

ΕΡΓΑΣΤΗΡΙΟ 2

Κανονικές εκφράσεις – grep & awk

Άσκηση 1. Δημιουργήστε ένα νέο directory κάτω από τον προσωπικό σας κατάλογο με όνομα της επιλογής σας. Προηγουμένως ελέγξτε αν το όνομα υπάρχει και στη συνέχεια αν έχει δημιουργηθεί.

```
amarg@amarg-vbox:~$ mkdir testDir
```

Άσκηση 2. Από τον κατάλογο /etc/ αντιγράψτε στο νέο directory το αρχείο passwd, με όνομα passwd1. Πριν από την αντιγραφή ελέγξτε (με το μάτι, όχι με κώδικα – αυτό θα το δούμε στο επόμενο εργαστήριο) ότι το αρχείο υπάρχει. Μετά την αντιγραφή ελέγξτε αν η αντιγραφή έγινε σωστά.

```
amarg@amarg-vbox:~$ cp /etc/passwd ~/testDir/passwd1
```

Άσκηση 3. Να δημιουργήσετε ένα νέο αρχείο passwd2 που θα περιέχει μόνον τις γραμμές του passwd1 που αρχίζουν με τα γράμματα A-M και a-m.

```
amarg@amarg-vbox:~$ grep '^[A-Ma-m]' ~/testDir/passwd1
```

Η παραπάνω κανονική έκφραση επιστρέφει τις γραμμές εκείνες του αρχείου /etc/passwd1 που ξεκινούν από κεφαλαίο γράμμα A έως M ή από μικρό γράμμα a έως m. Αυτό είναι κάτι που το δηλώνουμε ως [A-Ma-m]. Προκειμένου να δείξουμε πως αυτό το γράμμα είναι το πρώτο γράμμα της γραμμής, βάζουμε μπροστά από την παραπάνω έκφραση το χαρακτήρα ^. Το αποτέλεσμα αυτής της διαδικασίας παρουσιάζεται στη συνέχεια.

```
amarg@amarg-vbox:~$ grep '^[A-Ma-m]' ~/testDir/passwd1
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:109:116:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
cups-pk-helper:x:113:120:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
avahi:x:115:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/usr/sbin/nologin
hplip:x:119:7:HPLIP system user,,,:/run/hplip:/bin/false
colord:x:121:126:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:122:127::/var/lib/geoclue:/usr/sbin/nologin
gnome-initial-setup:x:124:65534::/run/gnome-initial-setup:/bin/false
gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
amarg:x:1000:1000:Athanasios Margaritis,,,:/home/amarg:/bin/bash
```

Προκειμένου το παραπάνω αποτέλεσμα να αποθηκευτεί στο αρχείο ~/testDir/passwd2 ανακατευθύνουμε την έξοδο της grep ως

```
amarg@amarg-vbox:~$ grep '^[A-Ma-m]' ~/testDir/passwd1 > ~/testDir/passwd2
```

Για να επαληθεύσετε το αποτέλεσμα εκτελέστε την εντολή `cat ~/testDir/passwd2`.

Άσκηση 4. Να δημιουργήσετε ένα νέο αρχείο `passwd3` που θα περιέχει όλους τους χρήστες που χρησιμοποιούν φλοιό `bash`.

```
amarg@amarg-vbox:~$ grep 'bash$' ~/testDir/passwd1
root:x:0:0:root:/root:/bin/bash
amarg:x:1000:1000:Athanasios Margaris,,,:/home/amarg:/bin/bash
```

Η κανονική έκφραση που εμφανίζει μόνο τις γραμμές που τελειώνουν σε `bash` είναι η `'bash$'` με το αποτέλεσμα της εφαρμογής της στο αρχείο `~/testDir/passwd1` να απεικονίζεται παραπάνω. Για να αποθηκεύσουμε την έξοδο της `grep` στο αρχείο `passwd` χρησιμοποιούμε την εντολή

```
grep 'bash$' ~/testDir/passwd1 >> ~/testDir/passwd3
```

και επαληθεύουμε το αποτέλεσμα εκτελώντας την εντολή `cat ~/testDir/passwd3` όπως φαίνεται παρακάτω.

```
amarg@amarg-vbox:~$ grep 'bash$' ~/testDir/passwd1 >~/testDir/passwd3
amarg@amarg-vbox:~$
amarg@amarg-vbox:~$ cat ~/testDir/passwd3
root:x:0:0:root:/root:/bin/bash
amarg:x:1000:1000:Athanasios Margaris,,,:/home/amarg:/bin/bash
amarg@amarg-vbox:~$
```

