



Εργαστήριο 6

Ασκήσεις: Το Εργαλείο Ωφελιμότητας Awk – Προγραμματισμός Κελύφους Bash

- 1) Τυπώστε όλες τις γραμμές του αρχείου myfile για τις οποίες το πρώτο πεδίο είναι διαφορετικό του πρώτου πεδίου της προηγούμενης.

```
$ awk '$1 != prev { print; prev = $1 }' myfile
```

- 2) Τυπώστε το συνολικό μέγεθος των αρχείων σε MBs που βρίσκονται στον τρέχον φάκελο και έχουν αλλαχτεί κατά το μήνα Νοέμβριο.

```
$ ls -l | awk '$6 == "Nov" { sum += $5 } END { print sum/1000000 " Mb" }'
```

Χωρίς να συμπεριλάβεις καταλόγους:

```
$ ls -l | grep ^- | awk '$6 == "Nov" { sum += $5 } END { print sum/1000000 " Mb" }'
```

```
ls -l | sed -n '/^[^d]/p' | awk '$6 == "Nov" { sum += $5 } END { print sum/1000000 " Mb" }'
```

- 3) Τυπώστε και ταξινομήστε με αντίστροφη αλφαβητική σειρά όλα τα login names (πρώτη στήλη) που βρίσκονται στο αρχείο /etc/passwd.

```
$ awk -F ":" '{ print $1 }' /etc/passwd | sort -r
```

ή

```
$ awk -F ":" '{ print $1 | "sort -r" }' /etc/passwd
```

- 4) Δεδομένου ενός αρχείου με τρεις στήλες, γράψτε ένα awk script το οποίο προσθέτει τις στήλες κάθε γραμμής και υπολογίζει το άθροισμά τους. Στην πρώτη γραμμή εξόδου θα πρέπει να εμφανίζεται η φράση "Print Totals" και στην τελευταία η "End Totals". Το αποτέλεσμα πρέπει να φαίνεται ως μαθηματικές πράξεις, δηλαδή στήλη1 + στήλη2 + στήλη3 = αποτέλεσμα.

```
$ BEGIN {print "Print Totals"}
      {total = $1 + $2 + $3}
      {print $1 " + " $2 " + " $3 " = "total}
      END {print "End Totals"}
```



5) Τυπώστε κάθε πεδίο κάθε γραμμής του αρχείου filename σε ξεχωριστή γραμμή.

```
$ awk '{for(i=1;i<=NF;i++) print $i }' filename
```

6) Τυπώστε τα πεδία της κάθε γραμμής του αρχείου myfile σε αντίστροφη σειρά.

```
$ awk '{ for (i = NF; i > 0; --i) printf("%s", $i); printf("\n")}'
myfile
```

7) Δεδομένου του αρχείου filename το οποίο περιέχει λιγότερες από 200 γραμμές κειμένου, εκτυπώστε τις γραμμές του filename σε αντίστροφη σειρά.

```
awk ' { line[NR] = $0 }
END { for ( i = NR; i > 0; i--) print line[i] } ' filename
```

ή

```
awk '{ line[i++]=$0}
END {for (j=i-1; j>=0;) print line [j--] }' filename
```

8) Να γραφεί μια απλή υπολογιστική μηχανή (έστω ονομάζεται math) στο κέλυφος Bash, η οποία να υποστηρίζει απλές αριθμητικές πράξεις (**πρόσθεση, αφαίρεση, πολλαπλασιασμό, διαίρεση και υπόλοιπο**) μεταξύ δύο ακεραίων.

Λύση:

```
#!/bin/bash
# Usage: math n1 op n2
#
case $2 in
+) echo "Addition requested."
  echo "$1 + $3 = $(( $1+$3 )) " ;;
-) echo "Subtraction requested."
  echo "$1 - $3 = $(( $1-$3 )) " ;;
\*) echo "Multiplication requested."
  echo "$1 * $3 = $(( $1*$3 )) " ;;
/) echo "Division requested."
  echo "$1 / $3 = `expr $1 / $3`" ;;
%) echo "Modulo arithmetic requested."
  echo "$1 % $3 = $(( $1%$3 )) " ;;
*) echo "Unknown operation specified." ;;
esac
```



9) Εξηγήστε τι κάνει το πιο κάτω πρόγραμμα του κελύφους Bash.

```
#!/bin/bash
cur_date=`date +%d%m%y`
last_date=""

if [ -s $HOME/.date ]; then
    last_date=`cat $HOME/.date`
fi

echo "$cur_date" > $HOME/.date

if [ "$last_date" != "$cur_date" ]; then
    echo "Hello!! How are you today?"
fi
```

Λύση:

Το πιο πάνω πρόγραμμα Bash εκτυπώνει στην οθόνη κάποιο το μήνυμα *"Hello!! How are you today?"* μόνο την πρώτη φορά που εκτελείται κατά τη διάρκεια μίας ημέρας.



10) Να γραφεί ένα πρόγραμμα στο κέλυφος Bash, το οποίο αλλάζει όλα τα ονόματα αρχείων που περιέχουν κεφαλαία γράμματα, σε μικρά γράμματα.

Λύση:

```
#!/bin/bash
# Usage: rename
#
for file in *
do
    lfile=`echo $file | tr A-Z a-z`
    #lfile=`echo $file | tr '[:upper:]' '[:lower:]'`
    if [ $file != $lfile ]; then
        mv $file $lfile
    fi
done
```



11) Να γραφεί ένα πρόγραμμα για το κέλυφος Bash (έστω ονομάζεται revstrs) που να αντιστρέφει δεδομένες συμβολοσειρές, καθώς επίσης να υπολογίζει και τα μήκη τους.

Hint: Χρησιμοποιηστε τις ακολουθες εκφρασεις:

`length=`expr length "string" ή ${#string}`

Επιστρέφει το μήκος του string;

``expr substr $string $position $length``

Επιστρέφει length χαρακτήρες από το string ξεκινώντας από το position

Λύση:

```
#!/bin/bash
#
# Usage: revstrs [string1 [string2 ...]]
#
for str
do
    strlen=`expr length $str` # ${#string}
    idx=$strlen
    while [ $idx -gt 0 ]
    do
        # Επέστρεψε 1 χαρακτήρα από το $str ξεκινώντας από το idx
        # expr substr $string $position $length`
        echo -n "`expr substr $str $idx 1`"
        ((idx--))
    done
    echo " --> $strlen character(s)."
done
```

Λύση με την χρήση πινάκων:

```
#!/bin/bash
declare -a ARRAY
for str
do
    strlen=${#str}
    i=0

    echo " $strlen character(s): "
    while [ ! $i -gt $strlen ]
    do
        # expr substr $string $position $length`
        ARRAY[$i]="`expr substr $str $i 1`"
        echo ${ARRAY[$i]}
        ((i++))
    done
    echo ${ARRAY[*]} # Εκτύπωση όλων στοιχείων ARRAY
done
```