

# Εισαγωγή στους Ηλεκτρονικούς Υπολογιστές Σχολή Θετικών Επιστημών – Τμ. Πληροφορικής



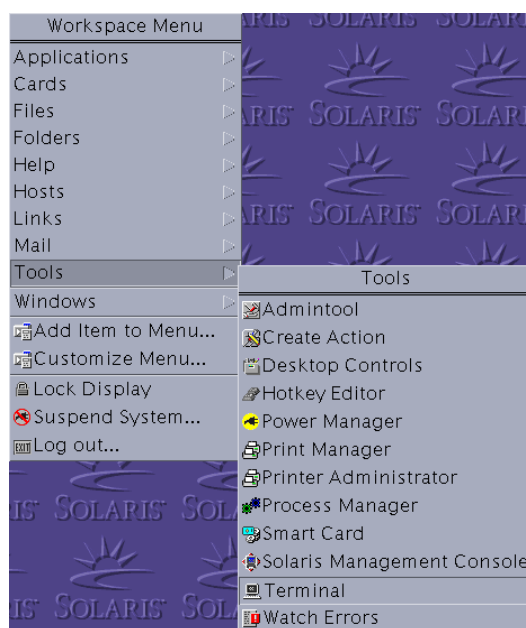
Διδάσκων: Ι. Η. Λαγαρής

## Εισαγωγή

### 1.0 *Solaris και UNIX*

Το UNIX είναι ένα λειτουργικό σύστημα που αναπτύχθηκε τις δεκαετίες του 60 και 70 από μία ομάδα ανθρώπων της εταιρείας AT&T, μεταξύ των οποίων οι Ken Thompson, Dennis Ritchie, και Douglas McIlroy στα εργαστήρια Bell. Έκτοτε πολλές εκδόσεις του εν λόγω λειτουργικού έγιναν. Μία από αυτές ήταν του συστήματος SunOS, τη δεκαετία του 80, από την εταιρεία SUN για να υποστηρίξει τη λειτουργία των σταθμών εργασίας που κατασκεύαζε. Αργότερα το λειτουργικό αυτό μετονομάστηκε σε SOLARIS. Στην ουσία η ονομασία SOLARIS υποδηλώνει την χρήση του λειτουργικού συστήματος SunOS και ενός γραφικού περιβάλλοντος για επικοινωνία με το χρήστη. Στον παρόντα οδηγό θα ασχοληθούμε μόνο με την περιγραφή εντολών για το SunOS.

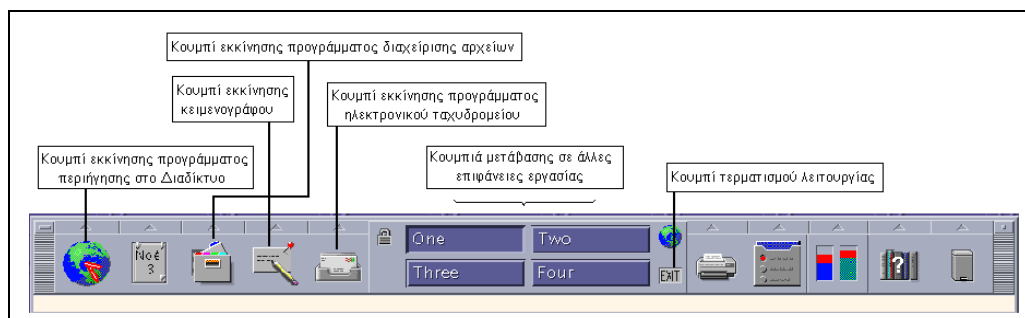
Στην ενότητα αυτή απλά θα δούμε με ποιόν τρόπο μπορούμε να δίνουμε εντολές στη γραμμή εντολών, ενώ στις επόμενες ενότητες θα δούμε ορισμένες βασικές εντολές του UNIX. Τα βήματα που χρειάζονται για να ανοίξουμε ένα τερματικό, ένα παράθυρο δηλαδή με γραμμή εντολών, δίνονται στην εικόνα που ακολουθεί.



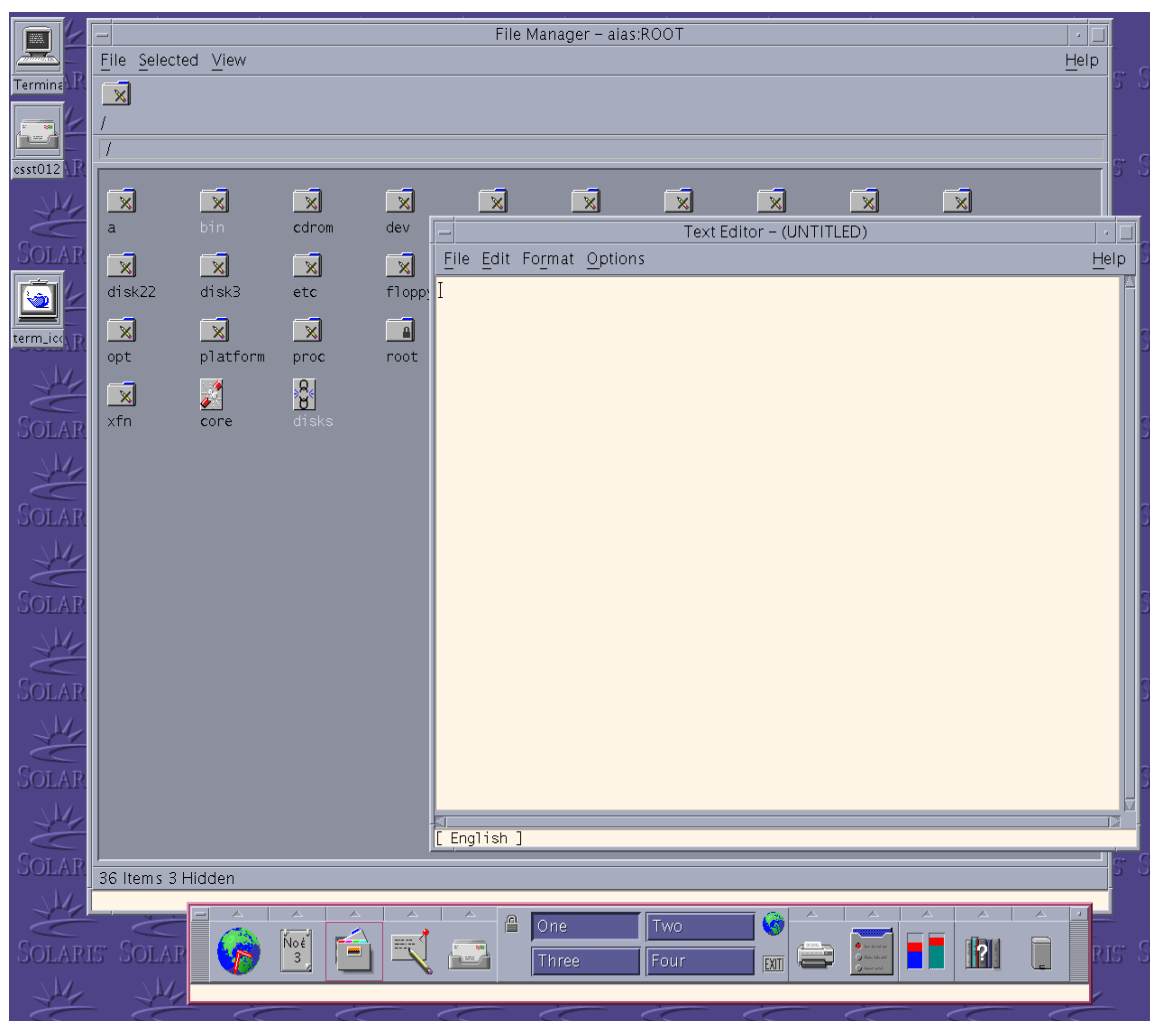
Εικόνα 1: με δεξί κλικ στην επιφάνεια εργασίας εμφανίζεται το παραπάνω πλαίσιο επιλογών.

## 1.1 Το γραφικό περιβάλλον CDE

Όπως είπαμε και παραπάνω, το SOLARIS διαθέτει ένα γραφικό περιβάλλον για επικοινωνία με τον χρήστη. Θα δώσουμε μία μικρή εισαγωγική περιγραφή του εν λόγω συστήματος, προκειμένου να υπάρξει μία πρώτη επαφή και εξοικείωση με αυτό. Το βασικό γραφικό περιβάλλον είναι το Common Desktop Environment – CDE. Η επιφάνεια εργασίας, μέσα στην οποία εμφανίζονται όλα τα παράθυρα, όπως και μία περιγραφή της κεντρικής μπάρας εργασιών δίνονται στις παρακάτω εικόνες.

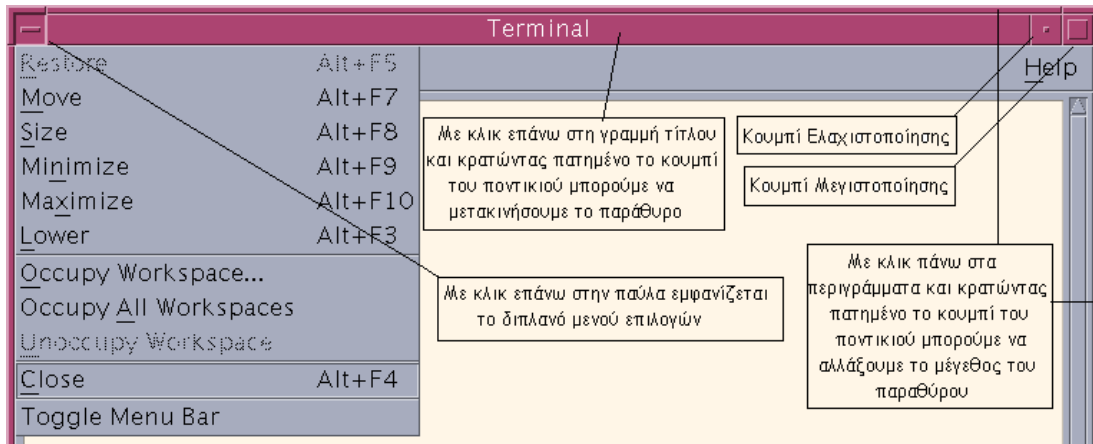


Εικόνα 2: κεντρική μπάρα εργασιών



Εικόνα 3: Επιφάνεια Εργασίας στο CDE.

Ας δούμε τώρα με ποιον τρόπο μπορούμε να αξιοποιήσουμε τα παράθυρα του CDE. Όπως κάθε γραφικό περιβάλλον διαχείρισης, το CDE επιτρέπει την ελαχιστοποίηση, μεγιστοποίηση, μετακίνηση, αλλαγή μεγέθους και κλείσιμο παραθύρου. Στην παρακάτω εικόνα περιγράφονται οι εν λόγω ενέργειες.



Εικόνα 4: βασικές παραθυρικές λειτουργίες.



Εικόνα 5: αποτέλεσμα της ελαχιστοποίησης – με κλικ πάνω στο εικονίδιο επαναφέρουμε το παράθυρο στην αρχική κατάσταση

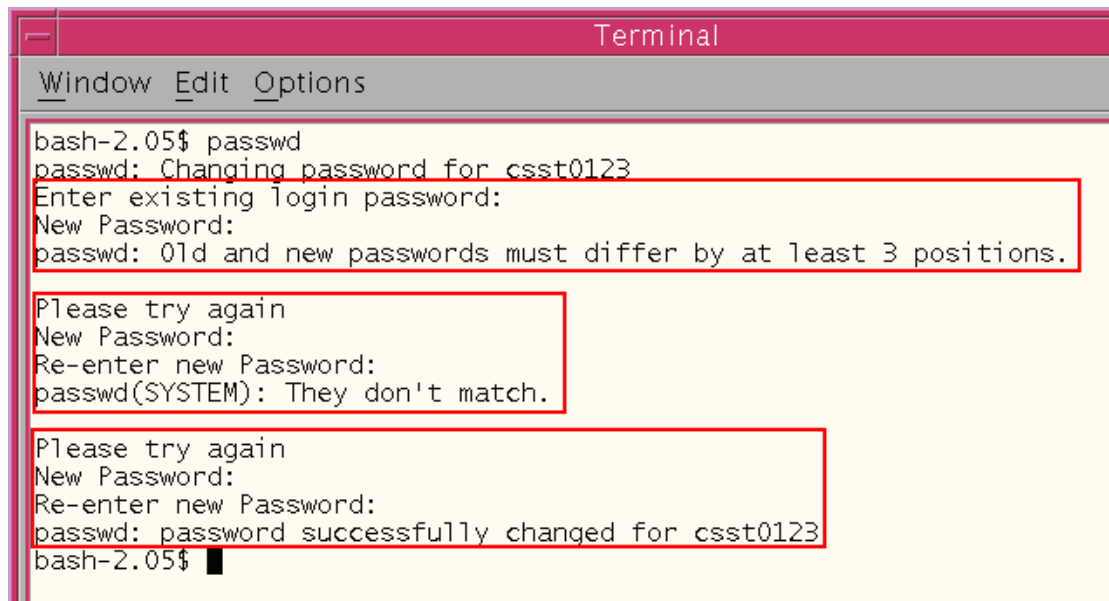


Εικόνα 6: λειτουργίες πάνω στο ελαχιστοποιημένο παράθυρο.

