

תיאור הפרוייקט

בפרוייקט הסיכום תיבנו תוכנת דואר, המאפשרת משלוח הודעות דואר (טקסט) בין המשתמשים השונים במערכת, דרך ספריית דואר משותפת.

לכל משתמש ישנה ספריית דואר. בכל ספריית דואר ישנם שלושה קבצים בדיוק:

- .inbox
- .inbox_index
- .outbox

הקובץ inbox (דואר נכנס) מכיל את תוכן כל ההודעות שהתקבלו על ידי המשתמש. בקובץ מופיעות שורות טקסט, והסימון לסוף הודעה הוא שורה בה מופיעה המחרוזת END_OF_MSG (באותיות גדולות, ללא כל תווים אחרים). לדוגמה:

```
This is the first message.  
END_OF_MSG  
This is the second message.  
This is the second line of the second message.  
END_OF_MSG  
This is the third message.  
END_OF_MSG
```

הקובץ inbox_index (אינדקס דואר נכנס) מכיל את המידע לגבי כל ההודעות שהתקבלו על ידי המשתמש, באותו הסדר בו הן מופיעות בקובץ ה-inbox. לכל הודעה מופיע השולח, מועד המשלוח וסטטוס (בדיוק 3 שורות). לדוגמה:

```
Alon Yavir  
Sun Jul 29 12:16:20 IDT 2001  
Unread  
Rachely Greenbaum  
Monday  
Read  
Yaniv Eliyahu  
Now  
Deleted
```

השולח - שמו המלא של שולח ההודעה.

מועד המשלוח - טקסט חופשי המתאר את המועד בו נשלחה ההודעה.

סטטוס - אחד מהבאים - Unread (ההודעה טרם נקראה), Read (ההודעה נקראה), Deleted (ההודעה נמחקה). יש לשמור על הכתיב המדויק של מילות הסטטוס, מאחר והן משמשות להשוואה. הודעה שמסומנת Deleted לא תופיע בכל תצוגה למשתמש.

הקובץ outbox (דואר יוצא) מכיל את תוכן כל ההודעות אשר נשלחו על ידי המשתמש, ועדיין לא הועברו לתיבות הדואר הנכנס המתאימות. עבור כל הודעה מופיע שם מקבל ההודעה, מועד המשלוח, מספר שורות הטקסט ושורות הטקסט עצמן. לדוגמה:

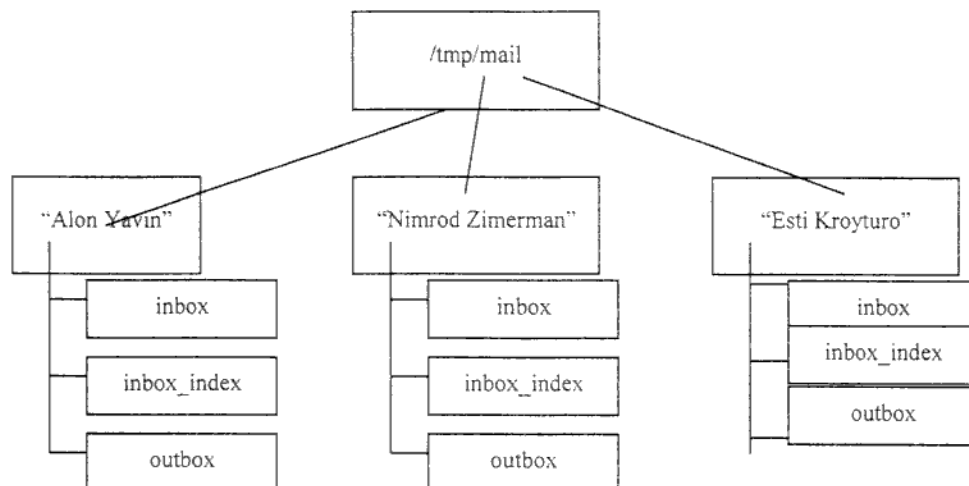
```
Sivan Solks
```

```

Two days from today
2
Line 1
Line 2
Hada Philus
01/01/2001
1
Just one line of text

```

להלן מבנה עץ ספריות דואר לדוגמה:



לכל משתמש שם כספריית הדואר שלו (לדוגמה, "Alon Yavin"). לכל משתמש יש גם שם Unix (לדוגמה, alony). החיבור בין השמות מבוצע על ידי שדה השם בקובץ הסיסמאות. כלומר, למשתמש Alon Yavin ייתכן וקוראים alony, ולכן שדה השם (החמישי) ברשומה של אלון בקובץ הסיסמאות יהיה Alon Yavin.

