



Εργαστήριο 1

Προχωρημένες Εντολές Unix (grep, ps, cut, find)

- 1) Βρείτε τον αριθμό των αρχείων στον τρέχον φάκελο τα οποία έχουν χρόνο τροποποίησης μεταξύ 08.00 – 08.59 π.μ. στις 20 Οκτωβρίου.

```
ls -l | grep '08:[0-5][0-9]' | grep "Oct 20" | wc -l
ls -l | grep '08:[0-5][0-9]' | grep -c "Oct 20"
```

Επεξήγηση:

Το αποτέλεσμα της εντολής `ls` περνιέται σαν είσοδος στην εντολή `grep` έτσι ώστε να φιλτράρουμε όλες τις γραμμές που περιέχουν τις γραμματοσειρές που συμβολίζουν τις ώρες 08:00 – 08:59. Στην συνέχεια η έξοδος ξανά φιλτράρεται με την εντολή `grep` έτσι ώστε να μείνουν μόνο οι γραμμές που περιέχουν την ημερομηνία που θέλουμε και τέλος μετρούμε τις γραμμές με τη χρήση της εντολής `wc`.

- `ls-l` : Εκτύπωση long list `ls`
- `grep '08:[0-5][0-9]'` : Παραμένουν όσα έχουν τις ώρες μεταξύ 08:00 και 08:59
- `grep "Oct 20"` : Παραμένουν όσα έχουν την συμβολοσειρά "Oct 20"
- `wc -l` : Μέτρηση των γραμμών

- 2) Βρείτε όλες τις λέξεις που αποτελούνται από ακριβώς πέντε γράμματα από το λεξικό `/usr/share/dict/words`. Πόσες είναι αυτές οι λέξεις; Εμφανίστε τις ταξινομημένες σε αντίστροφη αλφαβητική σειρά εκτυπώνοντας σελίδα – σελίδα στην οθόνη

```
grep '^.....$' /usr/share/dict/words ή
grep -E '^[[:alpha:]]{5}$' /usr/share/dict/words ή
grep -E '^[a-zA-Z]{5}$' /usr/share/dict/words

grep '^.....$' /usr/share/dict/words | wc -l
grep '^.....$' /usr/share/dict/words | sort -r | more
```

- 3) Βρείτε το συνολικό αριθμό γραμμών των αρχείων του τρέχοντος φακέλου που περιέχουν το string `Linux` αλλά όχι το string `Unix`

The following commands require that the current working directory contains only files (not directories)

```
cat * | grep "Linux" | grep -v "UNIX" | wc -l
cat * | grep "Linux" | grep -vc "UNIX"
cat * | grep "Linux" | grep -v -c "UNIX"
```

Επεξήγηση:

In the first of the four segments of this pipeline, the `cat` command, which is used to read and concatenate (i.e., string together) the contents of files, concatenates the contents of all of the files in the current directory. The `asterisk` is a wildcard that represents all items in a specified directory, and in this case it serves as an argument to `cat` to represent all objects in the current directory.

The first pipe sends the output of `cat` to the `grep` command, which is used to search text. The `Linux` argument tells `grep` to return only those lines that contain the string `Linux`. The second



pipe sends these lines to another instance of `grep`, which, in turn, with its `-v` option, eliminates those lines that contain the string `UNIX`. Finally, the third pipe sends this output to `wc -l`, which counts the number of lines and writes the result to the display screen.

Εναλλακτικά

```
ls -F | grep -v '*$' | grep -v '/$' | xargs cat | grep 'Linux' |
grep -v 'Unix' | wc -l
```

Με την πιο πάνω εντολή αποφεύγουμε να τροφοδοτούμε την εντολή `cat` με αρχεία εκτελέσιμα και με καταλόγους (directories). Τα εκτελέσιμα αρχεία είναι δυαδικά αρχεία. Με τη χρήση του `ls -F` παρουσιάζονται τα εκτελέσιμα αρχεία με το σύμβολο `` στο τέλος του ονόματος. Με τις επόμενες 2 `grep` εντολές αφαιρούμε τα εκτελέσιμα και τους καταλόγους.*

If we have both files and folders, we need to obtain only files, so we need find:

```
find . -maxdepth 1 -type f -exec cat {} \; | grep "Linux" | grep -v
"UNIX" | wc -l
find . -maxdepth 1 -type f | xargs cat | grep "Linux" | grep -v
"UNIX" | wc -l
```

Note: `-exec` executes the command `cat` on each individual file whereas `xargs` constructs an argument list from the output of the `find` command and passes it to `cat`

- 4) Εκτυπώστε στην οθόνη όλες τις διεργασίες που ανήκουν στον χρήστη `root` ταξινομώντας τις βάση του χρόνου έναρξης της διεργασίας (ξεκινώντας από την πιο πρόσφατη)

```
ps -ef | grep '^root' | sort -k 5
ps -aux | grep '^root' | sort -k 9
```

- 5) Το αρχείο `/etc/passwd` περιέχει μια γραμμή για κάθε χρήστη του συστήματος. Κάθε χρήστης περιγράφεται από 7 πεδία τα οποία διαχωρίζονται με `:`. Τα πεδία αυτά είναι τα `Username:Password:UID:GID:UID Info:Home directory:Command/shell`. Θέλουμε να δημιουργήσουμε το αρχείο `users_info` το οποίο θα περιέχει μόνο τα πεδία `Username` και το `UID info` σε αλφαβητική σειρά του `UID info`.

```
cut -d: -f1,5 /etc/passwd | tr ':' ' ' | sort -k 2 > users_info
```

- 6) Δημιουργήστε το `tar` αρχείο `weekly.tar` το οποίο θα περιέχει όλα τα αρχεία (όχι φακέλους και ειδικά αρχεία) του τρέχοντος καταλόγου τα οποία έχουν τύχει αλλαγών σε 7 ή λιγότερες μέρες. Χρησιμοποιήστε τα `man` pages για περισσότερες πληροφορίες για τις επιλογές της δημιουργίας του `tar` αρχείου.

```
find . -maxdepth 1 -type f -mtime -7 | xargs tar -rf weekly.tar
```

Επεξήγηση:

Η εντολή `find` θα εντοπίσει όλα τα συνηθισμένα αρχεία (`-type f`) τα οποία έχω τύχει αλλαγών τις τελευταίες 7 μέρες (`-mtime -7`). Η εντολή `xargs` μετατρέπει το ρεύμα εισόδου (σε αυτή την



περίπτωση το αποτέλεσμα της *find*) σε ορίσματα της γραμμής εντολής. Σε αυτή την περίπτωση η εντολή *tar* τοποθετεί τα αρχεία αυτά σε ένα *archive* χρησιμοποιώντας την παράμετρο *f* για να δείξουμε ότι θα ακολουθεί το όνομα του αρχείου και την παράμετρο *r* έτσι ώστε να προσθέσει (*append*) τα αρχεία στο υφιστάμενο *weekly.tar*. Αν θέλουμε να δημιουργήσουμε το *tar* αρχείο πρέπει αντί του *r* να χρησιμοποιήσουμε το *c*.

Επίσης το ερώτημα 6 απαντάται και με την εντολή:

```
xargs tar -cf weekly.tar `find . -maxdepth 1 -type f -mtime -7`
```

7) Σβήστε από το σύστημα όλα τα αρχεία των οποίων το όνομα ξεκινά από τη λέξη *core*. Τα αρχεία αυτά μπορούν να υπάρχουν σε διάφορους καταλόγους του συστήματος.

```
find / -name "core*" | xargs /bin/rm -f  
find / -name "core*" -exec /bin/rm -f '{}' \
```