

הרצאה 2 - עיבוד טקסט, הפניות וצינורות

Text Processing, Redirection and Pipes

תוכן עניינים

- הצגת קבצים — cat, more, less
- צפייה בחלקי קובץ — tail ו head
- ספירה — wc
- הפנית קלט/פלט — Redirection
- הוספה לקובץ — Append
- צינורות — Pipes
- חיתוך שדות — cut
- הדבקת קבצים — paste
- מיון — sort
- הסרת כפילויות — uniq
- החלפת תווים — tr
- דוגמאות משולבות
- סיכום

1. הצגת קבצים — cat, more, less

טקסט הוא הממשק העיקרי של מערכות Linux/UNIX. הקודות הבאות מציעות תוכן של קבצי טקסט:

מיקוד	תיאור
<code>cat file</code>	מציגה את כל תוכן הקובץ
<code>more file</code>	מציגה מסך אחד בכל פעם — גלילה למטה בלבד
<code>less file</code>	כמו <code>more</code> אבל ניתן לעלות ולמטה ולחפש

2. צפייה בחלקי קובץ — tail ו head

tail — סוף הקובץ

```
tail [options] [files]
```

ברירת מחדל: 10 שורות אחרונות.

- `-n` X — שורות אחרונות
- `-v` — מושרה X ואילך
- `-f` X — בתים אחרונים
- `-c` X — מבית X ואילך
- `-f` — מעקב חי אחר קובץ (log)

head — תחילת הקובץ

```
head [options] [files]
```

ברירת מחדל: 10 שורות ראשונות.

- `-n` X — שורות ראשונות
- `-v` — הכל חוץ מ-X האחרונות
- `-c` X — בתים ראשונים
- `-c` X — הכל חוץ מ-X בתים אחרונים

דוגמאות (על הקובץ phone_dir.txt)

```
head -n3 phone_dir.txt
ADAMS, Andrew 7583 BARRETT, Bruce 6466 BAYES, Ryan 6585

tail -n3 phone_dir.txt
NAKAMURA, Satoehi 6453 REEVE, Shirley 7391 ROSNER, David 6830

tail -n12 phone_dir.txt # from line 12 onwards
MEASDAY, David 6494 NAKAMURA, Satoehi 6453 REEVE, Shirley 7391 ROSNER, David 6830
```

3. ספירה — wc (Word Count)

הפקודת wc

— סופרת שורות, מילים ובתים בקובץ.

- `-l` — מספר שורות
- `-w` — מספר מילים
- `-c` — מספר בתים

ללא דגלים — מציגה את שלושתם.

4. הפנית קלט/פלט — Redirection

כל קודורה ב-Linux מקבלת קלט מ-stdin (מקלדת) ושולחת פלט ל-stdout ול-stderr (מסך).

אופרטור	תפקיד	דוגמה
<code>></code> או <code>>></code>	הפנית stdout לקובץ (דריסה)	<code>ls > files</code>
<code>>></code>	הפנית stdout לקובץ	<code>ls "a b" >> err</code>
<code>>>></code>	הפנית stdout + stderr לקובץ	<code>ls "a b" >>> all</code>
<code><</code>	הפנית קובץ כ-stdin לפקודה	<code>wc < file.txt</code>

דוגמה: הפרדת stdout ו stderr

```
ls -l *.txt *.tmp 1>out 2>err # stdout to 'out', stderr to 'err'
ls -l *.txt *.tmp 1>out 2>>err # stderr appended to 'err'
```

הערה
הפנית קלט עם `<` אינה נפוצה — רוב הפקודות מקבלות קובץ כארגומנט, ולשאר ניתן להשתמש ב-pipes.

5. הוספה לקובץ — Append

ההבדל בין > ל >>

- `>` — דורש את הקובץ (יוצר חדש אם לא קיים)
- `>>` — מוסיף לסוף הקובץ (append) ללא דריסה

דוגמה

```
echo first line > lines.txt
echo second line >> lines.txt
echo third line >>> lines.txt
cat lines.txt
first line
second line
third line

echo fourth line > lines.txt # overwrite!
cat lines.txt
fourth line
```

6. צינורות — Pipes

מהו Pipe?
Pipe (|) לוקח את ה-stdout של פקודה אחת ומעביר אותה כ-stdin של פקודה אחרת.

```
command1 [args] | command2 [args]
```

למה זה חשוב?
ניתן לשרשר מספר פקודות — ליצור כליים מורכבים מאבני בניין פשוטות. אין צורך בקבצים זמניים — הנענים זורמים ישירות. סדרת pipes נקראת לפעמים `filters`. Pipes הם סוג פשוט של **inter-process communication (IPC)**.

