

The Unix Shell

Μέρος 2ο :

Εντολές : find, sort, wc, tar, gzip, man.

Καθιερωμένη είσοδος/έξοδος και επανακαθορισμός τους, συνδέσεις προγραμμάτων (σωλήνες). Οι εντολες grep, awk, sed.

Version Control System : git.

Η εντολή find

Σύνταξη :

```
find <dir> -name <name>
```

```
find <dir> -mtime <time>
```

```
find <dir> -name <name> -exec <command> \;
```

Με την πρώτη μορφή βρίσκουμε τη θέση (διαδρομή) των αρχείων με το όνομα `name`. Η έρευνα ξεκινά από τον κατάλογο `dir`. Με τη δεύτερη μορφή εντοπίζουμε όσα αρχεία τροποποιήθηκαν πριν από `<time>` ημέρες. Με την τελευταία, για κάθε ένα από τα αρχεία με το όνομα `name` που εντοπίζονται εκτελείται η εντολή `command`.

Η εντολή find

Παραδείγματα

```
find ./ -name 'old_file.txt'
```

```
find /usr/local/ -name '*[0-2]*.dat'
```

```
find ./ -mtime -2
```

```
find ./ -mtime +2
```

```
find ./ -mtime 2
```

```
find /usr/lib/ -name '*.2del' -exec rm -rf {} \;
```

```
find ./ -name '*.PREV' -exec wc {} \;
```

Η εντολή `sort`

Σύνταξη : `sort <files>`
 `sort -k <field> <files>`
 `sort -n -k <field> <files>`

Στην πρώτη μορφή τα περιεχόμενα των οριζόμενων αρχείων μπαίνουν σε αλφαβητική σειρά. Στη δεύτερη μορφή, για την ταξινόμηση χρησιμοποιείται η στήλη (το πεδίο) που ορίζεται από το `field`. Στην τρίτη μορφή η ταξινόμηση γίνεται αριθμητικά (και όχι αλφαβητικά).

Η εντολή wc

[w]ord [c]ount

Σύνταξη : wc <files>

Η wc τυπώνει τον αριθμό γραμμών, λέξεων και χαρακτήρων που περιέχονται σε κάθε ένα από τα οριζόμενα αρχεία.

Η εντολή tar

[t]ape [ar]chive

Σύνταξη : tar cvf <file.tar> <files>
 tar tvf <file.tar>
 tar xvf <file.tar>

Στην πρώτη μορφή δημιουργείται ένα αρχείο με το όνομα file.tar το οποίο περιέχει όλα τα οριζόμενα αρχεία ή καταλόγους (και τα περιεχόμενα τους). Στη δεύτερη μορφή τυπώνονται τα ονόματα των αρχείων και καταλόγων που περιέχονται στο file.tar. Στη τρίτη μορφή τα περιεχόμενα του file.tar ανακτώνται από το αρχείο file.tar.

Η εντολή tar

Παραδείγματα

[c]reate :

```
tar cvf test.tar mydir/
```

lis[t] :

```
tar tvf test.tar
```

e[x]tract :

```
tar xvf test.tar
```

Η εντολές gzip & gunzip

Συμπίεση και αποσυμπίεση αρχείων

Σύνταξη : gzip <files>
 gzip -r <dirs>
 gunzip <files>
 gunzip -r <dirs>

Οι εντολές gzip (για [G]nu-zip) και gunzip (για [G]nu-unzip) συμπιέζουν και αποσυμπιέζουν αρχεία.

Η εντολή man

[man]ua1 page

Σύνταξη : man <εντολή>
 man -k <λέξη-κλειδί>
 apropos <λέξη-κλειδί>
 man -K <λέξη-κλειδί>

Στην πρώτη μορφή τυπώνονται οι οδηγίες χρήσης της εντολής (ή θέματος) που ορίζεται (εάν υπάρχει). Στη δεύτερη μορφή τυπώνονται τα ονόματα των σελίδων βοήθειας ή σύντομη περιγραφή των οποίων περιέχει τη λέξη-κλειδί. Στην τρίτη μορφή ερευνώνται όλες οι σελίδες βοήθειας για την ύπαρξη της λέξης-κλειδί.

Καθιερωμένη είσοδος/έξοδος

- Ένα πρόγραμμα σε C μπορεί να διαβάσει δεδομένα είτε από κάποιο αρχείο ή από την 'καθιερωμένη είσοδο' (standard input) η οποία (απουσία επανακαθορισμού) αντιστοιχεί στο πληκτρολόγιο.
- Αντίστοιχα, ένα πρόγραμμα μπορεί να εξάγει τα αποτελέσματα του είτε σε ένα αρχείο ή στην 'καθιερωμένη έξοδο' (standard output) η οποία (απουσία επανακαθορισμού) αντιστοιχεί στο τερματικό (οθόνη).
- Για την έξοδο των διαγνωστικών μηνυμάτων (π.χ. σφαλμάτων) υπάρχει η 'καθιερωμένη έξοδος σφαλμάτων' (standard error) η οποία (απουσία επανακαθορισμού) επίσης αντιστοιχεί στο τερματικό.

Επανακαθορισμός εισόδου/εξόδου

Το unix παρέχει τη δυνατότητα επανακαθορισμού των καθιερωμένων εισόδων/εξόδων με τη χρήση των συμβόλων '>' και '<':

- Το σύμβολο '>' επανακαθορίζει την καθιερωμένη έξοδο.
- Το σύμβολο '<' επανακαθορίζει την καθιερωμένη είσοδο.