# Σύντομος Οδηγός του UNIX

## 2.0 Αλλαγή συνθηματικού χρήστη - *passwd*

Η εντολή `passwd` μας επιτρέπει να αλλάζουμε τον κωδικό πρόσβασης στο λειτουργικό. Ο κωδικός πρόσβασης είναι ένα προσωπικό στοιχείο και καλό θα ήταν να μην το εμπιστευόμαστε σε τρίτους. Στην εικόνα που ακολουθεί περιγράφεται ο τρόπος αλλαγής κωδικού και ορισμένα πιθανά προβλήματα που μπορεί να εμφανιστούν.



```
Terminal
Window Edit Options
bash-2.05$ passwd
passwd: Changing password for csst0123
Enter existing login password:
New Password:
passwd: Old and new passwords must differ by at least 3 positions.

Please try again
New Password:
Re-enter new Password:
passwd(SYSTEM): They don't match.

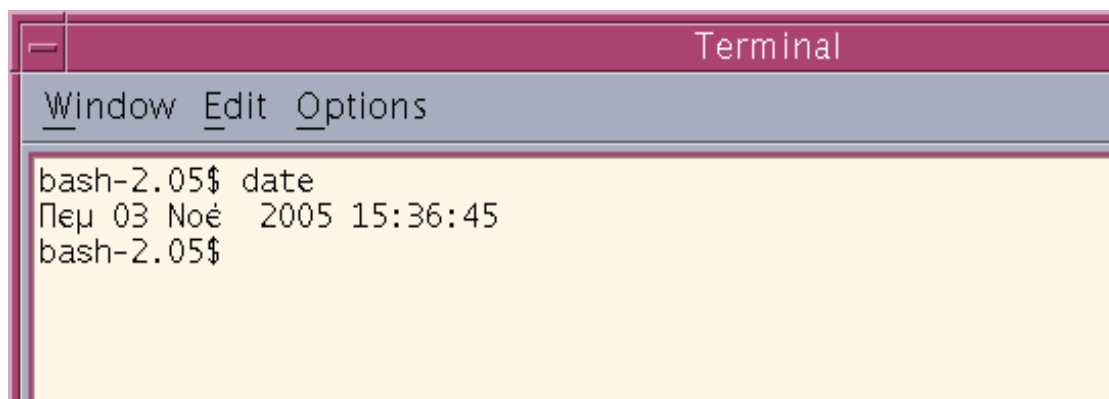
Please try again
New Password:
Re-enter new Password:
passwd: password successfully changed for csst0123
bash-2.05$ █
```

Εικόνα 7: επίδειξη χρήσης `passwd`

## 2.1 Εποπτεία του χρόνου – *date, cal, time*

Η εντολή `date`

Η εντολή `date` απλά μας πληροφορεί για την τρέχουσα ώρα και ημερομηνία του υπολογιστή στον οποίο δουλεύουμε. Να σημειώσουμε ότι η ώρα αυτή είναι η τοπική ώρα του μηχανήματος.



```
Terminal
Window Edit Options
bash-2.05$ date
Πεμ 03 Νοέ 2005 15:36:45
bash-2.05$
```

Εικόνα 8: επίδειξη χρήσης `date`

## Η εντολή cal

Η εντολή cal δίνει ημερολογιακές πληροφορίες. Ειδικότερα η πληκτρολόγηση cal επιστρέφει το ημερολόγιο του τρέχοντος μήνα, cal <αριθμός> επιστρέφει το ημερολόγιο για το έτος <αριθμός>, ενώ cal <μήνας> <έτος> επιστρέφει το ημερολόγιο για τον μήνα <μήνα> του έτους <έτος>. Στις εικόνες που ακολουθούν δίνονται μερικά παραδείγματα της cal.

```
Terminal
Window Edit Options
bash-2.05$ cal
  Νοέμβριος 2005
  S  M Tu  W Th  F  S
      1  2  3  4  5
  6  7  8  9 10 11 12
 13 14 15 16 17 18 19
 20 21 22 23 24 25 26
 27 28 29 30
bash-2.05$ █
```

Εικόνα 9: επίδειξη χρήσης cal

```
Terminal
Window Edit Options
  Ιαν          Φεβ          Μάρ
  S  M Tu  W Th  F  S   S  M Tu  W Th  F  S   S  M Tu  W Th  F  S
      1  2  3  4  5       1  2  3  4  5       1  2  3  4  5
  2  3  4  5  6  7  8       6  7  8  9 10 11 12       6  7  8  9 10 11 12
  9 10 11 12 13 14 15       13 14 15 16 17 18 19       13 14 15 16 17 18 19
 16 17 18 19 20 21 22       20 21 22 23 24 25 26       20 21 22 23 24 25 26
 23 24 25 26 27 28 29       27 28                       27 28 29 30 31
 30 31
  Απρ          Μάι          Ιού
  S  M Tu  W Th  F  S   S  M Tu  W Th  F  S   S  M Tu  W Th  F  S
      1  2       1  2  3  4  5  6  7       1  2  3  4
  3  4  5  6  7  8  9       8  9 10 11 12 13 14       5  6  7  8  9 10 11
 10 11 12 13 14 15 16       15 16 17 18 19 20 21       12 13 14 15 16 17 18
 17 18 19 20 21 22 23       22 23 24 25 26 27 28       19 20 21 22 23 24 25
 24 25 26 27 28 29 30       29 30 31                       26 27 28 29 30
  Ιού          Αύγ          Σεπ
  S  M Tu  W Th  F  S   S  M Tu  W Th  F  S   S  M Tu  W Th  F  S
      1  2       1  2  3  4  5  6       1  2  3
  3  4  5  6  7  8  9       7  8  9 10 11 12 13       4  5  6  7  8  9 10
 10 11 12 13 14 15 16       14 15 16 17 18 19 20       11 12 13 14 15 16 17
 17 18 19 20 21 22 23       21 22 23 24 25 26 27       18 19 20 21 22 23 24
 24 25 26 27 28 29 30       28 29 30 31                       25 26 27 28 29 30
 31
  Οκτ          Νοέ          Δεκ
  S  M Tu  W Th  F  S   S  M Tu  W Th  F  S   S  M Tu  W Th  F  S
      1  2       1  2  3  4  5       1  2  3
  2  3  4  5  6  7  8       6  7  8  9 10 11 12       4  5  6  7  8  9 10
  9 10 11 12 13 14 15       13 14 15 16 17 18 19       11 12 13 14 15 16 17
 16 17 18 19 20 21 22       20 21 22 23 24 25 26       18 19 20 21 22 23 24
 23 24 25 26 27 28 29       27 28 29 30                       25 26 27 28 29 30 31
 30 31
```

Εικόνα 10: επίδειξη χρήσης cal 2005

```
Terminal
Window Edit Options
bash-2.05$ cal 11 2004
    Νοέμβριος 2004
  S  M Tu  W Th  F  S
    1  2  3  4  5  6
  7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30
bash-2.05$ █
```

Εικόνα 11: επίδειξη χρήσης cal 11 2004

### Η εντολή time

Με την εντολή time μπορούμε να μετρήσουμε, σε seconds, τον χρόνο εκτέλεσης μίας εντολής, για την ακρίβεια της εντολής που δίνουμε ως όρισμα στην time. Να σημειώσουμε ότι η χρονομετρούμενη εντολή δίνεται όπως ακριβώς θα τη γράφαμε στη γραμμή εντολών, δηλαδή η χρήση ορισμάτων είναι επιτρεπτή. Στην εικόνα που ακολουθεί δίνεται ένα παράδειγμα χρονομέτρησης της εντολής ls -R.

```
Terminal
Window Edit Options Help
bash-2.05$ time ls -R
.:
dir1 dir2dir5 dir4 file1.txt
dir2 dir3 dir4dir6

./dir1:

./dir2:

./dir2dir5:

./dir3:
file1.txt

./dir4:

./dir4dir6:

real    0m0.020s
user    0m0.000s
sys     0m0.000s
bash-2.05$ █
```

Εξόδος της εντολής ls -R

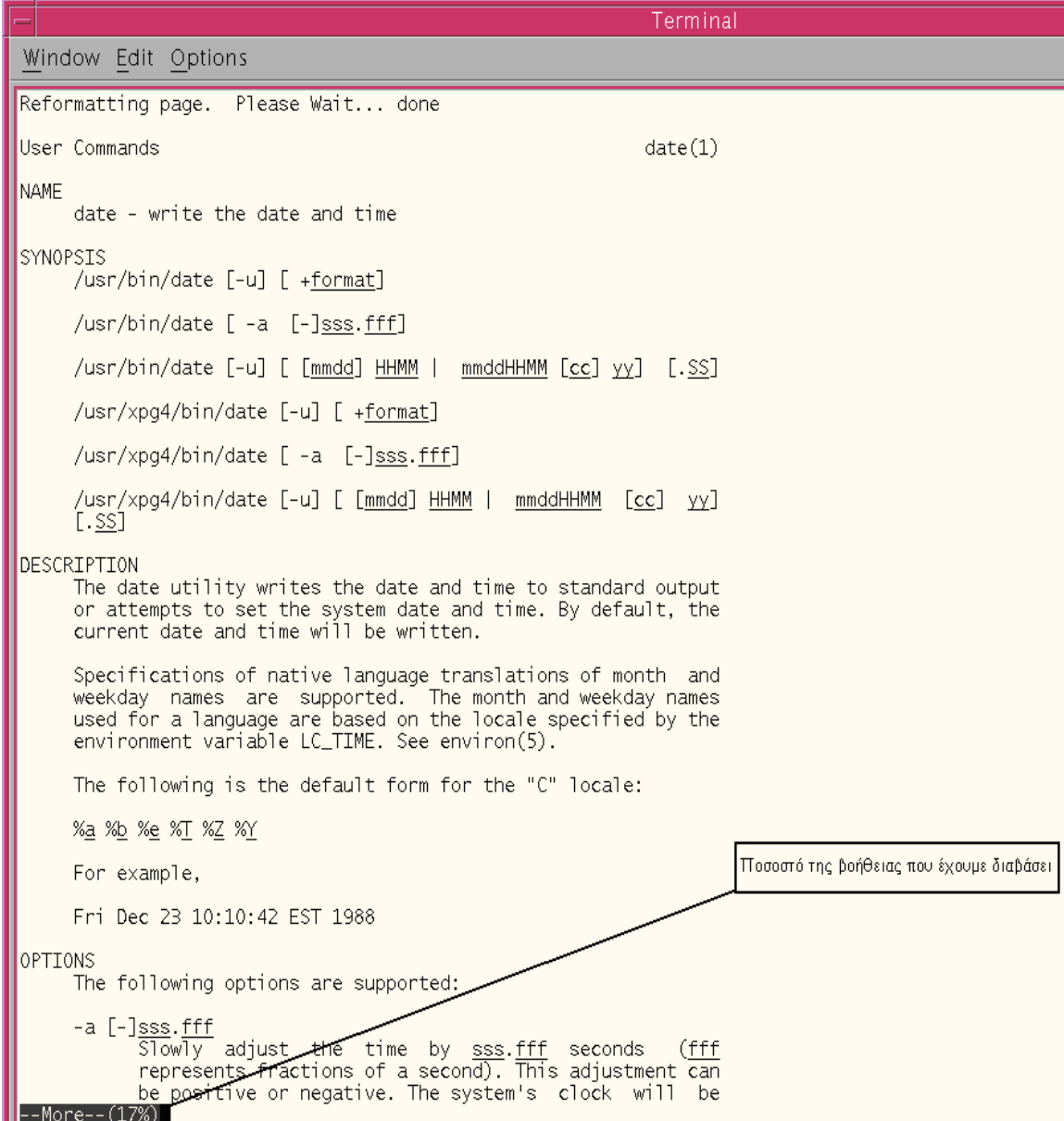
Χρόνοι εκτέλεσης της εντολής. Το πεδίο real αναφέρεται στο πόσα seconds πέρασαν από τη στιγμή που άρχισε η εκτέλεση μέχρι και το πέρας της

Εικόνα 12: επίδειξη χρήσης time ls -R

## 2.2 Τεκμηρίωση εντολών – man

Η εντολή man είναι από τις πιο βασικές στο UNIX, δεδομένου ότι παρέχει βοήθεια για οποιαδήποτε εντολή δοθεί ως όρισμα στη γραμμή εντολών. Στην εικόνα που ακολουθεί φαίνεται το αποτέλεσμα της πληκτρολόγησης man date, δηλαδή η βοήθεια για την εντολή date.

Πολλές φορές η βοήθεια περιλαμβάνει πολλές σελίδες. Η man επιστρέφει το αποτέλεσμα τμηματικά. Στο κάτω μέρος της οθόνης εμφανίζεται το ποσοστό της βοήθειας που έχει εμφανιστεί ως τώρα. Για μετάβαση στην επόμενη γραμμή πατάμε το πλήκτρο enter, και για μετάβαση στην επόμενη σελίδα το space. Τέλος για να τερματίσουμε την προβολή βοήθειας και να επιστρέψουμε στη γραμμή εντολών πατάμε το πλήκτρο q.



```
Terminal
Window Edit Options
Reformatting page. Please Wait... done
User Commands                                date(1)
NAME
date - write the date and time
SYNOPSIS
/usr/bin/date [-u] [ +format]
/usr/bin/date [ -a [-]sss.fff]
/usr/bin/date [-u] [ [mdd] HHMM | mddHHMM [cc] yy] [.SS]
/usr/xpg4/bin/date [-u] [ +format]
/usr/xpg4/bin/date [ -a [-]sss.fff]
/usr/xpg4/bin/date [-u] [ [mdd] HHMM | mddHHMM [cc] yy]
[.SS]
DESCRIPTION
The date utility writes the date and time to standard output
or attempts to set the system date and time. By default, the
current date and time will be written.

Specifications of native language translations of month and
weekday names are supported. The month and weekday names
used for a language are based on the locale specified by the
environment variable LC_TIME. See environ(5).

The following is the default form for the "C" locale:

%a %b %e %T %Z %Y

For example,
Fri Dec 23 10:10:42 EST 1988

OPTIONS
The following options are supported:

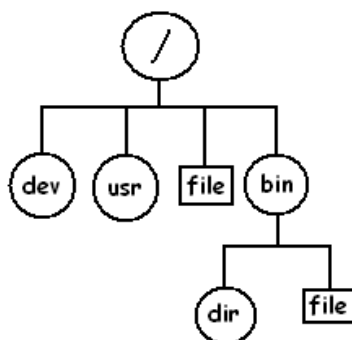
-a [-]sss.fff
    Slowly adjust the time by sss.fff seconds (fff
    represents fractions of a second). This adjustment can
    be positive or negative. The system's clock will be
--More-- (17%)
```

Εικόνα 13: επίδειξη χρήσης man date

## 2.3 Διαχείριση αρχείων & καταλόγων – *pwd, ls, mkdir, rmdir, cp, mv, rm, ln*

### Οργάνωση αρχείων στο UNIX

Το Unix, όπως και όλα τα λειτουργικά συστήματα, ακολουθεί μία δενδρική δομή στην οργάνωση των αρχείων. Δηλαδή υπάρχει ένας βασικός κατάλογος (που ονομάζεται ριζικός – root directory) και μέσα στον οποίο ανήκουν άλλοι υποκατάλογοι ή/και αρχεία. Κάθε υποκατάλογος μπορεί να διαθέτει άλλους υποκαταλόγους ή/και αρχεία. Ένα παράδειγμα τέτοιας οργάνωσης είναι αυτό που φαίνεται στην παρακάτω εικόνα, όπου μέσα με κύκλο παριστάνονται οι κατάλογοι και με ορθογώνιο τα αρχεία. Η δομή αυτή μπορεί να εκτείνεται και περισσότερο, με την έννοια ότι ένα υποκατάλογος μπορεί να έχει και άλλους υποκαταλόγους και αυτοί άλλους κ.ο.κ.

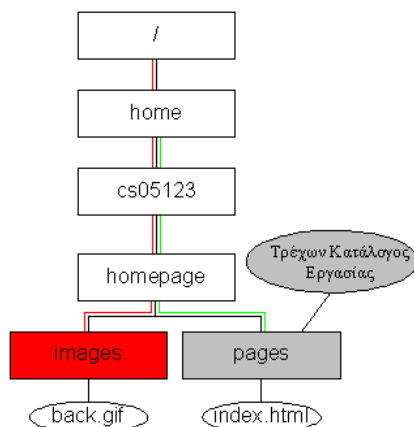


Εικόνα 14: Παράδειγμα ενός δέντρου συστήματος αρχείων.

