקרוא מסר ב-ASCII מכל מחשב שהוא בעולם. כששולחים מסר כזה, אפשר להיות בטוחים שהצד שמקבל אותו יוכל לקרוא אותו.

במעבדי-תמלילים, לעומת זאת, כששומרים קובץ נשמרים ביחד עם הטקסט גם "תווי-בקרה" מחוץ לטווח התווים של ASCII. התווים האלה מסמנים את העיצוב שנבחר (דוגמת אותיות מעובות או נוטות) וכל-מני עניינים אחרים, פנימיים, של מעבד-התמלילים. התווים האלו הם לא סטנדרט אוניברסלי. כדי לקרוא את המסמך יש צורך בעותק של מעבד-התמלילים שבו נוצר המסמך, או, לחלופין, בתוכנת-המרה מתאימה. אם תפתחו קובץ שהפיק מעבד-תמלילים בעורך-טקסט תראו (בנוסף לטקסט עצמו, או לחלקים ממנו) הרבה "סימנים מצחיקים": אלו תווי-הבקרה הבינאריים.

כיוון שעורך-טקסט לא מוסיף תווי-בקרה בינאריים, במקרה שרוצים לפרט את העיצוב של הטקסט יש צורך להשתמש ב**שיטת-סימון מוסכמת**. אם נרצה לכתוב טקסט באותיות נוטות, לדוגמה, נצטרך להוסיף סימון, ב-ASCII, שיפורש על-ידי **תוכנת הסדר** באופן מתאים. בסדר LaTeX, עליו נרחיב בהמשך, נכתוב לדוגמה "\textit{כיתוב באותיות נוטות}". למעשה, אם עורך-הטקסט תומך ב-LaTeX, אפילו לא צריך להקליד את הפקודה ידנית. אפשר להשתמש באיזה קיצור-מקשים, בבחירה של הפקודה מתפריט, או בלחיצה על אייקון, והפקודה תופיע אוטומטית במקום המבוקש. מבחינת המכניזם של הממשק, צורת העבודה ליצירת קובץ ASCII שמיועד להזנה ל-LaTeX לא שונה בהרבה מזו שמשמשת במעבדי-תמלילים מודרניים.

2.5 כמה מעלות טובות ל-ASCII

לגישה של כתיבת קובץ ASCII[1] עם סימונים מתאימים באמצעות עורך-טקסט, והזנת הקובץ לתוכנת-סדר לקבלת הפלט הסופי יש כמה יתרונות נוספים:

1. פורטביליות: כמו שהוזכר קודם, **כל אחד** יכול, בעזרת כל מחשב, לקרוא את קובץ הטקסט, אפילו כמות-שהוא, לפני שעובד בתוכנת-הסדר. לעומת זאת, הקובץ שהפיק מעבד תמלילים, נאמר XYZ 9.0, יכול להיות בלתי-קריא לחלוטין למי שאין ברשותו עותק של אותו מעבד-התמלילים באותה גרסה בדיוק (אלא אם כן למי שמנסה לפתוח את הקובץ יש את הידע הטכני המתאים כדי לחלץ את התוכן מתוך כל ה"זבל" הבינארי). העדר תאימות בין הגרסאות השונות של מעבדי-תמלילים יכול לפגוע גם במי שכתב את המסמך: יתכן שתהיה לכם בעיה לפתוח קובץ שהפיק XYZ 8.0 בעזרת XYZ 9.0 ולהפך, אבל לעולם לא תהיה בעיה לפתוח קבצי ASCII ישנים.

2. קומפקטיות: התוכן שבקובץ ה-ASCII מייצג **רעיונות**, בלי כל-מני עניינים פנימיים של מעבד-התמלילים. במיוחד במסמכים קצרים, קובץ ה-ASCII יכול להיות קטן בהרבה מקובץ מעבד-התמלילים המקביל לו מבחינת התוכן הרלוונטי, אפילו עד כדי עשירית הגודל.

3. הגנה מפני פגמים טכניים: הגישה של "עורך-טקסט בשילוב עם תוכנת-סדר" מבטיחה ברמה העקרונית העדר מוחלט של תקלות טכניות במסמכים (אלא אם כן הדיסק הקשיח נפגם, או שאסון דומה קורה לנתונים). קובץ המקור תמיד קיים, גם אם תוכנת-הסדר לא מצליחה משום-מה לעבד אותו. אם אתם משתמשים תדיר במעבדי-תמלילים **ולא** חוויתם בעיות של קבצים פגומים, אתם ברי-מזל!