דוגמאות

```
# Count .txt files in current directory
ls *.txt | wc -w

# Print lines 3-4 of phone_dir.txt
head -n4 phone_dir.txt | tail -n2
BAYES, Ryan 6585 BECK, Bill 6346

# Same result, different approach
tail -n3 phone_dir.txt | head -n2

# Most recently changed file
ls -t | head -n1
```

7. חיתוך שדות — cut

הפקודת cut — מרסיסה תת-קבוצה של תווים/שדות מכל שורה.

- `-c` 3,4,6,9 — חיתוך לפי **דגלים** (לדוגמה: -c3-4,6,9)
- `-d` - — חיתוך לפי **שדות** (לדוגמה: -d-)
- `-d` - — הגדרת תו **מפריד** (delimiter) ברירת מחדל: tab
- RANGE** מוגדר עם מקפים () ופסיקים (,).

דוגמאות

```
# First 2 first names (space as delimiter, field 2)
cut -f2 -d" " phone_dir.txt | head -n2
Andrew Bruce

# Last 3 last names (comma as delimiter, field 1)
cut -f1 -d",", phone_dir.txt | tail -n3
NAKAMURA REEVE ROSNER
```

8. הדבקת קבצים — paste

הפקודת paste — שמה קבצים זה לצד זה (side by side), שורה מול שורה.

- `-s` — תו מפריד בין הקבצים. ברירת מחדל: tab

דוגמאות

```
# Paste phone_dir.txt alongside zipcodes, separated by tab
paste -d"|" phone_dir.txt zipcodes | head -n3
ADAMS, Andrew 7583 34980 BARRETT, Bruce 6466 12567 BAYES, Ryan 6585 87492

# Repeat each line 3 times, separated by colons
paste -d":" zipcodes zipcodes zipcodes
34980:34980:34980 12567:12567:12567 ...
```

9. מיון — sort

הפקודת sort — ממיינת שורות לפי סדר לקסיקוגרפי (אלפביתי) כברירת מחדל.

- `-c` — סדר הפוך
- `-n` — מיון מספרי (9 לפני 21, במקום 21 לפני 9)
- `-k` 3,5 — מיון לפי **מפתח** ממנילה את N עד סוף המנילה-M ברירת מחדל: מפריד = whitespace (רווח, tab).

הבדל חשוב: k2,2 מול k2

- `-k2,2` — ממיינ לפי שדה 2 בלבד. שוויונות נשמרות בסדר המקורי.
- `-k2` — ממיינ לפי שדה 2 ועד סוף השורה (שדות 2, 3, 2... מאוחדים למחרוזת אחת).

מפתחות מיון ראשיים ומשניים

```
# Primary: field 2 numerically, Secondary: field 3 numerically
sort -k2,2n -k3,3n fruits.txt
date 1 10 banana 1 20 apple 2 10 cherry 2 5 fig 3 7

# Get first names + phone numbers, sort by name then by number
cut -f2-3 -d" " phone_dir.txt | sort -k1,1 -k2,2n
```

10. הסרת כפילויות — uniq

הפקודת uniq — מאחדות שורות עקבות זהות לשורה אחת.

- `-c` — מוסיפה ספירה (count) לפני כל שורה
- שינוי לב** — מוסיפה חתימה רק כפילויות **רצופות** לכן כמעט תמיד משלבים עם `sort` קודם.

דוגמאות

```
cat repeat.txt uniq repeat.txt sort repeat.txt | uniq
first line
first line first line
second line second line first line
third line third line first line third line

sort repeat.txt | uniq -c
3 first line 1 second line 1 third line
```

11. החלפת תווים — tr (Translate)

הפקודת tr — מחליפה כל תו מ-set1 לתו מ-set2.

חשוב: קוראת מ-stdin בלבד (אין ארגומנט קובץ). יש להשתמש עם pipe או `<`.