Η διαχείριση αρχείων είναι μία πολύ σημαντική λειτουργία για ένα λειτουργικό σύστημα. Για το λόγο αυτό υπάρχουν αρκετές εντολές που σχετίζονται με αυτή τη διαδικασία. Στην ενότητα αυτή θα δούμε πώς μπορούμε να δημιουργούμε καταλόγους, να τους διαγράφουμε και να βλέπουμε τα περιεχόμενά τους. Τέλος θα ασχοληθούμε με ορισμένες εντολές που μας βοηθάνε στη διάσχιση του δέντρου που περιγράψαμε, δηλαδή πώς να μαθαίνουμε το που βρισκόμαστε και να μεταβαίνουμε από τον ένα κατάλογο σε στον άλλο.

### Απόλυτη και Σχετική διευθυνσιοδότηση αρχείων

Όπως προαναφέραμε, το σύστημα αρχείων είναι ένα δέντρο, οργανωμένο σε αρχεία και καταλόγους. Κάθε αρχείο (ή κατάλογος) μπορεί να οριστεί μοναδικά και πλήρως από τη διαδρομή στην οποία βρίσκεται. Υπάρχουν δύο τρόποι να οριστεί μια διαδρομή ενός αρχείου (ή καταλόγου): ο απόλυτος και ο σχετικός. Θα προσπαθήσουμε να ορίσουμε τις δύο έννοιες μέσα από ένα παράδειγμα. Για το λόγο αυτό θα πρέπει να ορίσουμε ένα δέντρο, αντίστοιχο με αυτό που είδαμε στην παραπάνω ενότητα, και το οποίο δίνεται στη συνέχεια.



Εικόνα 15: Ένα δέντρο που παριστάνει τη δομή ενός συστήματος αρχείων



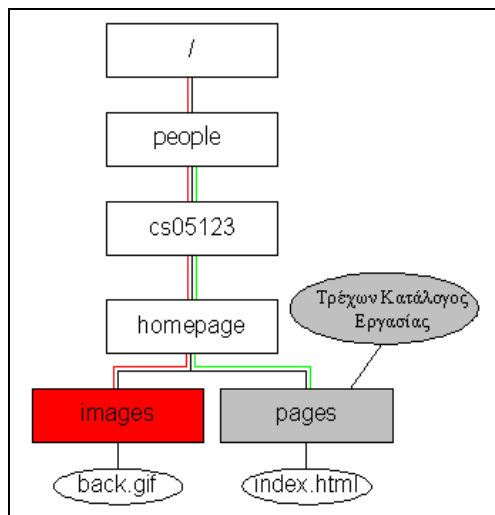
Στο δένδρο της εικόνας 15, μέσα με ορθογώνια απεικονίζουμε τους καταλόγους και με κύκλο τα αρχεία. Ο κατάλογος pages, που είναι σκιαγραφημένος, υποδηλώνεται ως «Τρέχοντας Κατάλογος Εργασίας», δηλαδή η πληκτρολόγηση της εντολής `pwd` θα επέστρεφε `/home/cs05123/homepage/pages/`.

Ας υποθέσουμε τώρα ότι θέλουμε να μεταβούμε στον κατάλογο images που βρίσκεται και αυτός μέσα στον κατάλογο homepage. Για το λόγο αυτό θα χρησιμοποιήσουμε την εντολή `cd` (για την περιγραφή της ανατρέξατε στο προηγούμενο εργαστήριο). Η `cd` παίρνει σαν όρισμα ένα όνομα καταλόγου και τον θέτει ως τρέχοντα κατάλογο εργασίας. Έτσι λοιπόν για να κάνουμε τη μετάβαση που ζητάμε, θα γράφαμε `cd /home/cs05123/homepage/images`, δηλαδή να ακολουθήσουμε την κόκκινη διαδρομή στο παραπάνω δέντρο. Ο τρόπος αυτός παράστασης διαδρομών ονομάζεται απόλυτη διευθυνσιοδότηση, γιατί για να περιγράψουμε ένα αρχείο ξεκινάμε από την κορυφή, δηλαδή τη ρίζα και ακολουθούμε βήμα-βήμα τη διαδρομή στο δέντρο μέχρι να φτάσουμε το αρχείο ή τον κατάλογο που ζητάμε.

Ένας άλλος τρόπος για να προσπελάσουμε αρχεία βασίζεται στην παρατήρηση του επιπέδου στο οποίο βρίσκεται το αρχείο, στο παράδειγμά μας, οι κατάλογοι images, pages βρίσκονται στο ίδιο επίπεδο, αφού και οι δύο έχουν τοποθετηθεί μέσα στον κατάλογο homepage. Αντίστοιχα, ο κατάλογος homepage βρίσκεται ένα επίπεδο παραπάνω, κ.ο.κ. Έτσι λοιπόν για να μεταβούμε από τον κατάλογο pages στον images, θα ακολουθήσουμε την εξής πορεία: Θα μεταβούμε ένα επίπεδο επάνω (δηλαδή στον homepage) και μετά στον pages. Η έννοια ένα επίπεδο πάνω στο Unix παριστάνεται με δύο συνεχόμενες (δηλαδή χωρίς ενδιάμεσο κενό) τελείες, «..». Έτσι η ζητούμενη μετάβαση θα μπορούσε να γίνει ως `cd ../images`. Κάθε δυάδα τελειών σημαίνει και ένα επίπεδο παραπάνω στο δέντρο αρχείων. Έτσι στο παράδειγμά μας η πληκτρολόγηση `cd ../../` θα μας μετέφερε στον κατάλογο home, αφού από τον κατάλογο pages μεταβαίνουμε τρία επίπεδα παραπάνω (δηλαδή ακολουθούμε την πράσινη γραμμή). Αντίστοιχα, υπάρχει και συμβολισμός για το τρέχον επίπεδο, η μία τελεία. Έτσι στο παράδειγμά μας η διαδρομή `/home/cs05123/homepage/pages/` είναι ισοδύναμη με «..». Ο τρόπος αυτό απεικόνισης διαδρομών με επίπεδα ονομάζεται σχετική διευθυνσιοδότηση, αφού ο προσδιορισμός ενός αρχείου γίνεται έμμεσα, με βάση τα επίπεδα.

Τέλος υπάρχει και μία ακόμα συντομογραφία που ανήκει στην σχετική διευθυνσιοδότηση, κατά κάποιο τρόπο, για τον προσδιορισμό του ριζικού καταλόγου χρήστη. Ο συμβολισμός αυτός είναι ο «~/». Με άλλα λόγια, στο παραπάνω παράδειγμα, αν υποθέσουμε ότι ο ριζικός κατάλογος χρήστη είναι ο `/home/cs05123/`, τότε η συντομογραφία `~/` είναι ισοδύναμη με την προαναφερθείσα διαδρομή. Για να διαπιστώσετε ποιος είναι ο ριζικός κατάλογος χρήστη απλά πληκτρολογήστε `cd` και μετά `pwd`. Τότε το αποτέλεσμα που θα σας επιστρέψει η `pwd` στην οθόνη θα είναι ισοδύναμο με `~/`.

Στο σημείο αυτό να προσέξετε ότι η σχετική διευθυνσιοδότηση είναι πιο γενική από την απόλυτη. Φανταστείτε ότι η δομή του παραπάνω δέντρου, άλλαζε όπως στη συνέχεια.

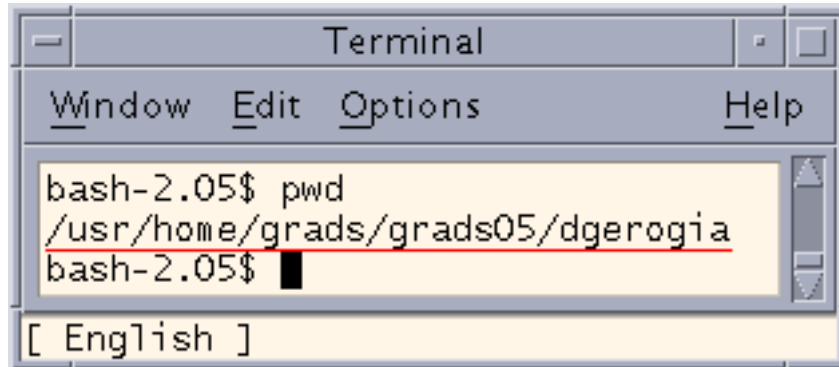


Εικόνα 16: Παραλλαγή του δέντρου της εικόνα 15.

Τότε η εντολή `cd /home/cs05123/homepages/images` θα επέστρεφε σφάλμα, αφού πλέον ο κατάλογος home έχει μετονομαστεί σε people. Αντίθετα η `cd ../images` δε θα είχε κανένα πρόβλημα γιατί η ιεραρχία των καταλόγων έχει διατηρηθεί. Και πάλι οι images, pages βρίσκονται στο ίδιο επίπεδο.

Η εντολή `pwd` (= **p**rint **w**orking **d**irectory)

Η εντολή αυτή τυπώνει στην οθόνη τον τρέχοντα κατάλογο στον οποίο δουλεύουμε.

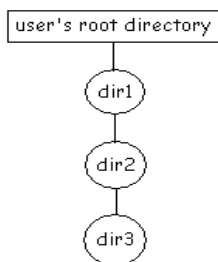


```
Terminal
Window Edit Options Help
bash-2.05$ pwd
/usr/home/grads/grads05/dgerogia
bash-2.05$
[ English ]
```

Εικόνα 17: Το αποτέλεσμα της `pwd`.

Η εντολή `cd` (= **c**hange **d**irectory)

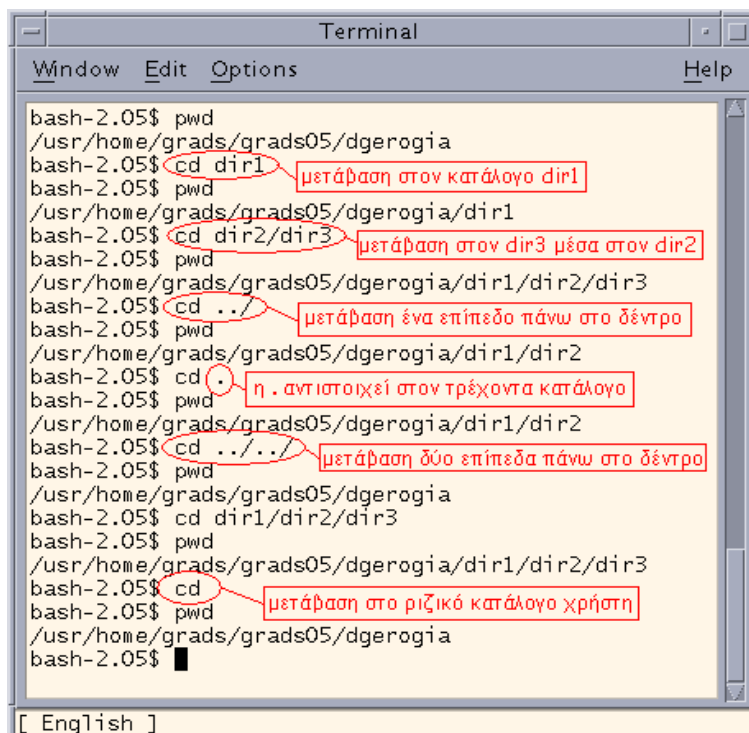
Η εντολή `cd` μας βοηθάει στην πλοήγηση μας μέσα στο σύστημα αρχείων. Ειδικότερα, μπορούμε με την `cd` να μεταβαίνουμε από τον ένα κατάλογο στον άλλο. Στο παράδειγμα που θα δούμε στη συνέχεια, υποθέτουμε την εξής δομή στους καταλόγους μας:



Ο κατάλογος `dir3` βρίσκεται μέσα στον `dir2`, ο οποίος βρίσκεται μέσα στον `dir1`, που τελικά ανήκει στον ριζικό κατάλογο χρήστη.

**ΠΑΡΑΤΗΡΗΣΗ:**

Στο Unix ο λογαριασμός χρήστη συνοδεύεται και με έναν κατάλογο μέσα στο σύστημα αρχείων, ο οποίος ονομάζεται ριζικός κατάλογος χρήστη. Ο κατάλογος αυτός ανήκει αποκλειστικά στον χρήστη, ο οποίος και έχει πλήρη δικαιώματα επί του καταλόγου αυτού και των περιεχομένων του.

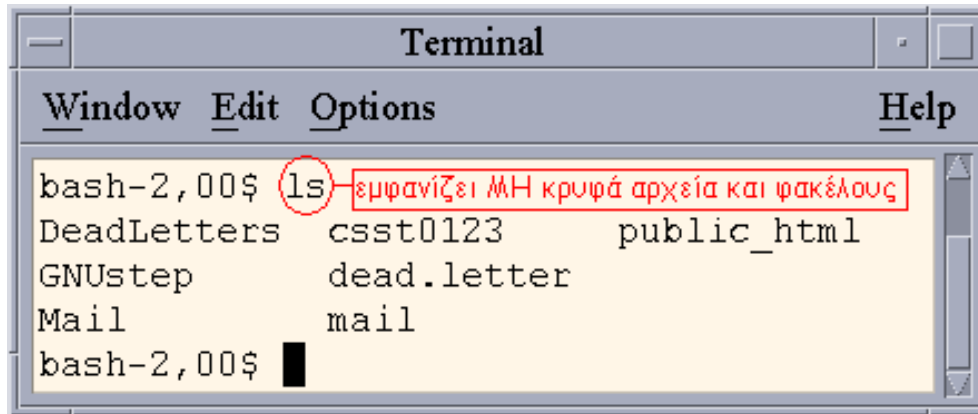


```
Terminal
Window Edit Options Help
bash-2.05$ pwd
/usr/home/grads/grads05/dgerogia
bash-2.05$ cd dir1
bash-2.05$ pwd
/usr/home/grads/grads05/dgerogia/dir1
bash-2.05$ cd dir2/dir3
bash-2.05$ pwd
/usr/home/grads/grads05/dgerogia/dir1/dir2/dir3
bash-2.05$ cd ../
bash-2.05$ pwd
/usr/home/grads/grads05/dgerogia/dir1/dir2
bash-2.05$ cd .
bash-2.05$ pwd
/usr/home/grads/grads05/dgerogia/dir1/dir2
bash-2.05$ cd ../../
bash-2.05$ pwd
/usr/home/grads/grads05/dgerogia
bash-2.05$ cd dir1/dir2/dir3
bash-2.05$ pwd
/usr/home/grads/grads05/dgerogia/dir1/dir2/dir3
bash-2.05$ cd
bash-2.05$ pwd
/usr/home/grads/grads05/dgerogia
bash-2.05$
```

Εικόνα 18: Διάφορες χρήσεις της “`cd`”. Προσέξτε τον τρόπο χρήσης “`..`” και “`.`”