[1] מה שהמחבר, אלין קוטרל, כותב על ASCII מתאים לשפות שמערכת-הכתב שלהן מבוססת על זו הלטינית. לשפות אחרות, כמו עברית או יפאנית לדוגמה, הקידוד הטבעי היום הוא Unicode. עם זאת, כל מה שנכתב כאן על ASCII תקף גם ל-Unicode: הפורטביליות, הקומפקטיות וההתמודדות עם תקלות. י.ר.

2.6 תוכנת-הסדר

עכשיו אגש לפרט על החלק של תוכנת-הסדר בשיטה שאני מצדד בה. לא אכנס לפרטים טכניים, אבל אנסה לתת מושג לגבי מה שאני חושב עליו.

תוכנת-הסדר הבסיסית שאני מתכוון אליה נקראת TEX, והיא נכתבה בידי דונאלד קנות (Donald Knuth) מאוניברסיטת סטנפורד. TEX זמינה בחינם (היא ניתנת להורדה מאתרים רבים באינטרנט), ותומכת כמעט בכל פלטפורמת-מחשב. (אם תרצו, אפשר לרכוש CDROM עם חבילת TEX מלאה בעבור מחיר צנוע). קנות החל לעבוד על TEX ב-1977; בשנת 1990 הכריז על כך שהוא לא מעוניין להמשיך ולפתח את התוכנה — לא בגלל העדר עניין, אלא בגלל שהתוכנה הגיעה לרמת שלמות מספקת. היא נקיה מבאגים, עד כמה שתוכנה יכולה להיות כזו, ועושה את עבודת הסידור היטב, בין אם מדובר בטקסט פשוט או בכזה שמשלב מתמטיקה גבוהה.

הזכרתי קודם את LATEX. אם TEX היא תוכנת-הסדר הבסיסית, המנוע, הרי ש-LATEX היא אוסף של מאקרואים שנכתבו עבור TEX, בתחילה על-ידי לזלי למפורט (Laslie Lamport) בשנות השמונים של המאה הקודמת, וכיום על-ידי קבוצה של מומחים ברחבי-העולם. אותם מאקרואים מקלים מאוד על החיים של של המשתמש הממוצע. LATEX נמצאת עדיין בפיתוח פעיל, ויכולות חדשות וחבילות חדשות נוספות על בסיס תוכנת-הסדר. יש ל-TEX תוספים רבים בפיתוח, כגון מערכת שמאפשרת הפקה של קובץ PDF (Portable Document Format של Adobe) ישירות מקובץ המקור ב-ASCII. (כשאני כותב על תוכנה שהיא "בפיתוח", אני מתכוון לכך שממשיכים לעבוד עליה ולשפר אותה, ורק לכך. התוכנות האלו כבר בשלב שבו הן יציבות ומתפקדות כראוי).

כמו שכתבתי קודם את המבנה הרצוי של המסמך מזינים ל-LATEX על-ידי קבוצה של סימונים. יש הרבה ספרים ואתרים ברשת שמפרטים את הסימונים האלה, ולא אכתוב עליהם כאן. הסימונים הנפוצים הם פשוטים וקלים לזכירה, מה גם שאפשר להשתמש בעורך-טקסט ידידותי ל-LATEX, שמאפשר גישה מקוצרת אליהם.

אחת התכונות האטרקטיביות ב-LATEX היא היכולת לשנות באופן קיצוני **ועקבי** את העיצוב של המסמך בעזרת כמה פקודות. המראה הכללי נשלט על-ידי

1. "מחלקת המסמך" שבחרתם (דוגמת דו"ח, מכתב, מאמר או ספר).
2. "חבילות" שונות וקבצי-סגנון שאפשר לטעון למסמך.

אפשר, לדוגמה, להחליף את משפחת-הפונטים שבשימוש (באופן עקבי: בטקסט עצמו, בכותרות, בהערות ובכל שאר המקומות בהם מופיע כיתוב) ו/או את הגדלים של הפונטים בעזרת שינוי פרמטר או שניים ב"רישא" של הקובץ. בדומה, ניתן לסדר את הכתוב בשתי עמודות, או לסובב אותו כך שהשורות יפרשו לרוחב הדף. אפשר לעשות את כל אלה גם בעזרת מעבד-תמלילים, אבל בדרך-כלל זה פחות נוח, והסבירות ליצירת חוסר-עקביות במראה המסמך גדולה יותר.