הערות מימוש

מאחר והפרוייקט מורכב ממספר תוכניות קטנות, מומלץ מאוד להשתמש בקובץ פונקציות וקבועים גלובליים עבור Bourne Shell. הקובץ יכיל את שם ספריית הדואר, פונקציות למציאת שם המשתמש הנוכחי, וכדי קובץ ה-Python יכיל הגדרות מקבילות בתחילת הקובץ.

יש מקרים בהם התוכניות צריכות לדעת מי המשתמש הנוכחי. **חייבים** לשאוב את השם הזה מסביבת הריצה – אסור שהוא יהיה חלק מקובץ התוכנית עצמו.

כדי שהאינפורמציה המקשרת בין שמכם לבין שם המשתמש בו אתם עובדים, עליכם להריץ את הפקודה "passwd -f" ולעדכן את שמכם.

בנוסף, אסור ששמות ספריות יהוו חלק מהקוד – התוכנית צריכה להיות בלתי תלויה בסביבה בה היא רצה, פרט לספריית הדואר, ולקבצי מערכת בעלי מיקום סטנדרטי.

דוגמאות ההרצה – באות כדוגמאות בלבד. כל עוד לא מצויין אחרת בהגדרת כל תרגיל, ניתן לשנות את הפלט כרצונכם.

הכנת הסביבה

כדי שיהיו מספר משתמשים במערכת, כך שתוכלו לשלוח מייל ממשתמש אחד למשתמש אחר נבצע את הפעולות הבאות:

1. התחבר למערכת בתור root.

a. Useradd ishai

b.

תיעוד

לא, זה לא מה שאתם חושבים. אנחנו לא מחפשים תיעוד פנימי או חיצוני נרחב מדי. שורה פה ושם במקומות שדורשים הסבר בהחלט יספיקו (את העבודה הקשה בהקשר הזה נשאיר לכם ליחידות, או בקורסים שעוסקים בהנדסת תוכנה).

מה שאנחנו כן מבקשים הוא קובץ המפרט את כל הדברים הלא-סטנדרטיים שעשיתם. בפרט, אם כתבתם איזשהו בונוס (מידע על הבונוסים תקבלו בהמשך הפרוייקט) – ציינו זאת. שם הקובץ המתאר את הדברים הללו צריך להיות README (באותיות גדולות, כדי שיהיה קל לזהותו).

אחרי הכל, אם לא נדע מה עשיתם, לא נוכל להעניק את הבונוסים...

דרך צלחה

בהצלחה!

יצירת ספריית משתמש

בשלב הראשון, עליכם לכתוב תוכנית היוצרת ספרייה עבור משתמש.

ניתן להפעיל את התוכנית בשתי דרכים:

1. תקבל שם משתמש כארגומנט יחיד, ותיצור עבורו תת-ספריית דואר בתוך ספריית הדואר.

2. תופעל ללא ארגומנטים. על התוכנית לייצור את תת-ספריית דואר בתוך ספריית הדואר עבור המשתמש שמריץ את הפקודה.

עליכם ליצור את כל הקבצים הנחוצים עם ההרשאות המתאימות.

יש לטפל בכל מקרי הקצה.

שימו לב ששם משתמש יכול להכיל רווחים.

כדי להקל עליכם נכתוב את הפרויקט בשני שלבים:

1. מימוש הפרויקט בהנחה ששם משתמש אינו מכיל רווחים.

2. לאחר שתסיימו יש להעתיק את קבצי הפרויקט למחיצה אחרת ולעדכן את הפרויקט כך שיתמוך בשם משתמש שמכיל רווחים.

את התוכנית יש לכתוב ב- Bourne Shell. שמה יהיה `create_users.sh`.



```
./create_user.sh "Alon Yavin"
```

```
./create_user.sh
```

דוגמת הרצה

הצגת רשימת המשתמשים

כעת, לאחר שניתן ליצור ספריות משתמש, יש לכתוב תוכנית המציגה אותן.

התוכנית לא מקבלת כל ארגומנטים, ומדפיסה למסך עבור כל משתמש את גודל ספרייתו (סך גודל הקבצים בספרייה, בבתים), ואת שמו. הרשימה המודפסת תהיה ממויינת לפי גודל הספרייה.