## Η εντολή ls (= list)

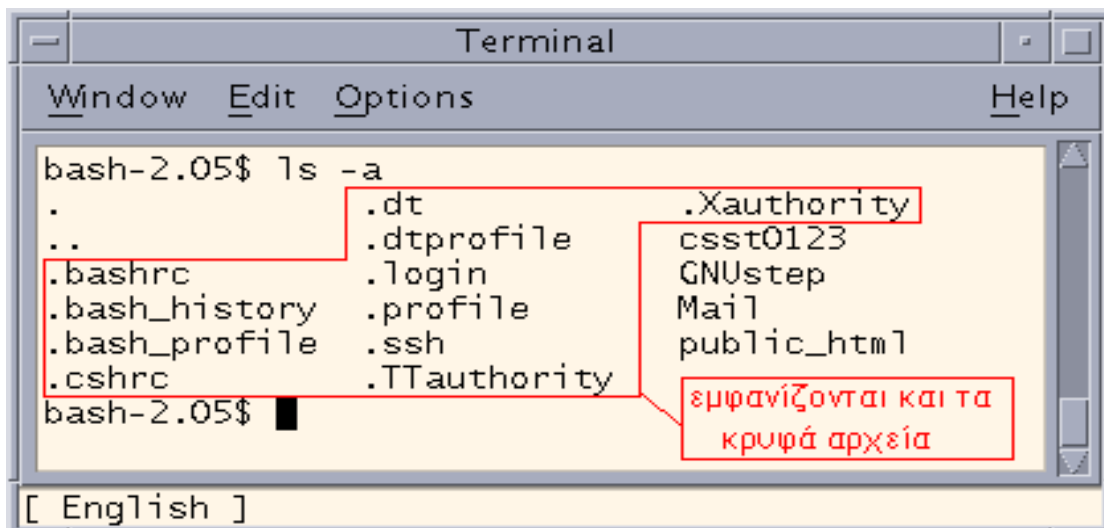
Ας δούμε τώρα πώς μπορούμε να προβάλουμε τα περιεχόμενα ενός καταλόγου στην οθόνη. Η λειτουργία αυτή επιτυγχάνεται στο Unix με την εντολή ls. Στην απλούστερη μορφή της (δηλαδή απλά πληκτρολογώντας ls) το αποτέλεσμα είναι να πάρουμε σε διατεταγμένη μορφή τα περιεχόμενα του τρέχοντος καταλόγου εργασίας (όπως αυτός ορίζεται από την pwd), δηλαδή τα αρχεία και τους υποκαταλόγους που περιέχει. Διάφορες περιπτώσεις χρήσεις της ls δίνονται στις εικόνες που ακολουθούν.



```
Terminal
Window Edit Options Help
bash-2,00$ ls
DeadLetters csst0123 public_html
GNUstep     dead.letter
Mail        mail
bash-2,00$
```

εμφανίζει ΜΗ κρυφά αρχεία και φακέλους

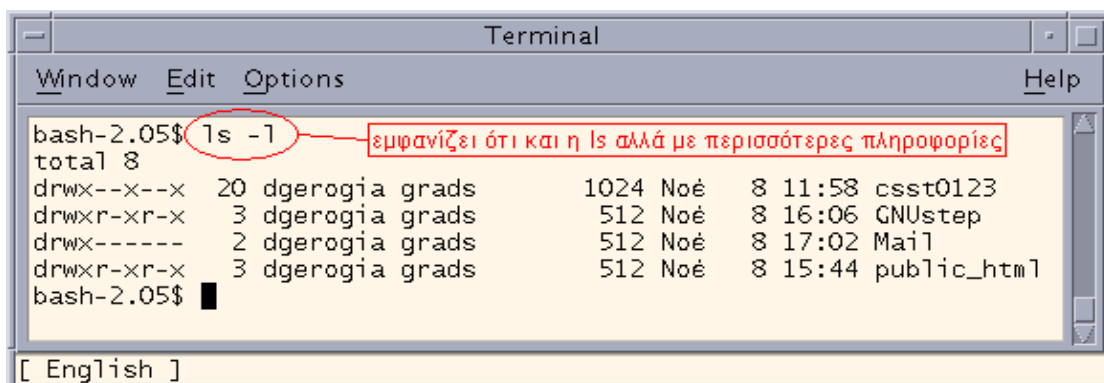
Εικόνα 19: ls χωρίς παραμέτρους



```
Terminal
Window Edit Options Help
bash-2.05$ ls -a
.      .dt      .Xauthority
..     .dtprofile csst0123
.bashrc .login   GNUstep
.bash_history .profile Mail
.bash_profile .ssh     public_html
.cshrc  .TTauthority
bash-2.05$
```

εμφανίζονται και τα κρυφά αρχεία

Εικόνα 20: ls -a για να δούμε και τα κρυφά αρχεία



```
Terminal
Window Edit Options Help
bash-2.05$ ls -l
total 8
drwx--x--x 20 dgerogia grads 1024 Νοέ  8 11:58 csst0123
drwxr-xr-x  3 dgerogia grads  512 Νοέ  8 16:06 GNUstep
drwx----- 2 dgerogia grads  512 Νοέ  8 17:02 Mail
drwxr-xr-x  3 dgerogia grads  512 Νοέ  8 15:44 public_html
bash-2.05$
```

εμφανίζει ότι και η ls αλλά με περισσότερες πληροφορίες

Εικόνα 21: ls -l για να δούμε περισσότερες πληροφορίες για αρχεία και φακέλους

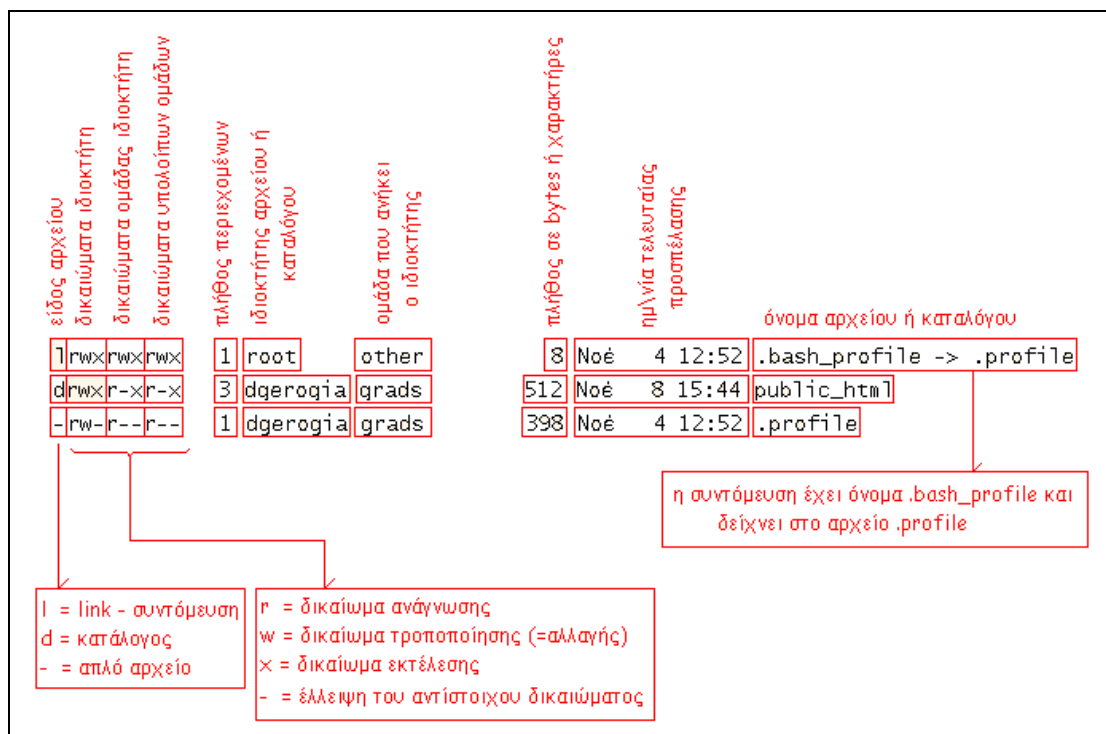
```

Terminal
Window Edit Options Help
bash-2.05$ ls -al
total 34
drwxr-xr-x  8 dgerogia grads    512 Νοέ 10 15:46 .
drwxr-xr-x 26 root    other    512 Νοέ  7 14:01 ..
lrwxrwxrwx  1 root    other      8 Νοέ  4 12:52 .bashrc -> .profile
-rw-----  1 dgerogia grads   456 Νοέ 10 15:57 .bash_history
lrwxrwxrwx  1 root    other      8 Νοέ  4 12:52 .bash_profile -> .profile
-rw-r--r--  1 dgerogia grads   136 Νοέ  4 12:52 .cshrc
drwxr-xr-x 11 dgerogia grads    512 Νοέ 10 15:47 .dt
lrwxrwxrwx  1 root    other      8 Νοέ  4 12:52 .dtprofile -> .profile
-rw-r--r--  1 dgerogia grads   157 Νοέ  4 12:52 .login
-rw-r--r--  1 dgerogia grads   398 Νοέ  4 12:52 .profile
drwx-----  2 dgerogia grads    512 Νοέ 10 15:50 .ssh
-rw-----  1 dgerogia grads   152 Νοέ 10 15:45 .TTauthority
-rw-----  1 dgerogia grads   200 Νοέ 10 15:45 .Xauthority
drwx--x--x 20 dgerogia grads   1024 Νοέ  8 11:58 csst0123
drwxr-xr-x  3 dgerogia grads    512 Νοέ  8 16:06 GNUstep
drwx-----  2 dgerogia grads    512 Νοέ  8 17:02 Mail
drwxr-xr-x  3 dgerogia grads    512 Νοέ  8 15:44 public_html
bash-2.05$

```

Εικόνα 22: ls -al για να δούμε περισσότερες πληροφορίες για όλα τα αρχεία και φακέλους (κρυφά και μη)

Ας δούμε τώρα τι πληροφορίες επιστρέφονται από την “ls -l”



Εικόνα 23: Επεξήγηση των πληροφοριών που επιστρέφει η ls -l

```

Terminal
Window Edit Options Help
arthur % ls UNIX
dir1      dir3      file4
dir2      file1.txt
arthur % ls -d UNIX
UNIX
arthur % █

```

Το αρχείο UNIX είναι ένας κατάλογος

Η διαφορά ls και ls -d είναι ότι το όριμα -d εμφανίζει μόνο τους καταλόγους και όχι τα απλά αρχεία

Εικόνα 24: Επεξήγηση των πληροφοριών που επιστρέφει η ls -d

```

Terminal
Window Edit Options Help
arthur % ls
dir1      dir3      file4
dir2      file1.txt
arthur % ls -l
dir1
dir2
dir3
file1.txt
file4
arthur % ls -R
dir1      dir3      file4
dir2      file1.txt

dir1:
dir2

dir1/dir2:
fb_aa    fb_ab    file2.txt

dir2:

dir3:
file3.txt
arthur % █

```

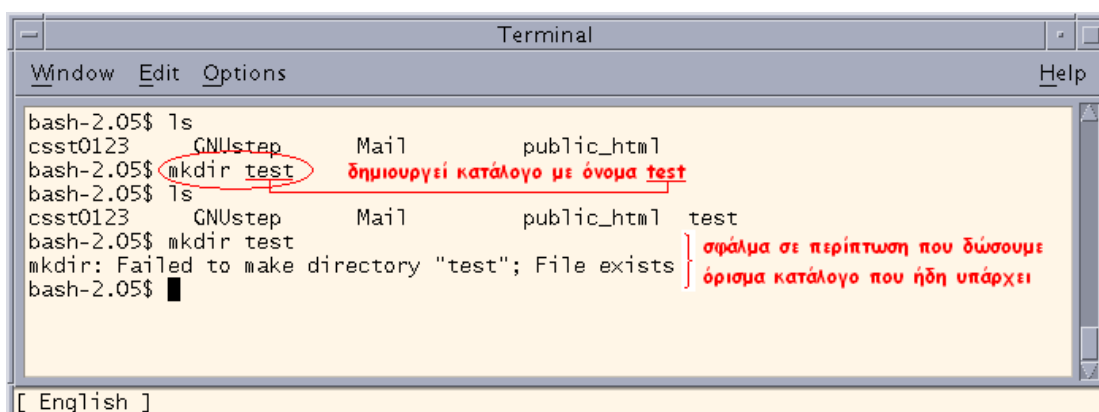
Η ls -l εμφανίζει ότι και η ls αλλά σε μια στήλη

με το όρισμα -R εμφανίζονται αναδρομικά όλα τα περιεχόμενα των φακέλων και υποφακέλων

Εικόνα 25: Επεξήγηση των πληροφοριών που επιστρέφει η ls -R, ls -l

### Η εντολή mkdir (= make directory)

Στην ενότητα αυτή θα δούμε πώς δημιουργούνται κατάλογοι στο UNIX. Η λειτουργία αυτή επιτελείτε στο Unix με την εντολή mkdir, η οποία και περιγράφεται στην εικόνα που ακολουθεί.



```
Terminal
Window Edit Options Help
bash-2.05$ ls
csst0123  GNUstep  Mail      public_html
bash-2.05$ mkdir test
bash-2.05$ ls
csst0123  GNUstep  Mail      public_html  test
bash-2.05$ mkdir test
mkdir: Failed to make directory "test"; File exists
bash-2.05$
```

δημιουργεί κατάλογο με όνομα test

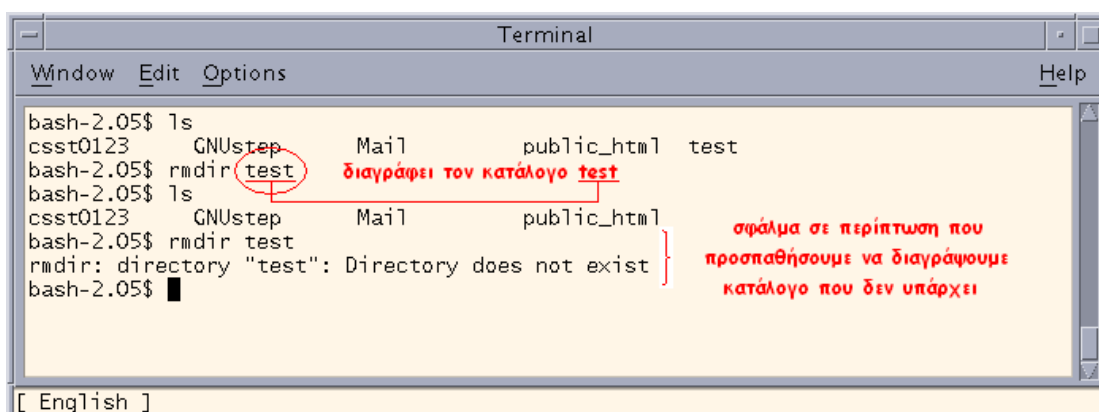
σφάλμα σε περίπτωση που δώσουμε όρισμα κατάλογο που ήδη υπάρχει

[ English ]

Εικόνα 26: Δημιουργία καταλόγου

### Η εντολή rmdir (=remove directory)

Αφού μάθαμε να δημιουργούμε καταλόγους και να βλέπουμε τα περιεχόμενά τους, θα κλείσουμε την ενότητα αυτή, δείχνοντας πώς μπορούμε να διαγράψουμε καταλόγους. Για το σκοπό αυτό θα χρησιμοποιήσουμε την εντολή rmdir, η οποία περιγράφεται στις παρακάτω εικόνες.



