

```

basicsys@mars~/lec11>cat P1
BEGIN{ s="ab12cd34"
  x=gsub("[1-3]","&&",s)
  print "x=" x
  print "s=" s
}

basicsys@mars~/lec11>awk -f P1
x=3
s=ab1122cd334

basicsys@mars~/lec11>cat P10
BEGIN{ s="ab12cd34"
  x=sub("[1-3]","&&",s)
  print "x=" x
  print "s=" s
}

basicsys@mars~/lec11>awk -f P10
x=1
s=ab112cd34

basicsys@mars~/lec11>cat F1
30
200
10

basicsys@mars~/lec11>cat P1
BEGIN {
  system("sort -n F1")
}

basicsys@mars~/lec11>awk -f P1
10
30
200

```

```
basicsys@mars~/lec11>cat P1
BEGIN {
    system("echo abc>|F2")
    system("echo def>>F2")
    system("cat F2")
}
```

```
basicsys@mars~/lec11>awk -f P1
abc
def
```

```
basicsys@mars~/lec11>cat P1
BEGIN {
    getline <"F1"
    print $0
    getline <"F1"
    print $0
}
```

```
basicsys@mars~/lec11>awk -f P1
30
200
```

```
basicsys@mars~/lec11>cat P1
BEGIN {
    getline <"F1"
    print $0
    getline <"F1"
    print $0
    close("F1")
    getline <"F1"
    print $0
}
```

```
basicsys@mars~/lec11>awk -f P1
30
200
30
```

```
basicsys@mars~/lec11>cat P1
BEGIN {
  getline x <"F1"
  print "$0=" $0
  print "x=" x
  getline x <"F1"
  print "$0=" $0
  print "x=" x
}
```

```
basicsys@mars~/lec11>awk -f P1
```

```
$0=
x=30
$0=
x=200
```

```
basicsys@mars~/lec11>cat P1
BEGIN {
  y=getline x<"F1"
  print "x=" x,"y=" y
  y=getline x<"F1"
  print "x=" x,"y=" y
  y=getline x<"F1"
  print "x=" x,"y=" y
  y=getline x<"F1"
  print "x=" x,"y=" y
  y=getline z<"F1"
  print "z=" z,"y=" y
}
```

```
basicsys@mars~/lec11>awk -f P1
x=30 y=1
x=200 y=1
x=10 y=1
x=10 y=0
z= y=0
```

```
basicsys@mars~/lec11>cat P1
BEGIN {
  A[8]="ab"
  A[30]=15
  A[4]=70
  system("echo -n "" >|tmp")
  for (x in A) {
    system("echo " x " >>tmp")}
  system("sort -n tmp>|tmp1")
  while (getline x < "tmp1") {
    print x,A[x]
  }
}
```

```
basicsys@mars~/lec11>awk -f P1
```

```
4 70
8 ab
30 15
```

```
basicsys@mars~/lec11>cat P1
{ print $0
  getline
  print $0
}
```

```
basicsys@mars~/lec11>awk -f P1 F1
30
200
10
10
```

```
basicsys@mars~/lec11>cat P1
{ print $0
  getline x
  print $0
}
```

```
basicsys@mars~/lec11>awk -f P1 F1
30
30
10
10
```

```
basicsys@mars~/lec11>cat F2
ab
cd
```

```
basicsys@mars~/lec11>cat P1
{ print $0
  getline x<"F2"
  print "x=" x
  print $0
}
```

```
basicsys@mars~/lec11>awk -f P1 F1
30
x=ab
30
200
x=cd
200
10
x=cd
10
```

## הפונקציות sub ו-gsub ב-awk (המשך)

הפונקציות sub ו-gsub מחזירות את מספר ההחלפות שבוצעו. עבור sub המספר המוחזר הוא 0 או 1 עבור gsub המספר המוחזר יכול להיות גדול מ-1. לדוגמה, נניח ש-`s="abcdababef"` הפקודה `gsub("(ab)+",xy,s)` מחזירה 2 כי בוצעו שתי החלפות: אחת של המחרוזת ab והשניה של המחרוזת abab. לפיכך, אם `s="abcdababef"` ונבצע `n = gsub("(ab)+",xy,s)` המשתנה n יקבל ערך 2.