רמת-המורכבות של השימוש ב-LATEX משתנה לפי בחירה. אפשר לבחור בגישה "אוטומטית": רק לבחור את המחלקה של המסמך, ולתת למאקרואים הדיפולטיביים לעשות את כל שאר העבודה. הגישה הזו נותנת תוצאות טובות בדרך-כלל, באיכות טובה בהרבה מזו שתופק על-ידי מעבד-תמלילים. (באופן טבעי, דברים כמו מספור הפרקים, הסעיפים, ההערות, התאמת מראי-מקום, וכדומה, נעשים אוטומטית). אפשר גם לבחור בגישה אחרת, ולטעון חבילות שונות (או אפילו לכתוב חבילות לבד) כדי לשלוט על אספקטים שונים של העיצוב הטיפוגראפי. אם אתם נוטים לכיוון הזה, של השימוש המורכב יותר, ניתן ליצור פלט יפה ובעל מראה אישי.

2.7 לשלב הכל ביחד

איך הכל פועל ביחד? אם מותקן אצלכם עותק של TEX כראוי, התהליך הוא כזה: מקלידים את הטקסט בעורך-טקסט ידידותי ל-TEX. אפשר להקליד את הסימונים ישירות, או להשתמש בעורך ולהוסיף אותם על-ידי תפריטים או כפתורים. כשמגיעים לשלב שבו רוצים לראות את התוצאה הסופית, המסודרת, בוחרים את הפקודה המתאימה מהתפריט, או לוחצים על כפתור, והסדר מופעל. פקודה אחרת מהתפריט, או כפתור אחר, יגרמו לתוכנה לצפיה מקדימה להפתח ולהציג את הטקסט כפי שיודפס. בדרך-כלל זה "WYSIWYG" אמיתי — התוצאה שתוצג קרובה מאוד לפלט המודפס. אפשר לעשות זום החוצה ופנימה,

לדפדף בין הדפים השונים, וכן הלאה. אפשר לשלוח את הפלט להדפסה, או לחזור לעריכה.

נניח שבשלב מסויים, אחרי שערכתם את המסמך, אתם רוצים לראות תצוגה מקדימה מעודכנת. לחצו שוב על הכפתור של הסדר. הפעם לא צריך להפעיל מחדש את התוכנה של התצוגה המקדימה: אם השארתם אותה פועלת ברקע, הגרסה המסודרת המעודכנת תופיע אוטומטית. כשסיימתם את העריכה, אפשר למחוק את הגרסה המסודרת של הקובץ כדי לחסוך במקום בדיסק. כל מה שצריך לשמור הוא קובץ ה-ASCII; את הגרסה המסודרת ניתן ליצור מחדש בקלות בעת הצורך.

3 הפצה דיגיטלית

הסעיף הקודם התמקד בהפקת פלט מודפס נאה. ישנם כמה שיקולים נוספים שנוגעים להכנה של מסמכים שמיועדים להעברה דיגיטלית (דואל, דפי-רשת וכדומה).

תחילה נעסוק בדואל. בדרך-כלל כשרוצים לשלוח הודעה קצרה, אד הוק, מקלידים אותה ישירות ללקוח הדואל, בין אם מדובר בתוכנות "מסורתיות" כמו Pine, ובין אם מדובר בתוכנות עם GUI (Graphic User Interface — ממשק-משתמש גראפי), כמו Netscape או Eudora. במקרה כזה, של הודעות קצרות, ההודעה נשלחת על-ידי תוכנת-הלקוח ב-ASCII או ב-HTML (HyperText Mark-up Language — שפת-הסימון שמשמשת בדפי-אינטרנט, ובנויה על-בסיס יצוג ב-ASCII). אבל מה קורה כשרוצים לשלוח טקסט ארוך יותר, שהוכן בתוכנה נפרדת מלקוח הדואל?

הפתרון הנפוץ ביותר במקרה כזה הוא לצרף להודעה קובץ של מעבד-תמלילים. איך השיטה האלטרנטיבית שאני מציע משתלבת כאן?

צריך להפריד בין שני מקרים שונים: הפתרון יהיה שונה לטקסט פשוט וקצר (מזכר, מכתב, קביעת-פגישה וכו') ולטקסט מורכב יותר (מאמר אקדמי, שאולי יתכן בו שימוש רב ברישום מתמטי, דו"ח עם תרשימים, ספר, וכו'). בגישה של "ASCII עם סדר" הפתרון שונה בכל אחד מהמקרים.