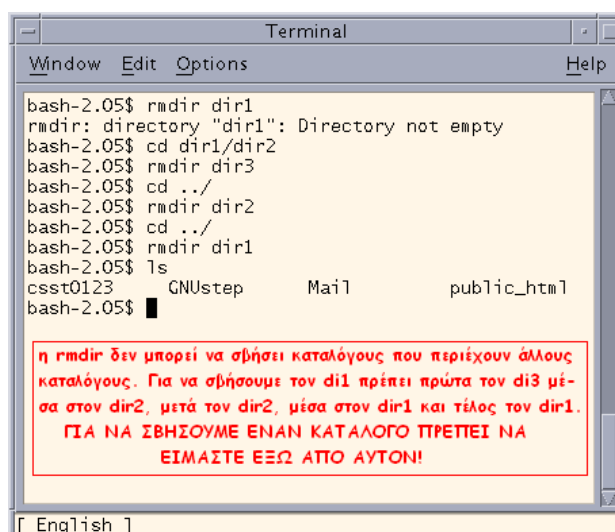
```
Terminal
Window Edit Options Help
bash-2.05$ ls
csst0123  GNUstep  Mail      public_html  test
bash-2.05$ rmdir test
bash-2.05$ ls
csst0123  GNUstep  Mail      public_html
bash-2.05$ rmdir test
rmdir: directory "test": Directory does not exist
bash-2.05$
```

διαγράφει τον κατάλογο test

σφάλμα σε περίπτωση που προσπαθήσουμε να διαγράψουμε κατάλογο που δεν υπάρχει

[ English ]

Εικόνα 27: Διαγραφή καταλόγου



```
Terminal
Window Edit Options Help
bash-2.05$ rmdir dir1
rmdir: directory "dir1": Directory not empty
bash-2.05$ cd dir1/dir2
bash-2.05$ rmdir dir3
bash-2.05$ cd ../
bash-2.05$ rmdir dir2
bash-2.05$ cd ../
bash-2.05$ rmdir dir1
bash-2.05$ ls
csst0123  GNUstep  Mail      public_html
bash-2.05$
```

η rmdir δεν μπορεί να σβήσει καταλόγους που περιέχουν άλλους καταλόγους. Για να σβήσουμε τον dir1 πρέπει πρώτα τον dir3 μέσα στον dir2, μετά τον dir2, μέσα στον dir1 και τέλος τον dir1.  
ΓΙΑ ΝΑ ΣΒΗΣΙΟΥΜΕ ΕΝΑΝ ΚΑΤΑΛΟΓΟ ΠΡΕΠΕΙ ΝΑ ΕΙΜΑΣΤΕ ΕΞΩ ΑΠΟ ΑΥΤΟΝ!

[ English ]

Εικόνα 28: Διαγραφή καταλόγου που δεν είναι άδειος → σφάλμα

### Η εντολή mv

Η βασική χρήση της mv είναι να μετακινεί αρχεία και καταλόγους από μία θέση σε μία άλλη. Έτσι αν θεωρήσουμε ότι ένα αρχείο ή κατάλογος μεταφέρεται στην ίδια θέση, αλλά με άλλο όνομα, τότε ουσιαστικά έχουμε πετύχει την μετονομασία. Ο τρόπος χρήσης της mv δίνεται στις παρακάτω εικόνες. Παρατηρήστε ότι κατά την μετακίνηση αρχείων η mv διαγράφει το αρχείο από την αρχική θέση που βρισκόταν. Η εικόνα στη συνέχεια περιγράφει τον τρόπο λειτουργίας της mv.

```
bash-2.05$ ls -l
total 4
drwxr-xr-x  2 dgerogia grads      512 Νοέ  22 13:11 dir1
drwxr-xr-x  2 dgerogia grads      512 Νοέ  22 13:11 dir2
bash-2.05$ ls -l dir1/
total 0
-rwxrwxrwx  1 dgerogia grads        0 Νοέ  22 13:06 data.txt
bash-2.05$ ls -l dir2/
total 0
bash-2.05$ mv dir1/data.txt dir2/
bash-2.05$ ls -l dir1/
total 0
bash-2.05$ mv dir1/data.txt dir2/
mv: cannot access dir1/data.txt
bash-2.05$ ls -l dir2/
total 0
-rwxrwxrwx  1 dgerogia grads        0 Νοέ  22 13:06 data.txt
bash-2.05$ █
```

**Στον δίσκο υπάρχει ΜΟΝΟ ΕΝΑ αρχείο data.txt**

Μεταφέρει το data.txt από τον κατάλογο dir1 στον dir2

Η mv χρησιμοποιείται για να μετονομάσουμε ή να μετακινήσουμε ένα αρχείο. Στην περίπτωση μετακίνησης, λειτουργεί όπως και η cp ΑΛΛΑ διαγράφει το αρχείο από τον κατάλογο από τον οποίο αρχίζει η μετακίνηση. Παρατηρήστε παραπάνω ότι αρχικά υπάρχει το αρχείο data.txt στον κατάλογο dir1, όμως μετά το mv το αρχείο αντιγράφηκε στον κατάλογο dir2, διεγράφη όμως από τον dir1.

Εικόνα 29: Χρήση της mv για μετακίνηση αρχείου.

```
bash-2.05$ cat data1.txt
123
bash-2.05$ mv data1.txt data2.txt
bash-2.05$ cat data2.txt
123
bash-2.05$ ls
data2.txt  dir1      dir2
bash-2.05$ █
```

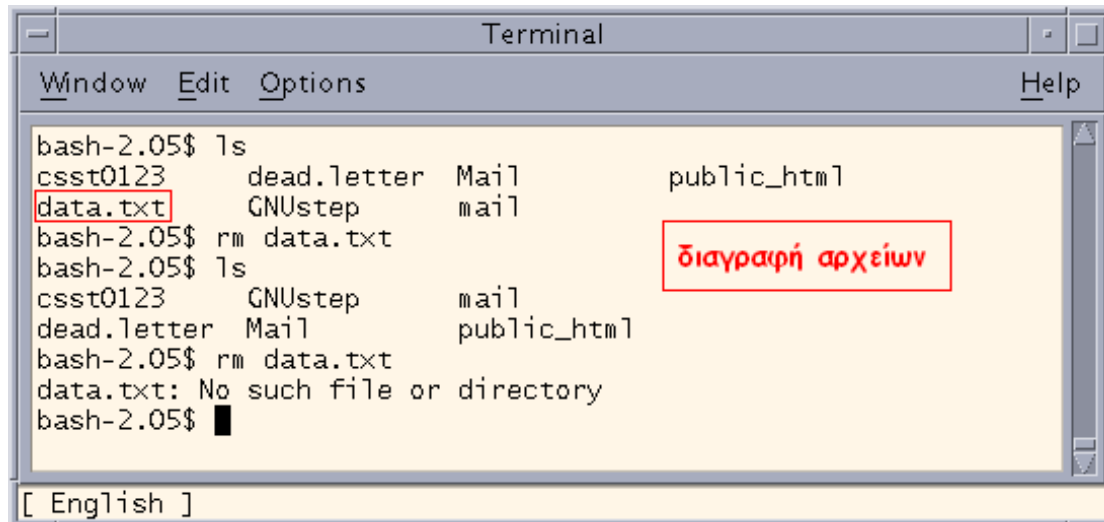
Μετονομασία του αρχείου data1.txt σε data2.txt

Τα περιεχόμενα του αρχείου δεν έχουν αλλάξει

Εικόνα 30: Χρήση της mv για μετονομασία αρχείου..

## Η εντολή rm

Αντίστοιχα με την rmdir, που διαγράφει καταλόγους, υπάρχει και η rm που διαγράφει αρχεία. Με κατάλληλα ορίσματα όμως μπορεί να χρησιμοποιηθεί για να διαγράψει και καταλόγους. Στις εικόνες που ακολουθούν φαίνεται ένα παράδειγμα για το πώς λειτουργεί η rm.

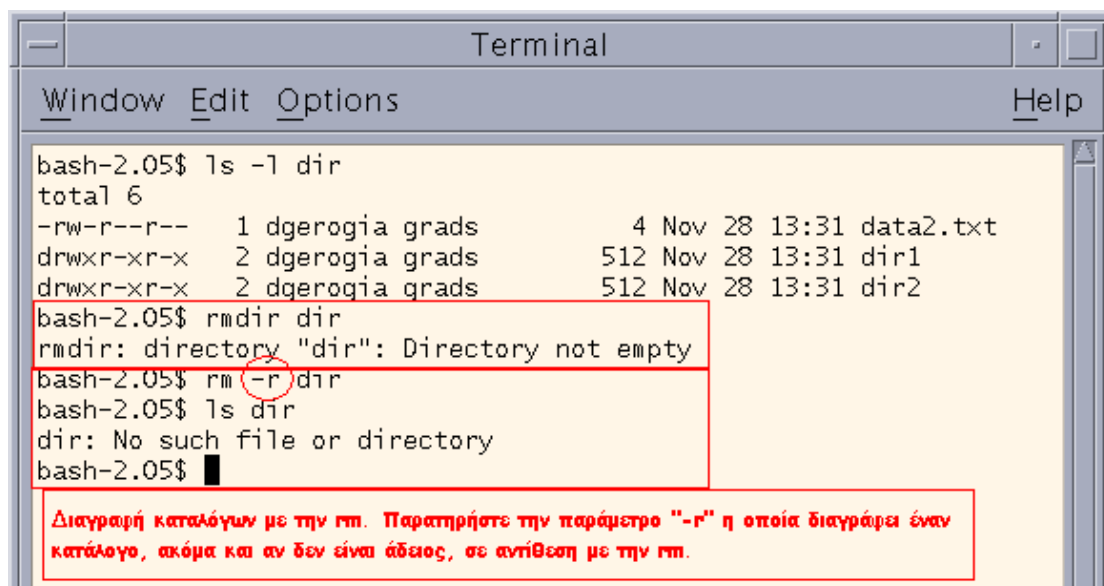


```
Terminal
Window Edit Options Help
bash-2.05$ ls
csst0123  dead.letter  Mail      public_html
data.txt  GNUstep      mail
bash-2.05$ rm data.txt
bash-2.05$ ls
csst0123  GNUstep      mail
dead.letter  Mail      public_html
bash-2.05$ rm data.txt
data.txt: No such file or directory
bash-2.05$
```

διαγραφή αρχείων

[ English ]

Εικόνα 31: Διαγραφή αρχείου με την rm.



```
Terminal
Window Edit Options Help
bash-2.05$ ls -l dir
total 6
-rw-r--r--  1 dgerogia grads      4 Nov 28 13:31 data2.txt
drwxr-xr-x  2 dgerogia grads    512 Nov 28 13:31 dir1
drwxr-xr-x  2 dgerogia grads    512 Nov 28 13:31 dir2
bash-2.05$ rmdir dir
rmdir: directory "dir": Directory not empty
bash-2.05$ rm -r dir
bash-2.05$ ls dir
dir: No such file or directory
bash-2.05$
```

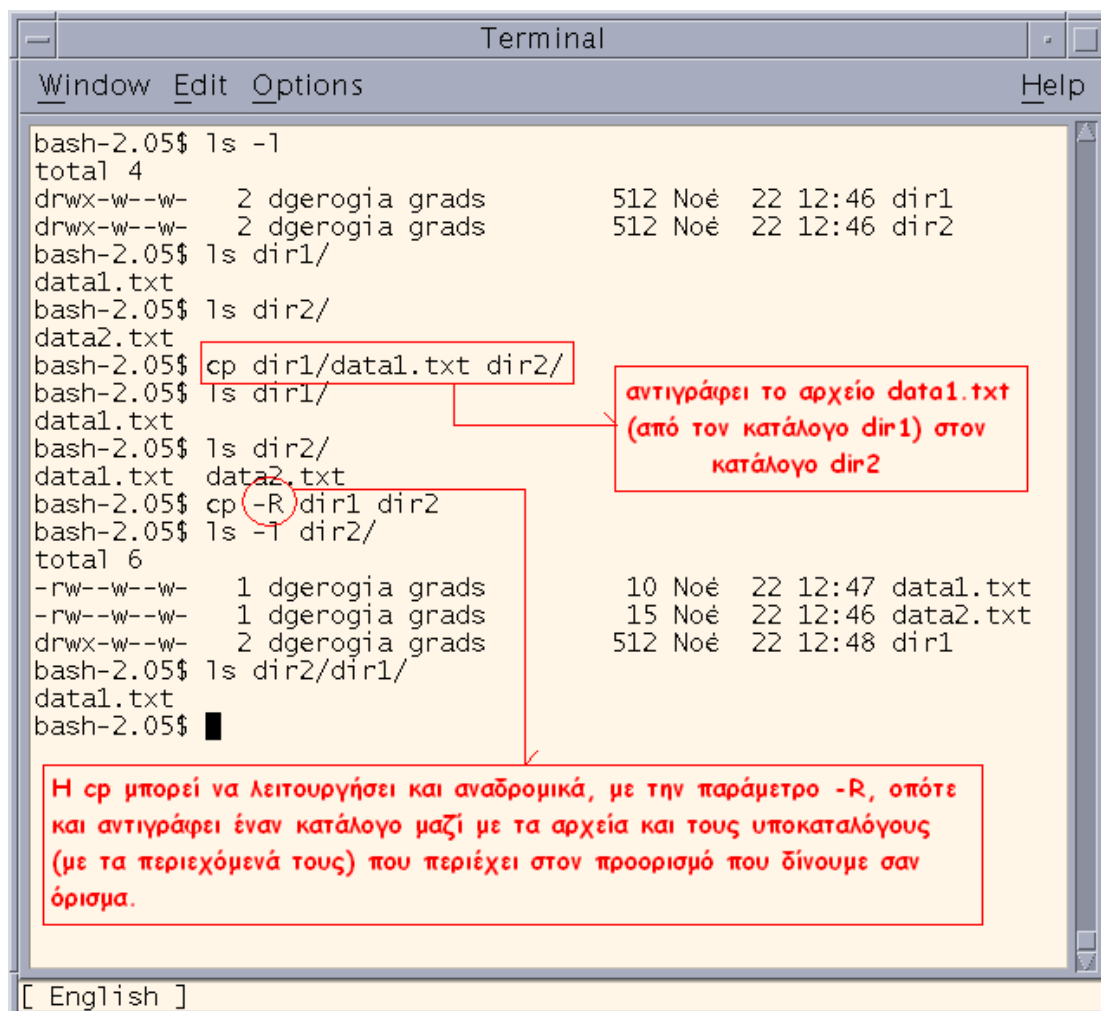
Διαγραφή καταλόγων με την rm. Παρατηρήστε την παράμετρο "-r" η οποία διαγράφει έναν κατάλογο, ακόμα και αν δεν είναι άδειος, σε αντίθεση με την rm.