Άσκηση 5. Χρησιμοποιείτε την εντολή `wc` για να μετρήσετε πόσες γραμμές αντιστοιχούν στις προηγούμενες δύο περιπτώσεις και να γράψετε το πλήθος τους στο τέλος των αρχείων `passwd2`.

```
wc -l ~/testDir/passwd2 >> ~/testDir/passwd2
wc -l ~/testDir/passwd3 >> ~/testDir/passwd2
```

Οι δύο παραπάνω εντολές είναι και αυτές που υλοποιούν τη ζητούμενη διαδικασία. Η εντολή `wc -l ~/testDir/passwd2` μετράει τις γραμμές του αρχείου `~/testDir/passwd2` και εκτυπώνει το αποτέλεσμα στην οθόνη. Προκειμένου να αποθηκεύσουμε αυτό το αποτέλεσμα στο αρχείο **χωρίς αυτό να επανεγγραφεί** (δηλαδή να διαγραφούν τα υπάρχοντα περιεχόμενά του και να αντικατασταθούν από τα νέα) δεν χρησιμοποιούμε τον τελεστή `>` αλλά τον τελεστή `>>` ο οποίος ανοίγει το αρχείο, πηγαίνει στο τέλος του και γράφει το νέο περιεχόμενο αμέσως μετά το προηγούμενο **χωρίς** αυτό να διαγράφεται. Τα αποτελέσματα αυτών των εντολών ακολουθούν στη συνέχεια.

```
amarg@amarg-vbox:~$ wc -l ~/testDir/passwd2 >> ~/testDir/passwd2
amarg@amarg-vbox:~$ cat ~/testDir/passwd2
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailng List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
messagebus:x:103:106:/:nonexistent:/usr/sbin/nologin
avahi-autoipd:x:109:116:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
cups-pk-helper:x:113:120:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
avahi:x:115:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/usr/sbin/nologin
hplip:x:119:7:HPLIP system user,,,:/run/hplip:/bin/false
colord:x:121:126:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:122:127:/:/var/lib/geoclue:/usr/sbin/nologin
gnome-initial-setup:x:124:65534:/:/run/gnome-initial-setup:/bin/false
gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
amarg:x:1000:1000:Athanasios Margaris,,,:/home/amarg:/bin/bash
22 /home/amarg/testDir/passwd2
amarg@amarg-vbox:~$
```

```
amarg@amarg-vbox:~$ wc -l ~/testDir/passwd3 >> ~/testDir/passwd3
amarg@amarg-vbox:~$ cat ~/testDir/passwd3
root:x:0:0:root:/root:/bin/bash
amarg:x:1000:1000:Athanasios Margaris,,,:/home/amarg:/bin/bash
2 /home/amarg/testDir/passwd3
amarg@amarg-vbox:~$
```

Παρατηρήστε πως σε αμφότερες τις περιπτώσεις, η έξοδος της `wc` αποθηκεύεται στο τέλος του αρχείου χωρίς τα περιεχόμενα που ήδη υπήρχαν σε αυτό να έχουν διαγραφεί.

Άσκηση 6. Να δημιουργήσετε ένα αρχείο κειμένου με όνομα `test`. Στο αρχείο αυτό θα πρέπει να περιέχονται γραμμές με το παρακάτω περιεχόμενο (προφανώς θα περιέχονται και γραμμές με κανονικό κείμενο):

1. Κενές γραμμές
2. Γραμμές μήκους ενός χαρακτήρα
3. Γραμμές που περιλαμβάνουν αριθμούς που αντιστοιχούν στους αριθμούς μηνών (π.χ. 02, 2, 11, 9, 07 κλπ)
4. Γραμμές που περιλαμβάνουν δεκαδικούς αριθμούς με 1,2 ή 3 το πολύ δεκαδικά ψηφία και 5 το πολύ ακέραια ψηφία (π.χ. 27.59, 1.9, 4589.772)
5. Αναγνωριστικά (identifiers) της γλώσσας Java
6. Ώρα σε 12ωρη βάση με την ένδειξη `am` ή `pm`
7. Γραμμές που τελειώνουν με τελεία (.)

Στη συνέχεια να γράψετε τις κατάλληλες κανονικές εκφράσεις σε εντολές `grep` ώστε να εμφανίσετε στην οθόνη όλες τις γραμμές που αντιστοιχούν στα παραπάνω.