3.1 מסמכים פשוטים

במקרה של מסמכים פשוטים, נשאלת השאלה האם צריך בכלל להשתמש בהם בסדר. האם לא יהיה יעיל יותר וחסכוני יותר לשלוח את הטקסט ב-ASCII פשוט, עם העיצוב הבסיסי ש-ASCII מאפשר? כך גם נחסך רוחב-פס (כזכור, קבצים של מעבדי-תמלילים יכולים להיות גדולים **בהרבה** מקבצי-ASCII שמכילים את אותו התוכן) וגם מובטח שלא תמנע מאף-אחד האפשרות לקרוא את הטקסט, רק בגלל שהוא לא מריץ את גרסה 9.0 של XYZ. כמו שאפשר לצרף לדואל קבצים של מעבד-תמלילים, כך ניתן לצרף גם קבצי ASCII, שנוצרו בעורך-טקסט. אפשר גם להדביק את הטקסט הכתוב לתוך התוכן של הדואל עצמו (אחרי הכל, מדובר בטקסט נטול-עיצוב). לקבצי-TEX פשוטים אפשר להתייחס באותו האופן, שהרי הם כתובים ב-ASCII, ובמסמכים פשוטים אין בהם יותר מדי סימנים מיוחדים של TEX, וגם אלו המעטים די ברורים ומסבירים את עצמם.

3.2 מסמכים מורכבים

מסמכים ארוכים יותר ומסובכים יותר קל יותר לקרוא כשהם מסודרים בסדר. קשה לכתוב מתמטיקה ב-ASCII, ובטח שקשה להעביר גם תרשימים מורכבים ותמונות. אז איך TEX עוזר לנו כאן? טענתי קודם שמעבדי-תמלילים הם בעייתיים לשימוש, בגלל שלא מובטח שלנמען יש בדיוק את אותו מעבד-התמלילים באותה הגרסה. זה לא נכון גם ל-TEX? הרי לכמה מהנמענים שלך מותקן TEX במחשב? זו שאלה הגיונית, שהתשובה לה פשוטה: הנמען לא צריך עותק של TEX על-מנת לקרוא את הגרסה המסודרת. קיימות מספר אפשרויות להפקת גרסה מסודרת שהנמען יוכל לקרוא:

1. להמיר את קובץ ה-TEX המקורי ל-HTML. יש תוכנות-המרה טובות למטרה הזו. (למעשה, יש דמיון משפחתי רב בין HTML ו-TEX — בשתיהן אופן הסימון של הטקסט הגיוני, כך שהמרה בין השתיים אפשרית ברמה גבוהה של דיוק)[2].

[2] הקוד הבינארי שמשמש מעבדי-תמלילים שונה באופן עקרוני, כך שהמרה הדדית בין TEX ופורמטים של מעבדי-תמלילים אינה פשוטה. בנוסף, מנוע הסידור של TEX עולה על זה שמשמש מעבדי-

כך, הנמען יוכל לקרוא את הטקסט בעזרת web browser.

2. אם לנמען יש מדפסת postscript, כמו שיש בהרבה מוסדות אקדמיים ועסקים, אפשר לשלוח גרסה מסודרת של המסמך בקובץ postscript, והנמען ידפיס אותה במדפסת או יציג אותה על המסך בעזרת תוכנת "ghostview" (זמינה להורדה בחינם מהאינטרנט).

3. אם לנמען יש עותק מותקן של קורא לקבצי Adobe PDF, כמו ה־Adobe Acrobat Reader, שזמין גם הוא להורדה בחינם, ניתן לשלוח קובץ PDF של המסמך המסודר.

מתוך העיסוק בדרכים להעברת טקסט בדואל, נגענו גם בהכנת טקסט לרשת. אפשר לכתוב HTML ידנית, או בעזרת עורך גראפי, כמו זה שמצורף ל־Netscape Communicator למשל. אפשר גם להפיק קבצי HTML בעזרת Word, אבל ה־HTML שמופק מסורבל וכמעט שלא ניתן לעריכה בעזרת תוכנות אחרות. אם משתמשים ב־TEX, קל להמיר מסמכים ל־HTML נקי שתואם לתקנים.