טווחים ומחלקות תווים

תחביר	משמעות
<code>"a-z"</code>	כל האותיות הקטנות
<code>"A-Z"</code>	כל האותיות הגדולות
<code>"a-zA-Z"</code>	D, E, F וכל אותיות א עד z
<code>[alpha]</code>	כל האותיות
<code>[upper]</code>	אותיות גדולות בלבד
<code>[lower]</code>	אותיות קטנות בלבד
<code>[digit]</code>	כל הספרות
<code>[space]</code>	רווח, newline, tab
<code>[punct]</code>	סימני פיקוח

דוגמאות

```
# Convert all uppercase to lowercase
cat phone_dir.txt | tr [:upper:] [:lower:]
adams, andrew 7583 barrett, bruce 6466 ...

# Swap case (upper-lower)
cat phone_dir.txt | tr "[:upper:] [:lower:] [:lower:] [:upper:]"
adams, aNDREW 7583 baRRett, BRUCE 6466 ...

# Delete all uppercase letters
echo 'AABaBcD' | tr -d [:upper:]
aBcD
```

12. דוגמאות משולבות — File Processing with Pipes

Task I — רשימת שמות פרטיים ייחודיים

```
# Step 1: cut second column
cut -f2 -d" " phone_dir.txt # Step 2: sort
cut -f2 -d" " phone_dir.txt | sort # Step 3: remove duplicates
cut -f2 -d" " phone_dir.txt | sort | uniq
Abe Andrew Bill ...
```

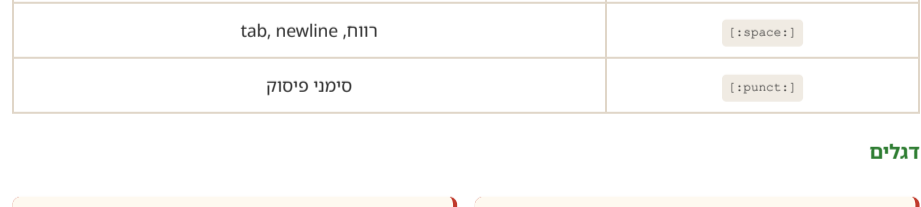
Task II — כמה שמות מתחילים בכל אות?

```
# Steps 1-2: cut first char of first name
cut -f2 -d" " phone_dir.txt | cut -c1 # Steps 3-4: sort and count unique
cut -f2 -d" " phone_dir.txt | cut -c1 | sort | uniq -c
2 A 2 B 2 D 1 J ...
```

Task III — ספר טלפונים "שנוי" עם מספרים מעורבים

```
# Step 1: cut 3rd column, sort numerically
cut -f3 -d" " phone_dir.txt # Step 2: paste alongside files, separated by dash
paste -d"-" phone_dir.txt nums.txt # Step 3: sort
cat phone_dir.txt | sort -n > nums.txt # Step 4: sort
paste -d"-" phone_dir.txt nums.txt
ADAMS, Andrew 7583-9749 BARRETT, Bruce 6466-7583 BAYES, Ryan 6585-7494 ...
```

13. סיכום ההרצאה



נקודות עיקריות

- הצגת קבצים:** `tail`, `head`, `less`, `more`, `cat`.
- ספירה:** `wc` (שורות), `wc -w` (מילים), `wc -c` (בתים).
- Redirection:** `>` (דריסה), `>>` (הוספה), `>>>` (הוספת stderr ו stdout).
- Pipes (|):** מעבירים stdout של פקודה אחת ל-stdin של הבאה. ניתן לשרשר.
- Text delimited tables:** `cut` (חיתוך/שדות/תווים), `paste` (הדבקת side-by-side), `sort` (מיון).
- uniq:** הסרת שורות עקבות זהות. קוראת מ-stdin בלבד. דגלים: `-c` (מספרים), `-s` (לפי סדר), `-k` (מפתח).
- tr:** החלפה/מחיקה/אחידות תווים. קוראת מ-stdin בלבד. דגלים: `-d` (squeeze), `-d` (delete).

טיפ לשיעורי בית

- השתמשו **בתוים יחסיים** (relative paths).
- השתמשו רק בפקודות/דגלים שנלמדו בהרצאות ובתרגילים.
- דאגו שהפתרון עובד על **השרת** (במידת האפשר hidden characters ומערכות הפעלה שונות).