Για την επίλυση της άσκησης δημιουργούμε το αρχείο `test` με τα επόμενα ενδεικτικά περιεχόμενα (μπορείτε να κατεβάσετε αυτό το αρχείο από το `eclass` από το φάκελο Έγγραφα → Εργαστήρια → Λύσεις εργαστηρίων → Εργαστήριο 2 με όνομα `test.txt`)

```
1 Hello world !!
2 Have a nice day !!
3
4 r
5 1
6 $
7
8 02
9 2
10 11
11 9
12 07
13 Department of Digital Systems
14 University of Thessaly
15 Larissa
16
17
18
19 e
20 0
21 1
22 11
23 456.34
24 12.3434535
25 12132425.01
26 class
27 Mathematics
28 private
29 C++
30 protected
31
32
33
34 12:54
35 03:59
36 45:45
37 Hi.
38 Yes.
39
40
41 11
42
```

Τα ζητούμενα αποτελέσματα προκύπτουν ως εξής:

1. Κενές Γραμμές

```
amarg@amarg-vbox:~$ gedit test
amarg@amarg-vbox:~$ grep '^$' test
```

```
amarg@amarg-vbox:~$
```

2. Γραμμές μήκους ενός χαρακτήρα

```
amarg@amarg-vbox:~$ grep '^.$' test
```

```
r
1
$
2
9
e
0
1
```

```
amarg@amarg-vbox:~$
```

3. Γραμμές που περιλαμβάνουν αριθμούς που αντιστοιχούν στους αριθμούς μηνών (π.χ. 02, 2, 11, 9, 07 κ.τ.λ)

```
amarg@amarg-vbox:~$ egrep '^(0?[1-9]|1[012]))$' test
```

```
1
2
11
9
07
1
11
11
```

```
amarg@amarg-vbox:~$
```

4. Γραμμές που περιλαμβάνουν δεκαδικούς αριθμούς με 1,2 ή 3 το πολύ δεκαδικά ψηφία και 5 το πολύ ακέραια ψηφία (π.χ. 27.59, 1.9, 4589.772)

```
amarg@amarg-vbox:~$ grep '^([0-9]{0,5}\.{0,4})[0-9]{0,2}$' test
456.34
amarg@amarg-vbox:~$
```

5. Αναγνωριστικά (identifiers) της γλώσσας Java

```
amarg@amarg-vbox:~$ egrep 'class|private|protected' test
class
private
protected
amarg@amarg-vbox:~$
```

6. Ώρα σε 12ωρη βάση

```
amarg@amarg-vbox:~$ egrep '^([0-9]|1[01]):[0-5][0-9]$' test
03:59
amarg@amarg-vbox:~$
```

7. Που τελειώνουν με τελεία

```
amarg@amarg-vbox:~$ grep '^.*\.$' test
Hi.
Yes.
amarg@amarg-vbox:~$
```

Άσκηση 7. Να κατασκευαστεί κανονική έκφραση που να αναγνωρίζει διευθύνσεις email του Πανεπιστημίου Θεσσαλίας με login name το οποίο να αποτελείται από τουλάχιστον οκτώ γράμματα (κεφαλαία ή/και μικρά) ή/και αριθμούς.

Η ζητούμενη κανονική έκφραση έχει την ακόλουθη μορφή σύμφωνα με την οποία οι λέξεις που παράγονται ή γίνονται δεκτές από την κανονική έκφραση έχουν τουλάχιστον 8 ($\{8,\}$) χαρακτήρες που μπορεί να είναι κεφαλαίοι ($[A-Z]$) ή μικροί ($[a-z]$) αγγλικοί χαρακτήρες ή ψηφία ($[0-9]$) ενώ αμέσως μετά του οκτώ χαρακτήρες ακολουθεί η σταθερή υποσυμβολοσειρά @uth.gr η οποία πρέπει να εμφανίζεται αυτούσια ως έχει (αφού όλα τα emails του Πανεπιστημίου Θεσσαλίας τελειώνουν με αυτή).