במחרוזת ההחלפה בפקודות sub ו-gsub אפשר להשתמש בסימן המיוחד & שמשמעותו: החלף את & בחלק של המחרוזת שהתאים לביטוי הרגולארי. לדוגמה, אם `s="12abc354"` ונבצע `gsub("[1-4]","&&",s)` אזי המשתנה s יקבל ערך 1122abc33544.

## הפונקציה system ב-awk

מבנה הפונקציה: `system(מחרוזת)`

הפונקציה מבצעת את המחרוזת כפקודת bash.

לדוגמה, הפקודה `system("sort F1")` תציג על המסך את שורות הקובץ F1 ממוינות בסדר לכסיקוגרפי עולה (בהתאם לפקודה sort של bash).

## הפונקציה getline ב-awk

יש מספר אפשרויות שימוש בפונקציה:

`getline`

שם משתנה `getline`

`getline < "שם קובץ"`

`getline < "שם קובץ" < שם משתנה`

להלן פרוט המשמעות של כל אחת מהאפשרויות הנ"ל.

`getline` גורמת לכך שמתבצע מעבר לשורת הקלט הבאה באופן מידי (ולא בסיום תוכנית ה-awk). המשתנים `NF,NR,FNR,$0,$1,$2,...` מתעדכנים לפי שורת הקלט הבאה. כאשר תוכנית ה-awk תגיע לסיימה יתבצע מעבר לשורת קלט נוספת (כפי שקורה בכל סיום תוכנית awk).

**שם משתנה getline** גורמת לכך שערך המשתנה מתעדכן לפי שורת הקלט הבאה. פרט לכך לא מתבצע כל שינוי נוסף, ולכן ערכי המשתנים `getline x NF,NR,FNR,$0,$1,$2,...` לדוגמה לאחר הפקודה `getline x` ערך המשתנה `x` ישווה לשורת הקלט הבאה והמשתנה `$0` ישאר ללא שינוי. כאשר תוכנית ה-`awk` תגיע לסיימה יתבצע מעבר לשורת הקלט הבאה (כפי שקורה בכל סיום תוכנית `awk`), ואז הערך של המשתנה `$0` יהיה שווה לערך המשתנה `x`.

**"שם קובץ" <getline** גורמת לכך שערכי המשתנים `NF,$0,$1,$2,...` מתעדכנים לפי שורת הקלט הבאה מתוך הקובץ. ערכי המשתנים `NR` ו-`FNR` לא מתעדכנים. לדוגמה, לאחר הפקודה `"F1" <getline` ערכי המשתנים `NF,$0,$1,$2,...` מתעדכנים לפי השורה הראשונה בקובץ `F1`. לאחר פקודה נוספת של `"F1" <getline` ערכי המשתנים הנ"ל מתעדכנים לפי השורה השנייה בקובץ `F1`, וכן הלאה. הפונקציה (`"F1" <close`) סוגרת את הקובץ `F1` וגורמת לכך שלאחריה הפקודה הבאה של `"F1" <getline` תעדכן את ערכי המשתנים הנ"ל לפי השורה הראשונה בקובץ.

**"שם קובץ" <שם משתנה getline** גורמת לכך שערך המשתנה מתעדכן לפי שורת הקלט הבאה מהקובץ. לפי שורת הקלט הבאה מתוך הקובץ. פרט לכך לא מתבצע כל שינוי נוסף, ולכן ערכי המשתנים `NF,NR,FNR,$0,$1,$2,...` לא מתעדכנים. לדוגמה, לאחר הפקודה `"F1" <getline x` ערך המשתנה `x` שווה לשורה הראשונה בקובץ `F1`. לאחר פקודה נוספת של `"F1" <getline x` ערך המשתנה `x` שווה לשורה השנייה בקובץ `F1` וכן הלאה.

בכל אחת מהצורות הנ"ל הפונקציה `getline` מחזירה 1 אם נקראה שורה. אם הפונקציה לא הצליחה לקרוא שורה כי הסמן נמצא בסוף הקובץ (בקריאה מקובץ) או בסוף הקלט (בקריאה מהקלט) אזי הפונקציה מחזירה 0. אם הפונקציה לא הצליחה לקרוא שורה בגלל שקרתה שגיאה אז היא מחזירה -1.