4 הסתייגות

במאמר ניסיתי לטעון לטובת השיטה של "ASCII בשילוב עם סדר" כאלטרנטיבה למעבדי־תמלילים. עם זאת, אני מודה שקיימים סוגים **מסויימים** של מסמכים שעבורם מעבד־תמלילים מסוג WYSIWYG הוא הבחירה הטבעית. כוונתי היא למסמכים קצרים, שבהם העיצוב של הדף הוא הדבר החשוב: פליירים, פוסטרים, הזמנות למסיבה ודברים דומים. אפשר להפיק דברים כאלה ב־TEX, אבל זה לא יהיה יעיל. מחלקות־המסמך הסטנדרטיות של LATEX (דו"ח, מאמר וכו') לא מתאימות לשימושים כאלו. בנוסף, למרות ש־LATEX הוא כלי מצויין לטיפול אוטומטי בכתיבה של טקסט פורמלי, עם פונטים "שמרניים", הוא לא מתאים לערבוב של פונטים רעשניים שמתאימים לצרכים האלה. המבנה הלוגי של המסמך בכלל לא משנה כאן: כל מה שצריך הוא לעצב את הדף ישירות. אם חשוב לך לדעת האם, לדוגמה, כשנשנה את הפונט של השורה **הזו** לגודל 36, זה ידחוף את השורה האחרונה מחוץ לדף, הרי ש־WYSIWYG מתאים לך.

אם רוב עיבוד־התמלילים שלך הוא כזה, מן הסתם כבר הפסקת לקרוא עד עכשיו. לעומת זאת, אם רוב הטקסטים שאתה כותב או עורך הם מסמכים פורמליים, ההסתייגות שלעיל בכלל לא רלוונטית.

5 תעמולה, תעמולה

ודאי שמתם לב לזה שאני מנסה לטעון בעד צד מסויים. לא מדובר רק בוויכוח אקדמי בנוגע לדרכים שונות להכנת־טקסט. יצרני־התוכנה הגדולים נמצאים כולם בצד אחד של המאזניים. אם לדבר במפורש, המצב כיום הוא כזה שבו Word של מיקרוסופט נוטה להיות **ה**תקן להכנת מסמכים בעזרת מחשב עבור רוב האנשים. אבל, למרות זאת, אין ל־Word, כתקן, אין מה להציע מלבד היותו "תקן" (או לפחות אחד שמתיימר להיות כזה).

זה מזכיר קצת את הסיפור של QWERTY. תהיתם פעם למה סידור־המקשים הסטנדרטי במכונות־כתיבה (וכתוצאה מכך גם במחשבים) הוא בסדר QWERTYUIOP בשורה העליונה? זה לא היה הסידור המקורי של המקשים במכונות־כתיבה. הוא תוכנן במטרה **להאיט את ההקלדה**. הבעיה היתה שמכונות־הכתיבה הראשונות לא הצליחו להדביק את מהירות־ההקלדה. קלדן מהיר היה יכול לתקוע את המכונה אם היה מקליד על מקש מסויים לפני שהמנוף שמחובר לו חזר למקום. המקשים בסידור QWERTY סודרו כך שההקלדה לא תוכל להיות כל־כך מהירה. ברור שמדובר בסידור־מקשים מטופש עבור מקלדות אלקטרוניות, אבל כבר מאוחר מדי לשנות את זה: QWERTY הוא סטנדרט, וכל הנסיונות לשנות את המקלדות לסידור הגיוני יותר נכשלו אל־מול המציאות.

באופן דומה, אני טוען של־Word אין **זכות** להיות סטנדרט להכנת־מסמכים, בגלל שהוא בבירור פחות יעיל (לרוב השימושים) מאשר אלטרנטיבות אחרות שיש בנמצא. אני מקווה שעדיין לא מאוחר מדי במקרה הזה, שעדיין אפשר להגיד "לא ל־Word". למעשה, במובן מסויים Word גרוע מ־QWERTY: הוא לא תקן, סטנדרט, אמיתי. ה"תקן" של חברת

תמלילים, ולכן, עקרונית, המרה של מסמך TEX ל־Word, נאמר, אינה אפשרית בלי לאבד חלק מהאינפורמציה של המסמך המקורי.

מיקרוסופט ליצוג בינארי של מסמכים הוא דבר שמשתנה לפי הגחמות שלה. עד שלא תהיה לה תחרות מסחרית צמודה, למיקרוסופט לא יהיה עניין מיוחד בביסוס תקן ארוך-טווח ליצוג בינארי של מסמכים. אדרבא, יש לה עניין לכפות על המשתמשים "לשדרג" את Word אחת לכמה זמן: אוי, Word N.0 לא קורא את המסמך שהקולגה שלח לך? עבור להשתמש בגרסה N+1, אפילו עם אין בגרסה N+1 תכונות שלא קיימות בגרסה N, לפחות לא כאלו שיש להן ערך אמיתי בשבילך[3]!