Εικόνα 32: Διαγραφή καταλόγου με την rm.



## Η εντολή cp

Θα ολοκληρώσουμε το εργαστήριο με την μελέτη της εντολής cp. Η εντολή αυτή αντιγράφει αρχεία και καταλόγους από μία θέση σε μία άλλη. Η λειτουργία της είναι ίδια με την mv, με μόνη διαφορά ότι δεν διαγράφεται το αρχικό αρχείο. Έτσι μετά το τέλος της αντιγραφής, στο δίσκο έχουμε δύο αντίγραφα του αρχείου ή του καταλόγου. Ο τρόπος χρήσης της cp περιγράφεται στην παρακάτω εικόνα.



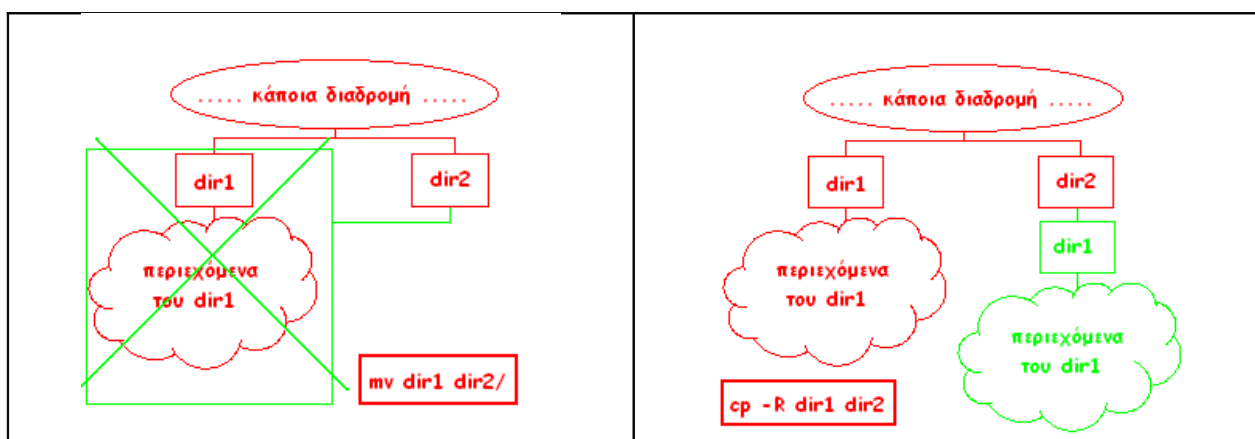
```
bash-2.05$ ls -l
total 4
drwx-w--w-  2 dgerogia grads      512 Νοέ  22 12:46 dir1
drwx-w--w-  2 dgerogia grads      512 Νοέ  22 12:46 dir2
bash-2.05$ ls dir1/
data1.txt
bash-2.05$ ls dir2/
data2.txt
bash-2.05$ cp dir1/data1.txt dir2/
bash-2.05$ ls dir1/
data1.txt
bash-2.05$ ls dir2/
data1.txt data2.txt
bash-2.05$ cp -R dir1 dir2
bash-2.05$ ls -l dir2/
total 6
-rw--w--w-  1 dgerogia grads      10 Νοέ  22 12:47 data1.txt
-rw--w--w-  1 dgerogia grads      15 Νοέ  22 12:46 data2.txt
drwx-w--w-  2 dgerogia grads      512 Νοέ  22 12:48 dir1
bash-2.05$ ls dir2/dir1/
data1.txt
bash-2.05$
```

αντιγράφει το αρχείο data1.txt (από τον κατάλογο dir1) στον κατάλογο dir2

Η cp μπορεί να λειτουργήσει και αναδρομικά, με την παράμετρο -R, οπότε και αντιγράφει έναν κατάλογο μαζί με τα αρχεία και τους υποκαταλόγους (με τα περιεχόμενά τους) που περιέχει στον προορισμό που δίνουμε σαν όρισμα.

[ English ]

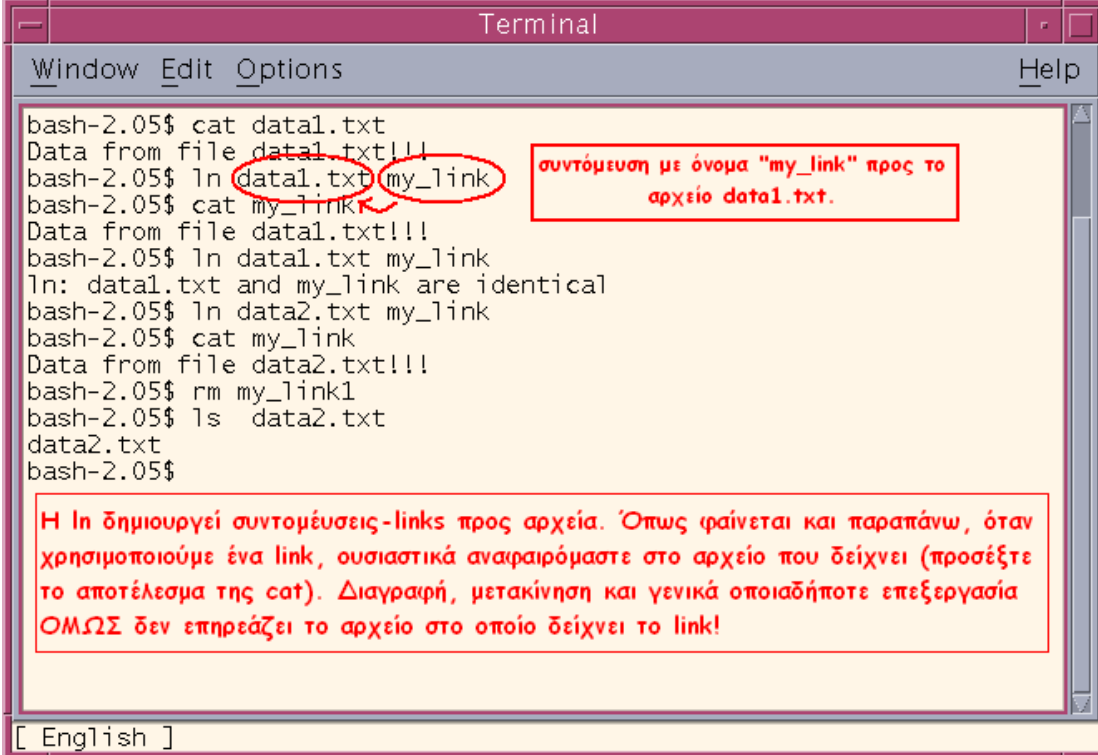
Εικόνα 33: Αντιγραφή καταλόγου και αρχείου με τη cp.



Εικόνα 34: Παράδειγμα διαφοράς mv και cp.

### Η εντολή ln

Όπως είχαμε επισημάνει όταν εξηγήσαμε την εντολή ls, υπάρχει περίπτωση ένα αρχείο να είναι συντόμευση (link) προς ένα άλλο αρχείο. Μπορούμε να πετύχουμε τη δημιουργία συντόμευσης προς ένα αρχείο με τη βοήθεια της εντολής ln, η οποία παρουσιάζεται στην παρακάτω εικόνα. Παρατηρήστε ότι οι αλλαγές που κάνουμε στη συντόμευση (π.χ. διαγραφή αλλά και γενικά κάθε λειτουργία) δεν επηρεάζουν το αρχικό αρχείο.



```
Terminal
Window Edit Options Help
bash-2.05$ cat data1.txt
Data from file data1.txt!!!
bash-2.05$ ln data1.txt my_link
bash-2.05$ cat my_link
Data from file data1.txt!!!
bash-2.05$ ln data1.txt my_link
ln: data1.txt and my_link are identical
bash-2.05$ ln data2.txt my_link
bash-2.05$ cat my_link
Data from file data2.txt!!!
bash-2.05$ rm my_link1
bash-2.05$ ls data2.txt
data2.txt
bash-2.05$
```

συντόμευση με όνομα "my\_link" προς το αρχείο data1.txt.

Η ln δημιουργεί συντομέυσεις-links προς αρχεία. Όπως φαίνεται και παραπάνω, όταν χρησιμοποιούμε ένα link, ουσιαστικά αναφαιρόμαστε στο αρχείο που δείχνει (προσέξτε το αποτέλεσμα της cat). Διαγραφή, μετακίνηση και γενικά οποιαδήποτε επεξεργασία ΟΜΩΣ δεν επηρεάζει το αρχείο στο οποίο δείχνει το link!

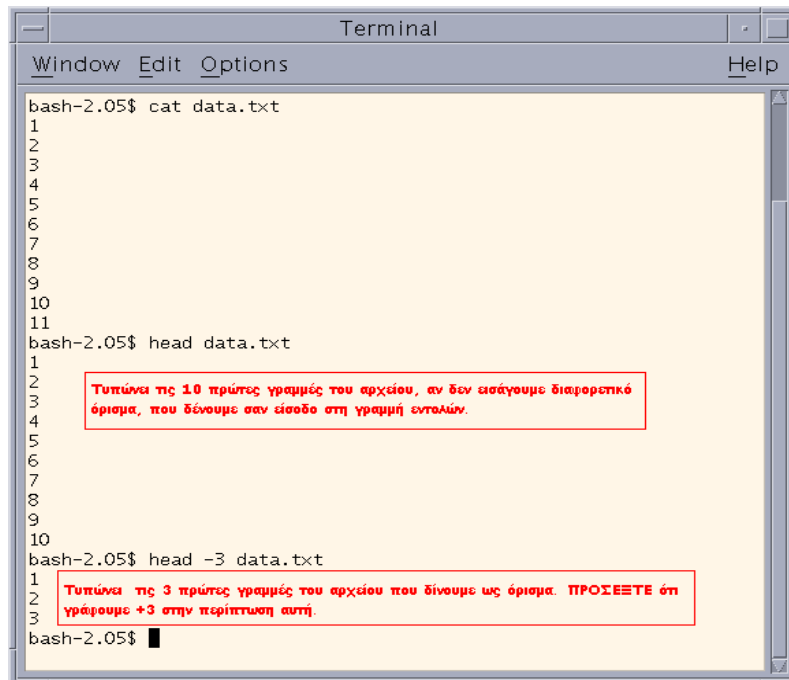
[ English ]

Εικόνα 35: Η εντολή ln.

## 2.4 Διαχείριση αρχείων κειμένου – head, tail, wc, grep, sort, more, cat

### Η εντολή head

Προκειμένου να τυπώνουμε ένα αρχείο κειμένου κατά γραμμές, το Unix διαθέτει το βοήθημα head. Η λειτουργία της εντολής περιγράφεται στις εικόνες που ακολουθούν.

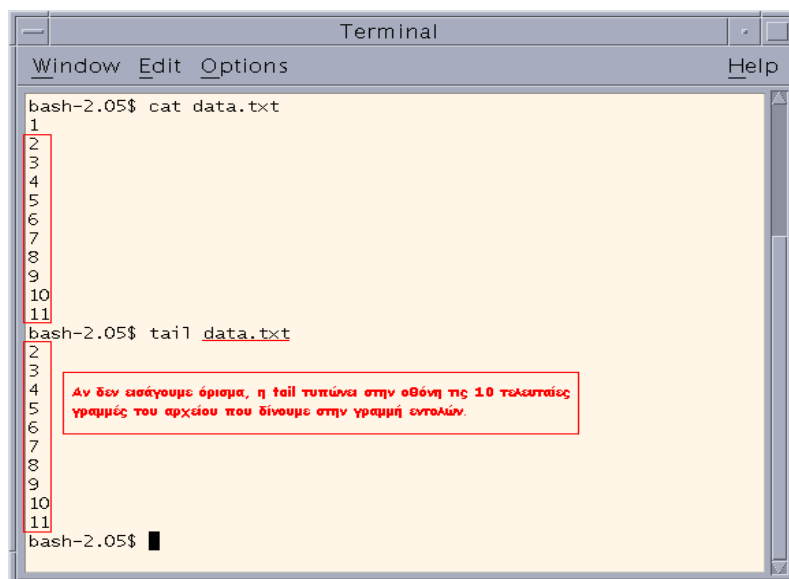


```
Terminal
Window Edit Options Help
bash-2.05$ cat data.txt
1
2
3
4
5
6
7
8
9
10
11
bash-2.05$ head data.txt
1
2
3
4
5
6
7
8
9
10
11
bash-2.05$ head -3 data.txt
1
2
3
bash-2.05$
```

Εικόνα 36: Η εντολή head για προβολή των γραμμών ενός αρχείου.

### Η εντολή tail

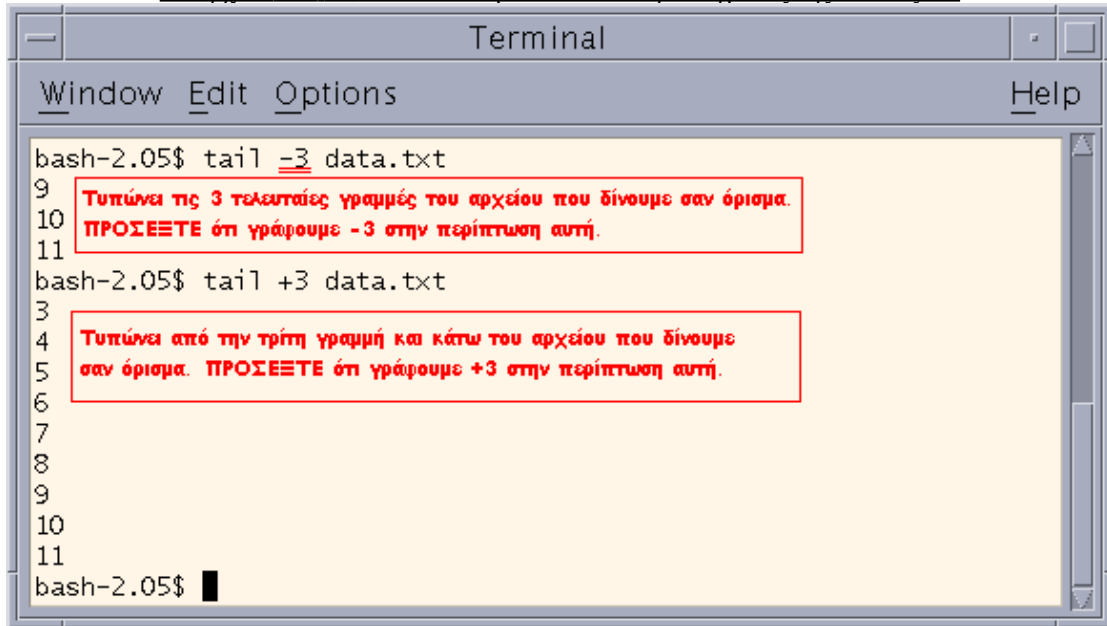
Αντίστοιχα με την head, υπάρχει η εντολή tail, που εκτελεί τις ίδιες λειτουργίες, αλλά ανάποδα, δηλαδή ξεκινώντας από το τέλος προς την αρχή. Η εντολή tail Περιγράφεται στις παρακάτω εικόνες.



```
Terminal
Window Edit Options Help
bash-2.05$ cat data.txt
1
2
3
4
5
6
7
8
9
10
11
bash-2.05$ tail data.txt
2
3
4
5
6
7
8
9
10
11
bash-2.05$
```

Εικόνα 37: Η εντολή tail για προβολή των γραμμών ενός αρχείου.