Επανακαθορισμός εισόδου/εξόδου

Παραδείγματα

```
find ./ -name '*.dat' > files.list
```

```
ls -R /usr/local > ~/list.dat
```

```
wc < test.txt
```

Συνδέσεις προγραμμάτων

Εάν θέλουμε να εκτελέσουμε διαδοχικά δύο προγράμματα έτσι ώστε η έξοδος από το πρώτο να χρησιμοποιηθεί ως είσοδος για το δεύτερο, μπορούμε να χρησιμοποιήσουμε το :

```
prog1 > temp
```

```
prog2 < temp
```

Το unix παρέχει έναν αυτόματο τρόπο για να γίνει αυτό, τους σωλήνες (pipes στα αγγλικά) :

```
prog1 | prog2
```

όπου η καθιερωμένη έξοδος του prog1 γίνεται (αυτόματα) η καθιερωμένη είσοδος του prog2.

Συνδέσεις προγραμμάτων

Παραδείγματα

```
ls | wc
```

```
grep 'test' * | wc
```

```
tar tvf myfiles.tar | grep 'thisfile.dat'
```

Η εντολή grep

Σύνταξη : `grep [options] <expression> <files>`

Η εντολή `grep` ερευνά τα οριζόμενα αρχεία για την παρουσία της έκφρασης `expression`. Κάθε φορά που η έκφραση βρίσκεται, η `grep` τυπώνει την γραμμή του αντίστοιχου αρχείου που περιέχει την έκφραση.

Η εντολή grep

Παραδείγματα

```
grep 'October' my_file.txt
```

```
grep 'Escherichia' *.dat
```

```
grep '[Tt]est' /usr/local/lib/*
```

```
grep -R 'test' ./teach/
```


Η εντολή grep

Παραδείγματα & regular expressions

```
grep -v 'October' my_file.txt      # invert
```

```
grep -i 'October' my_file.txt      # ignore case
```

```
grep -P 'at..t' DNA.seq            # find ATxxT
```

Η εντολή grep

Παραδείγματα & regular expressions

```
grep -o -P 'at..t' DNA.seq      # print only matching
```

```
grep -P '(at..t)|(tt..t)' DNA.seq # find ATxxT or TTxxT
```

```
grep '^ATOM' 1tii.pdb          # lines starting with ATOM
```

Η εντολή awk

Η awk είναι μια γλώσσα που έχει σχεδιαστεί για επεξεργασία κειμένου και χρησιμοποιείται συνήθως ως εργαλείο εξαγωγής δεδομένων. Η awk είναι μία πλήρης και πολύπλοκη γλώσσα προγραμματισμού. Εδώ θα δωθούν λίγα απλά παραδείγματα για συχνές χρήσεις της γλώσσας.

Η εντολή awk

Παραδείγματα

```
awk '{print $7,$8,$9}' file.txt
```

```
awk '{if ($1<35.0) print $1,$2}' numerical.dat
```

```
awk '{if ($1>=2.0 && $1<2.2) print $1}' numerical.dat
```

```
grep '^ATOM' 1tii.pdb | awk '{print $7,$8,$9}'
```

Η εντολή sed

[s]tream [ed]itor

Ο sed είναι ένας επεξεργαστής ροής (stream editor). Ένας επεξεργαστής ροής χρησιμοποιείται για την εκτέλεση βασικών μετασχηματισμών/αλλαγών στο κείμενο που δίδεται στο πρόγραμμα (το κείμενο δίδεται είτε από ένα αρχείο είτε μέσω ενός σωλήνα). Ενώ είναι παρόμοιος με ένα πρόγραμμα επεξεργασίας κειμένου, ο sed λειτουργεί κάνοντας μόνο ένα πέρασμα πάνω από την είσοδο και, κατά συνέπεια, είναι πιο αποτελεσματικό.

Η εντολή sed

Παραδείγματα

```
sed 's/hello/world/' input.txt > output.txt
```

```
sed 's/hello/world/g' input.txt > output.txt
```

```
sed -i 's/hello/world/g' file.txt
```

```
sed 's/ac..a/ac**a/g' DNA.seq
```

```
sed 's/[cg]g..a/*****/g' DNA.seq
```

Version Control System : git.

Το Git είναι λογισμικό ανοιχτού κώδικα για την παρακολούθηση αλλαγών σε οποιοδήποτε σύνολο αρχείων, που χρησιμοποιείται συνήθως για τον συντονισμό της εργασίας μεταξύ προγραμματιστών που αναπτύσσουν συνεργατικά τον πηγαίο κώδικα κατά την ανάπτυξη λογισμικού. Είναι πολλοί αυτοί που χρησιμοποιούν git σε κάθε τους project ώστε να το κρατούν οργανωμένο, να μπορούν να παρακολουθούν τις αλλαγές, και να μπορούν να επιστρέψουν με ασφάλεια σε προηγούμενες εκδόσεις των αρχείων τους.

Version Control System : git.

Κύκλος εργασίας για standalone git

```
git config --global user.email "my@emailaddress.com"
```

```
git config --global user.name "My name"
```

```
cd <directory>
```

```
git init
```

```
git add <file or directory name>
```

```
git commit -m "Commit message in quotes"
```


Version Control System : git.

Κύκλος εργασίας για standalone git

```
git status
```

```
git reset --hard <hash>
```