6 לעיון נוסף

אם הגעתם עד לכאן, יכול להיות שתתעניינו בעורכי-טקסט טובים, מערכת-הסידור "TEX", וכיו"ב.

המקום הטוב ביותר להתחיל בו לחיפוש מידע על TEX הוא האתר של TUG (ראשי-תיבות של TEX Users Group), בכתובת http://www.tug.org. באתר הזה יש את כל הקישורים שצריך בנושא, כשבין החשובים בהם נמצא ה-Comprehensive TEX Archive Network (ר"ת CTAN), משם אפשר להוריד חבילה מלאה של TEX לכל פלטפורמת-מחשב כמעט. בחבילות כאלו נמצאים הסדר עצמו, אוסף גדול של מקרואים, תוכנה לתצוגה מקדימה, ותוכנה להפקת קבצים להדפסה.

חבילות-TEX (לפחות אלו שמופצות בחינם) לא מכילות עורך-טקסט. במידה ואין לך עורך-טקסט מסויים שאתה כבר משתמש בו, יהיה צורך למצוא אחד. יש הרבה אפשרויות, אבל העורך המועדף עלי לעבודה עם TEX הוא Emacs (http://www.gnu.org/software/emacs/). בצירוף החבילה "AUC TEX" (http://sunsite.auc.dk/auctex/), שהופכת את Emacs לידידותי ל-TEX: שגיאות-תחביר של TEX מודגשות תוך כדי כתיבה, ופקודות רבות שקשורות ל-TEX מופיעות בתפריטים נוחים לשימוש.

תצלום-מסך של עריכת קובץ TEX בעזרת Emacs: http://ricardo.ecn.wfu.edu/~cottrell/emacs_screen.png.

[3] כמי שהשתמש מספר שנים ב-Word לפני שעבר ל-TEX, ויש לו עניין רב בסידור-טקסט, לדעתי לא נוספו לגרסת החלונות של Word תכונות משמעותיות החל מגרסה 2.0 של תחילת שנות התשעים.

הערות המתרגם

כמה דברים שלא הוזכרו במאמר, ולדעתי יעניינו את הקורא, במיוחד בכל הקשור לתמיכה בעברית:

- ל-LATEX יש תמיכה מצויינת בעברית (המסמך הזה הוכן באמצעות LATEX) בעזרת חבילות-הרחבה:
  - פרוייקט IvriTEX (http://ivritex.sourceforge.net) מספק את הבסיס לתמיכה העברית.
  - הפונטים הבסיסיים שבאים עם IvriTEX מוגבלים ופשוט נראים רע. יש אפשרות להשתמש בפונטים מסוג TrueType ב-LATEX, מה שמרחיב מאוד את אפשרויות הבחירה בכל הנוגע לפונטים, לפי ההוראות שנמצאות בדף: http://cs.haifa.ac.il/~dekelts/ttf/.
  - הקידוד הבסיסי שמשמש את IvriTEX הוא cp1255 הישן והפרימיטיבי. עדיף לעבוד בקידוד Unicode, שמאפשר פורטביליות גדולה יותר של התוכן בין תוכנות שונות, ושילוב נוח של כמה שפות באותו מסמך. כדי להשתמש ביוניקוד ב-LATEX יש צורך להתקין את החבילה הזו: http://www.unruh.de/DniQ/latex/unicode.
- עורכים נוספים:
  - עורך מצויין שלא הוזכר בטקסט עצמו הוא Vim (http://www.vim.org). ל-Vim תמיכה טובה ונוחה בעברית (http://iglu.org.il/faq/cache/101.html), וב-LATEX. את האינטגרציה של Vim ו-LATEX אפשר לשפר עוד בעזרת החבילה Vim-LATEX (http://vim_latex.sourceforge.net). Vim זמין לכל פלטפורמות-המחשב הנפוצות.
  - עורך יעודי ל-LATEX הוא Kile (http://kile.sourceforge.net). הוא בנוי בסביבת KDE, מה שמאפשר לו לרוץ באופן טבעי בלינוקס, BSD, מק ופלטפורמות נוספות.

– עריכה של LaTeX לא חייבת להתבצע ישירות בעזרת עריכה ידנית של קבצי־טקסט. LyX (http://www.lyx.org) הוא "front–end" נוח ל־LaTeX, עם תמיכה טובה בעברית (http://cs.haifa.ac.il/~dekelts/lyx).