Το αρχείο data.txt είναι το ίδιο με αυτό του παραδείγματος της εικόνας 36.



```
Terminal
Window Edit Options Help
bash-2.05$ tail -3 data.txt
9
10
11
bash-2.05$ tail +3 data.txt
3
4
5
6
7
8
9
10
11
bash-2.05$
```

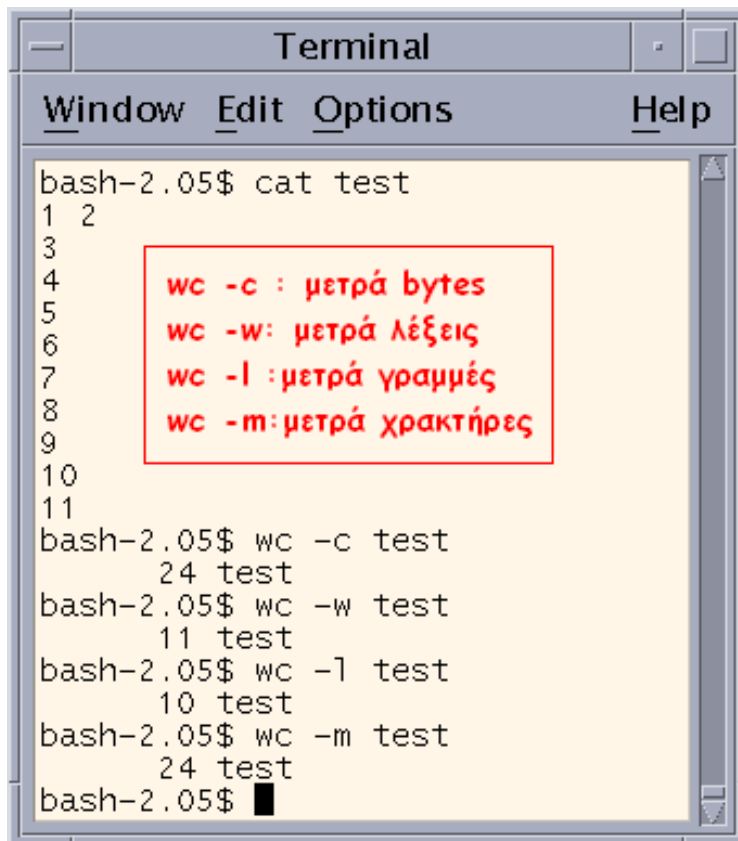
Annotations in red boxes:

- Line 9: **Τυπώνω τις 3 τελευταίες γραμμές του αρχείου που δίνουμε σαν όρισμα. ΠΡΟΣΕΞΤΕ ότι γράφουμε -3 στην περίπτωση αυτή.**
- Line 4: **Τυπώνω από την τρίτη γραμμή και κάτω του αρχείου που δίνουμε σαν όρισμα. ΠΡΟΣΕΞΤΕ ότι γράφουμε +3 στην περίπτωση αυτή.**

Εικόνα 38: Η εντολή tail (συνέχεια). Το αρχείο data.txt είναι το ίδιο με αυτό του παραδείγματος της εικόνας 36.

#### Η εντολή wc

Η τελευταία εντολή με την οποία θα ασχοληθούμε στην ενότητα αυτή εκτελεί μετρήσεις σε αρχεία. Τα δυνατά αποτελέσματα που επιστρέφει μπορείτε να τα δείτε στην εικόνα που ακολουθεί.



```
Terminal
Window Edit Options Help
bash-2.05$ cat test
1 2
3
4
5
6
7
8
9
10
11
bash-2.05$ wc -c test
24 test
bash-2.05$ wc -w test
11 test
bash-2.05$ wc -l test
10 test
bash-2.05$ wc -m test
24 test
bash-2.05$
```

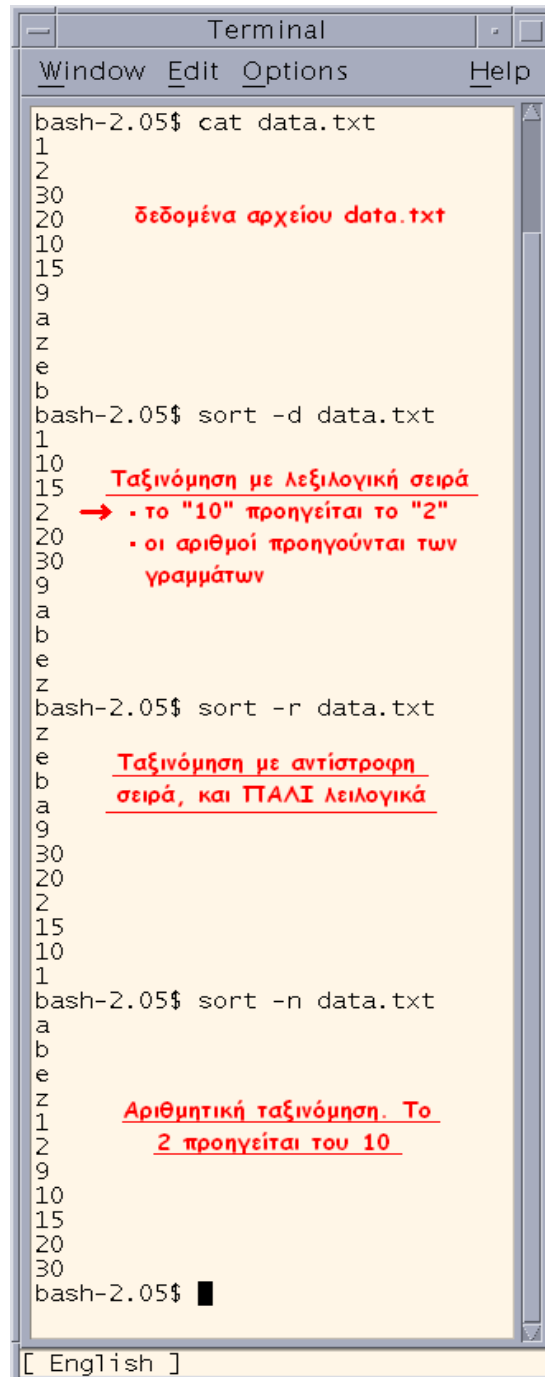
Annotations in red box:

- wc -c : μετρά bytes**
- wc -w : μετρά λέξεις**
- wc -l : μετρά γραμμές**
- wc -m : μετρά χαρακτήρες**

Εικόνα 39: Η εντολή wc.

## Η εντολή sort

Η εντολή sort είναι αρκετά απλή και πετυχαίνει την ταξινόμηση των περιεχομένων ενός αρχείου κειμένου, ανάλογα με τα ορίσματα που δίνουμε ως είσοδο. Παράδειγμα της sort δίνεται στην εικόνα που ακολουθεί.



```
Terminal
Window Edit Options Help
bash-2.05$ cat data.txt
1
2
30
20
10
15
9
a
z
e
b
bash-2.05$ sort -d data.txt
1
10
15
2
20
30
9
a
b
e
z
bash-2.05$ sort -r data.txt
z
e
b
a
9
30
20
2
15
10
1
bash-2.05$ sort -n data.txt
a
b
e
z
1
2
9
10
15
20
30
bash-2.05$
```

**δεδομένα αρχείου data.txt**

**Ταξινόμηση με λεξιλογική σειρά**  
→ • το "10" προηγείται το "2"  
• οι αριθμοί προηγούνται των γραμμάτων

**Ταξινόμηση με αντίστροφη σειρά, και ΠΑΛΙ λειλογικά**

**Αριθμητική ταξινόμηση. Το 2 προηγείται του 10**

[ English ]

Εικόνα 40: Η εντολή sort.

```
Terminal
Window Edit Options Help
arthur % cat file1.txt
20 103 A new Message
60 20 "A new Message"
10 40 "Line 1"
arthur % sort -t 0 file1.txt
10 40 "Line 1"
20 103 A new Message
60 20 "A new Message"
arthur % sort -t 1 file1.txt
10 40 "Line 1"
20 103 A new Message
60 20 "A new Message"
arthur % sort -t 2 file1.txt
10 40 "Line 1"
20 103 A new Message
60 20 "A new Message"
arthur %
```

Το όρισμα -t κάνει ταξινόμηση με βάση το πεδίο που δίνεται, στην περίπτωση αυτή με βάση την 1η στήλη

Εικόνα 41: Η εντολή sort (συνέχεια).

### Η εντολή grep

Στην υποενότητα αυτή θα δούμε την εντολή grep η οποία επιτρέπει αναζήτηση εκφράσεων σε αρχεία. Με τον όρο έκφραση εννοούμε είτε μία απλή λέξη (π.χ. int) είτε ολόκληρη πρόταση (π.χ. int x,y;). Αν θέλουμε να αναζητήσουμε προτάσεις τότε θα πρέπει να τις εισάγουμε μεταξύ εισαγωγικών (π.χ. "int x,y;"). Η εντολή grep αποδεικνύεται αρκετά χρήσιμη όταν θέλουμε να κάνουμε αναζήτηση προτάσεων σε πολλά κείμενα. Στην περίπτωση αυτή αν επιχειρούσαμε να ψάξουμε ένα-ένα τα αρχεία με το χέρι το πιο πιθανό είναι να εγκαταλείπαμε την όλη διαδικασία στα πρώτα κείμενα. Σίγουρα ένα πιο αυτοματοποιημένος τρόπος διευκολύνει την κατάσταση. Μερικές δυνατότητες της grep δίνονται στην εικόνα που ακολουθεί. Η ερμηνεία των ορισμάτων της grep δίνεται στο τέλος της εικόνας.

```
Terminal
Window Edit Options Help
bash-2.05$ cat main.c
#include <stdio.h>
/* some code here */
int main(int argc, char *argv[])
{
    int x1,x2;
    point p;

    return (0);
}
bash-2.05$ cat hello.c
#include <stdio.h>
void main(void)
{
    printf ("Hello World!\n");
}
bash-2.05$ grep int main.c
int main(int argc, char *argv[])
{
    int x1,x2;
    point p;
bash-2.05$ grep -w int main.c
int main(int argc, char *argv[])
{
    int x1,x2;
bash-2.05$ grep -n int main.c
3:int main(int argc, char *argv[])
4:{
5:    int x1,x2;
6:    point p;
bash-2.05$ grep int main.c hello.c
main.c:int main(int argc, char *argv[])
main.c:{
main.c:    int x1,x2;
main.c:    point p;
hello.c:{
hello.c:printf ("Hello World!\n");
bash-2.05$ grep -lw main.c hello.c
bash-2.05$ grep -lw int main.c hello.c
main.c
bash-2.05$ grep -v int main.c
#include <stdio.h>
/* some code here */

return (0);
}
bash-2.05$ █
```

**grep ...ορίσματα... ...έκφραση... ...αρχεία...**  
Τα ορίσματα μπορεί να είναι (και συνδυασμός αυτών):

- n: τυπώνει αριθμό γραμμής που εντοπίστηκε η έκφραση
- w: αναζητά ολόκληρες λέξεις
- l: τυπώνει μία λίστα με τα αρχεία στα οποία βρέθηκε η έκφραση
- v: τυπώνει όλες γραμμές δεν περιέχουν την έκφραση

Η παράλειψη ορισμάτων οδηγεί σε αναζήτηση γραμμών που περιέχουν την έκφραση είτε ως έχει είτε ως μέρος άλλης μεγαλύτερης (π.χ. παραπάνω int, point). Μπορούμε να κάνουμε αναζήτηση σε πολλά αρχεία. Στην περίπτωση αυτή στην έξοδο θα τυπώνεται το όνομα του αρχείου πριν τη γραμμή.

Εικόνα 42: Η εντολή grep.

### Η εντολή more

Πολλές φορές όμως τα δεδομένα είναι πολλά και δεν χωράνε σε μία οθόνη. Το Unix παρέχει την εντολή more για να μπορούμε να βλέπουμε τμηματικά τα περιεχόμενα ενός αρχείου κειμένου. Ο τρόπος χρήσης είναι ίδιος με την εντολή man. Η χρήση της more δίνεται στην παρακάτω εικόνα.

```
Terminal
Window Edit Options Help
bash-2.05$ more solver.f
BLOCK DATA INIT6
IMPLICIT DOUBLE PRECISION (A-H,O-Z)

COMMON /TRN_FLAG/ TF
COMMON /SOLUTION_SPACE/ UPPER, LOWER
END

DOUBLE PRECISION FUNCTION FACTORIAL (N)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
T=1.
DO 90, I=2,N
  T=T*I
  FACTORIAL=T
RETURN
END

DOUBLE PRECISION FUNCTION PERM (N,K)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
PERM=FACTORIAL (N) / (FACTORIAL (K) *FACTORIAL (N-K))
RETURN
END

DOUBLE PRECISION FUNCTION EULERIAN (K,N)
--More-- (4%)
[ English ]
```

Εικόνα 43: Η εντολή more για προβολή των περιεχομένων του αρχείου solver.f.