$[A-Za-z0-9]\{8,\}@uth.gr$

Για τον έλεγχο της ορθότητας της κανονικής έκφρασης κατασκευάσαμε το αρχείο emails περιεχόμενα (μπορείτε να κατεβάσετε αυτό το αρχείο από το eclass από το φάκελο Έγγραφα → Εργαστήρια → Λύσεις εργαστηρίων → Εργαστήριο 2 με όνομα test.txt) με περιεχόμενο

```
amarg@amarg-vbox:~/shared$ cat emails.txt
amarg@uom.gr
atmargaris@uth.gr
isavvas@uth.gr
amarg200x@yahoo.gr
amarg@teilar.gr
karetsos@uth.gr
amarg@amarg-vbox:~/shared$
```

Στην περίπτωση αυτή, η χρήση της egrep με αυτή την κανονική έκφραση και το παραπάνω αρχείο θα μας δώσει

```
amarg@amarg-vbox:~/shared$ egrep '[A-Za-z0-9]\{8,\}@uth.gr' emails.txt
atmargaris@uth.gr
karetsos@uth.gr
amarg@amarg-vbox:~/shared$
```

και δεν είναι δύσκολο να διαπιστώσει κανείς, πως τα αποτελέσματα στα οποία οδηγεί είναι πράγματι σωστά.

Άσκηση 8. Δίδεται το επόμενο αρχείο κειμένου με όνομα `results` το οποίο περιέχει τις επιδόσεις ενός φοιτητή κατά τη διάρκεια ενός χρονικού διαστήματος.

PROGRAMMING	EAR	2010	7
DATABASES	XEIM	2011	6
UNIX	EAR	2010	4
NETWORKS	EAR	2010	10
JAVA	XEIM	2011	8
MATH1	XEIM	2010	2
PHYSICS	XEIM	2010	5
C++	XEIM	2011	7
ENGLISH	XEIM	2010	9
MATH2	EAR	2011	5

Χρησιμοποιώντας τη γλώσσα `awk` να εκτυπώσετε τα μαθήματα που πέρασε ο φοιτητής κατά την εξεταστική περίοδο `XEIM 2010` και να υπολογίσετε το μέσο όρο των βαθμών για αυτά τα μαθήματα.

Το πρόγραμμα `awk` που επιλύει το παραπάνω πρόβλημα μπορεί να κληθεί μέσα από τη γραμμή εντολών ως

```
awk 'BEGIN { count=0; sum=0; mean } /XEIM 2010/ { sum+=$4; count++; print $0 } END { mean=sum/count; print "Mean="mean}' results
```

ή εναλλακτικά, να αποθηκευτεί σε αρχείο το οποίο θα περαστεί ως όρισμα στην `awk` ως

```
awk -f awkscript results
```

όπου το αρχείο `awkscript` έχει το περιεχόμενο που ακολουθεί στη συνέχεια

BEGIN { count=0; sum=0; mean=0 }	Το <code>BEGIN</code> block περιέχει εντολές που εκτελούνται μία και μοναδική φορά στην αρχή. Εδώ συνήθως δηλώνονται και αρχικοποιούνται μεταβλητές. Ο ζητούμενος μέσος όρος θα προκύψει ως $mean = sum / count$ όπου <code>count</code> το πλήθος των μαθημάτων που πληρούν το κριτήριο και <code>sum</code> το άθροισμα των βαθμών για αυτά τα μαθήματα.
/XEIM 2010/ { sum+=\$4; count++; print \$0 }	Ο κώδικας που ακολουθεί θα εκτελεστεί ΜΟΝΟ για εκείνες τις γραμμές που περιέχουν οπουδήποτε τη συμβολοσειρά <code>XEIM 2010</code> . Για κάθε τέτοια γραμμή το <code>count</code> αυξάνεται κατά ένα η τιμή του <code>\$4</code> που περιέχει το βαθμό του μαθήματος προστίθεται στο <code>sum</code> ενώ ολόκληρη η γραμμή (<code>\$0</code>) εκτυπώνεται στην οθόνη του χρήστη.
END { mean=sum/count; print "Mean="mean }	Το <code>END</code> block περιέχει εντολές που εκτελούνται μία και μοναδική φορά στο τέλος. Οι μεταβλητές <code>sum</code> και <code>count</code> έχουν τις σωστές τιμές οπότε υπολογίζουμε το μέσο όρο τους ως $mean = sum / count$ και εκτυπώνουμε το αποτέλεσμα.

(μπορείτε να κατεβάσετε το αρχείο `results` από το `eclass` από το φάκελο Έγγραφα → Εργαστήρια → Λύσεις εργαστηρίων → Εργαστήριο 2 με όνομα `test.txt`).

Το αποτέλεσμα είναι προκύπτει ως εξής:

```
amarg@amarg-vbox:~$ awk -f awkscript results
MATH1      XEIM 2010    2
PHYSICS    XEIM 2010    5
ENGLISH    XEIM 2010    9
Mean=5,33333
```