מומלץ לא להשתמש בקבצים זמניים.

את התוכנית יש לכתוב ב- Bourne Shell (ולא ב-awk!). שמה יהיה `list_users.sh`.



```
./list_users.sh
```

```
546 Alon Yavin
```

```
7624 Rachely Greenbaum
```

```
18233 Yaniv Eliyahu
```

דוגמת הרצה

העברת הודעה לתיבת הדואר היוצא

כדי שתוכלו למלא את ספריות המשתמש, נתחיל בכתובת הודעות לתיבת הדואר היוצא (משם ההודעות יועברו, מאוחר יותר, לתיבת היעד).

התוכנית מקבלת שני ארגומנטים – שם משתמש, ושם קובץ.

התוכנית מעדכנת את תיבת הדואר היוצא של המשתמש המריץ את הפקודה.

העדכון מתבצע לפי הפורמט שתואר למעלה: יש להעתיק את תוכן הקובץ שהתקבל בשורת הפקודה, לקבוע כיעד ההודעה את שם המשתמש שהתקבל בשורת הפקודה, ולהוסיף את התאריך כעת.

גם את התוכנית הזו תכתבו ב-Bourne Shell. שמה יהיה `send_message.sh`.



```
./send_message.sh "Alon Yavin" message.txt
```

דוגמת הרצה

העברת הודעות מתיבת הדואר היוצא לתיבת הדואר הנכנס

לאחר שדואר נכתב ומוצב בתיבת הדואר היוצא, יש להעביר אותו לתיבת הדואר הנכנס של המקבל.

כתבו תוכנית המקבלת שם קובץ תיבת דואר יוצא, ומעתיקה את הדואר היוצא מהקובץ לקבצי `inbox` ו-`inbox_index` של מקבלי ההודעות. שימו לב שהתוכנית מעתיקה את הדואר בלבד, היא לא מוחקת דבר מקובץ הדואר היוצא.

את התוכנית יש לכתוב ב-Awk. עליכם להשתמש ביכולות המיוחדות של Awk (ביצוע קוד עבור שורות המתאימות לתנאים מסויימים), ולא לכתוב קוד פרוצדורלי במסווה. שמה של התוכנית יהיה `transport_agent.awk`, והיא גם תקבל **שני משתנים**:

`mail_dir` - ספריית הבית של כל תיבות הדואר.

`me` - שם המשתמש השולח את ההודעות.

(שימו לב לדוגמת הריצה).



```
awk -f ./transport_agent.awk mail_dir=/mail me="Alon Yavin" "/mail/Alon Yavin/outbox"
```

דוגמת הרצה

כעת כיתבו מעטפת לתוכנית זו, אשר תקרא לה עם הפרמטרים המתאימים, ובסוף הפעולה גם תרוקן את קובץ הדואר היוצא של המשתמש.

המעטפת תהיה כתובה ב-Bourne Shell, ותקרא `transport_agent.sh`.



```
./transport_agent.sh
```

דוגמת הרצה

שינוי סטטוס הודעה

כאשר הודעה נקראת, יש לסמן עובדה זו בקובץ האינדקס. כאשר הודעה נמחקת, יש לסמן אותה כמחוקה.

כתבו תוכנית Bourne Shell קצרה אשר תקבל שני ארגומנטים - מספר הודעה וסטטוס חדש, ותשנה את סטטוס ההודעה באינדקס של המשתמש שהריץ אותה.

התוכנית תיקרא `set_status.sh`. לא לשכוח לבצע את החלק העיקרי שלה באמצעות `sed`.



```
./set_status.sh 3 Read
```

דוגמת הרצה

תוכנית הדואר

והתוכנית עצמה – המאפשרת לקרוא ולמחוק הודעות.

היא **תקרא** `mailer.pl` **ותהיה כתובה ב-Perl**. היא תציג למשתמש את רשימת כל ההודעות (חלא מחוקות) בתיבת הדואר הנכנס שלו. היא תאפשר לו לבחור הודעות ולהציגן, ותסמן כל הודעה שהוצגה כ- "נקראה" (באמצעות `set_status.sh` שנכתב מוקדם יותר). היא תאפשר לו למחוק הודעות כרצונו.

יש כמובן לוודא תקינות הפלט לתוכנית (לא ניתן למחוק הודעה שאינה קיימת, לדוגמה).



```
./mailer.pl
```

דוגמת הרצה