### Η εντολή cat

Η cat είναι η εντολή που μας επιτρέπει να προβάλλουμε στην οθόνη τα περιεχόμενα ενός αρχείου. Παράδειγμα της cat δίνεται στην εικόνα που ακολουθεί. Η διαφορά με την more είναι ότι τα περιεχόμενα προβάλλονται μονομιάς χωρίς διακοπές. Έτσι ένα μεγάλο αρχείο δεν είναι εύκολο να προβληθεί με την cat.

```
Terminal
Window Edit Options Help
arthur % cat file1.txt
20 103 A new Message
60 20 "A new Message"
10 40 "Line 1"
arthur %
```

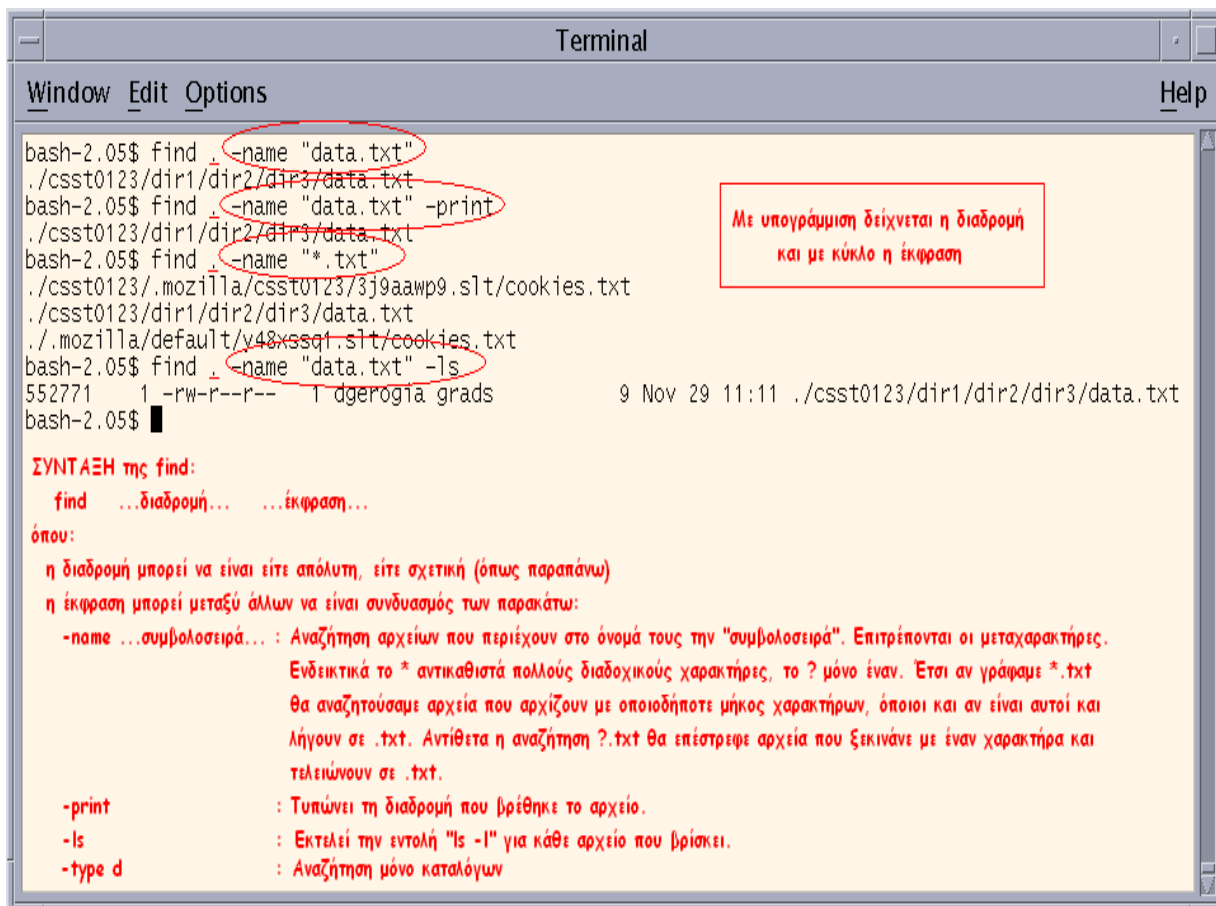
Εικόνα 44: Η εντολή cat.



## 2.5 Προχωρημένη διαχείριση αρχείων – find, diff, split, tar, gzip, file

### Η εντολή find

Μέχρι τώρα έχουμε ασχοληθεί με την αντιγραφή, διαγραφή και τροποποίηση αρχείων. Θα δούμε τώρα το βοήθημα find, το οποίο επιτρέπει την αναζήτηση αρχείων, ανάλογα με τα ορίσματα. Ορισμένες δυνατότητες της εντολής find περιγράφεται στην εικόνα που ακολουθεί.



```
Terminal
Window Edit Options Help
bash-2.05$ find -name "data.txt"
./csst0123/dir1/dir2/dir3/data.txt
bash-2.05$ find -name "data.txt" -print
./csst0123/dir1/dir2/dir3/data.txt
bash-2.05$ find -name "*.txt"
./csst0123/.mozilla/csst0123/3j9aawp9.slt/cookies.txt
./csst0123/dir1/dir2/dir3/data.txt
./mozilla/default/v48xssq1.slt/cookies.txt
bash-2.05$ find -name "data.txt" -ls
552771 1 -rw-r--r-- 1 dgerogia grads 9 Nov 29 11:11 ./csst0123/dir1/dir2/dir3/data.txt
bash-2.05$
```

**ΣΥΝΤΑΞΗ της find:**  
find ...διαδρομή... ...έκφραση...

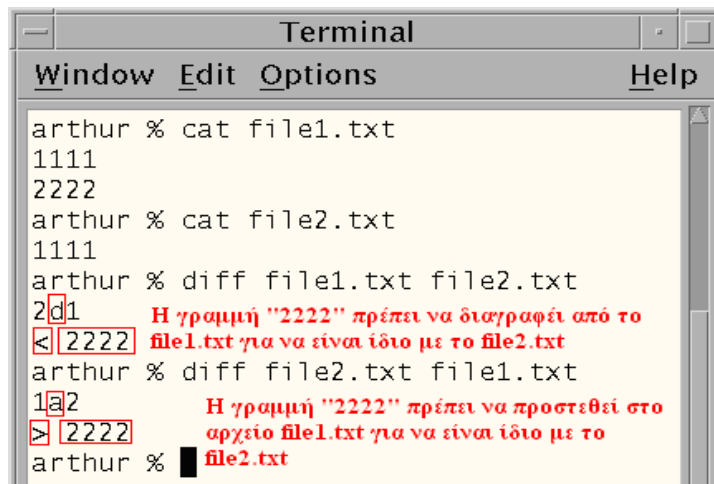
όπου:

- η διαδρομή μπορεί να είναι είτε απόλυτη, είτε σχετική (όπως παραπάνω)
- η έκφραση μπορεί μεταξύ άλλων να είναι συνδυασμός των παρακάτω:
  - name ...συμβολοσειρά... : Αναζήτηση αρχείων που περιέχουν στο όνομά τους την "συμβολοσειρά". Επιτρέπονται οι μεταχαρακτήρες. Ενδεικτικά το \* αντικαθιστά πολλούς διαδοχικούς χαρακτήρες, το ? μόνο έναν. Έτσι αν γράψαμε \*.txt θα αναζητούσαμε αρχεία που αρχίζουν με οποιοδήποτε μήκος χαρακτήρων, όποιοι και αν είναι αυτοί και λήγουν σε .txt. Αντίθετα η αναζήτηση ?.txt θα επέστρεφε αρχεία που ξεκινάνε με έναν χαρακτήρα και τελειώνουν σε .txt.
  - print : Τυπώνει τη διαδρομή που βρέθηκε το αρχείο.
  - ls : Εκτελεί την εντολή "ls -l" για κάθε αρχείο που βρίσκει.
  - type d : Αναζήτηση μόνο καταλόγων

Εικόνα 45: Η εντολή find.

### Η εντολή diff

Η diff είναι μία εντολή που εντοπίζει διαφορές μεταξύ δύο αρχείων και τις τυπώνει στην οθόνη. Το αποτέλεσμα είναι τέτοιο ώστε το ένα αρχείο να μοιάζει στο δεύτερο. Έτσι ανάλογα με τις διαφορές σε άλλες περιπτώσεις θα πρέπει να εισάγονται δεδομένα και σε άλλες να διαγράφονται.

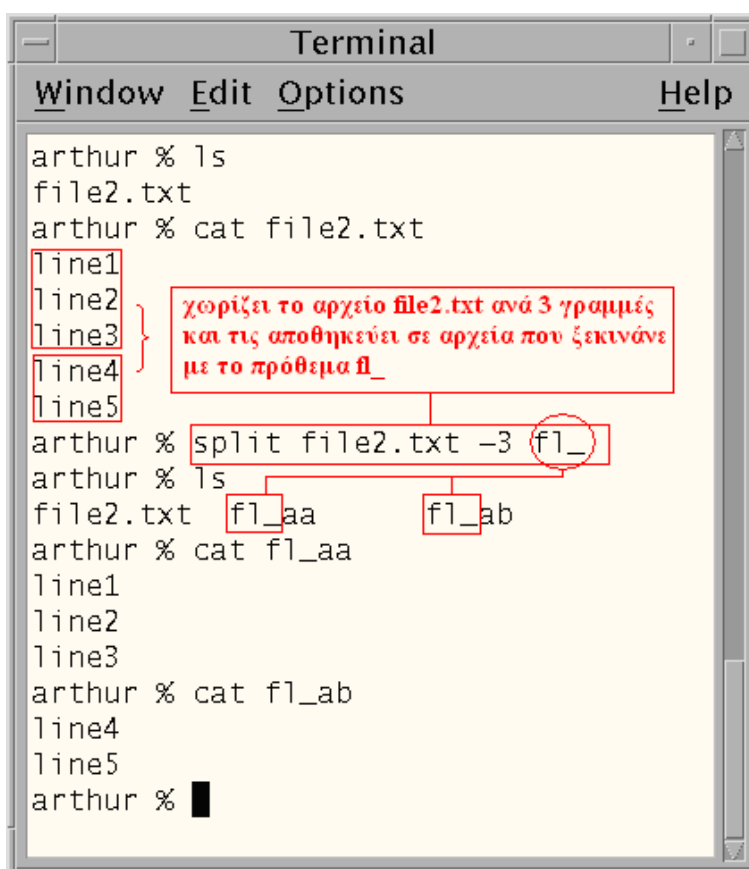


```
Terminal
Window Edit Options Help
arthur % cat file1.txt
1111
2222
arthur % cat file2.txt
1111
arthur % diff file1.txt file2.txt
2d1      Η γραμμή "2222" πρέπει να διαγραφεί από το
< 2222   file1.txt για να είναι ίδιο με το file2.txt
arthur % diff file2.txt file1.txt
1a2      Η γραμμή "2222" πρέπει να προστεθεί στο
> 2222   αρχείο file1.txt για να είναι ίδιο με το
         file2.txt
arthur %
```

Εικόνα 46: Η εντολή diff.

### Η εντολή split

Η split είναι ένα βοήθημα που επιτρέπει τον τεμαχισμό αρχείων με βάση τις επιλογές του χρήστη. Ο τεμαχισμός μπορεί να γίνει είτε κατά γραμμές είτε κατά bytes, δηλαδή χαρακτήρες



```
Terminal
Window Edit Options Help
arthur % ls
file2.txt
arthur % cat file2.txt
line1
line2
line3
line4
line5
arthur % split file2.txt -3 fl_
arthur % ls
file2.txt fl_aa fl_ab
arthur % cat fl_aa
line1
line2
line3
arthur % cat fl_ab
line4
line5
arthur %
```

Εικόνα 47: Η εντολή split.

```

Terminal
Window Edit Options Help
arthur % ls
file2.txt
arthur % cat file2.txt
this is line 1
this is line 2
this is line 3
this is line 4
arthur % split -b30 file2.txt fb_
arthur % ls
fb_aa      fb_ab      file2.txt
arthur % cat fb_aa
this is line 1
this is line 2 } 30 Bytes
arthur % cat fb_ab
this is line 3
this is line 4 } 30 Bytes
arthur % █

```

με το όρισμα -b το αρχείο χωρίζεται σε bytes και όχι με βάση τις γραμμές

-b30

30 Bytes

30 Bytes

Εικόνα 48: Η εντολή split (συνέχεια).

Η εντολή file

Η εντολή file επιστρέφει πληροφορίες για τον τύπο του αρχείου. Ένα παράδειγμα κλήσης της δίνεται στην εικόνα που ακολουθεί.

```

Terminal
Window Edit Options Help
arthur % file dir1
dir1:      directory
arthur % file file1.txt
file1.txt: English text
arthur % file file5.txt
file5.txt: ascii text
arthur % █

```

Δίνονται πληροφορίες για τον τύπο του αρχείου που περνάμε σαν όρισμα.

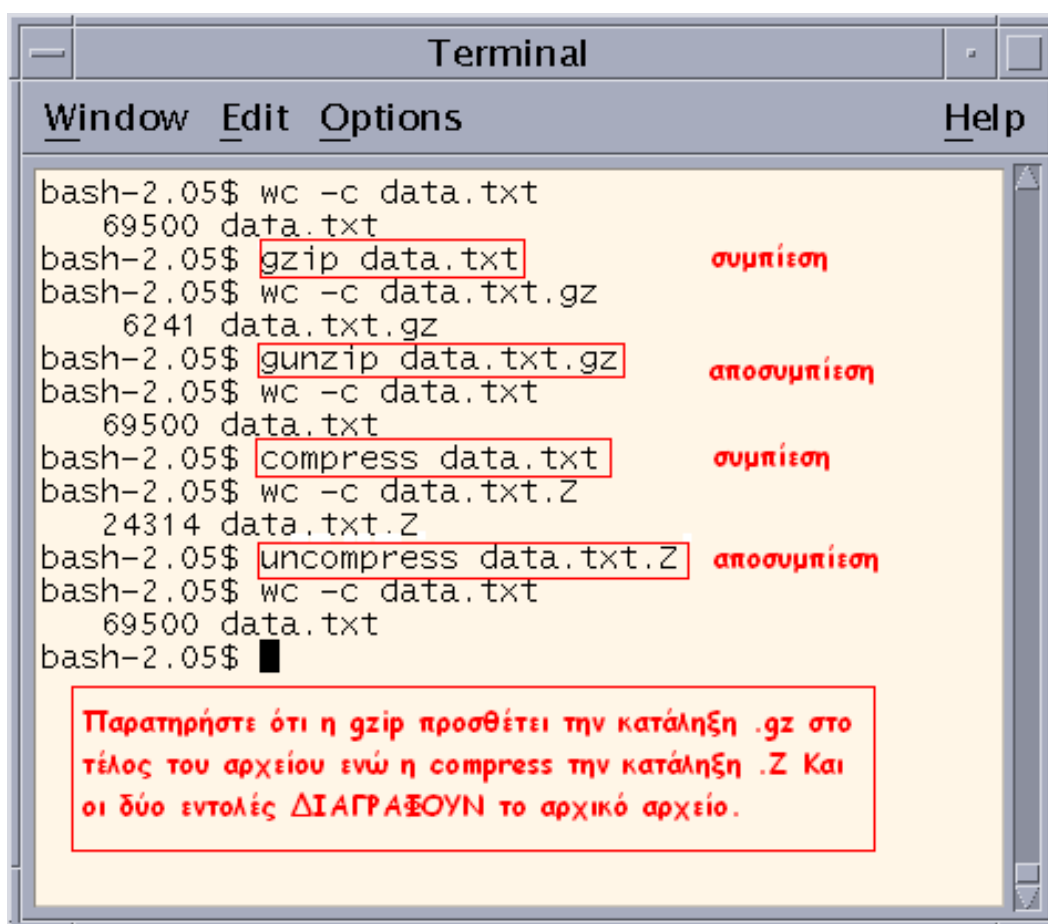
Εικόνα 49: Η εντολή file.

## 2.6 Συμπίεση / αποσυμπίεση και πακέτα αρχείων – *gzip, gunzip, compress, uncompress, tar*

Η εντολές `gzip/gunzip – compress/uncompress`

Πολλές φορές υπάρχει η ανάγκη κάποια μεγάλα αρχεία να συμπιέζονται για να χωρέσουν σε κάποιον αποθηκευτικό χώρο (π.χ. flash memory). Η απαίτηση που υπάρχει είναι τα δεδομένα που συμπιέζονται να μην χάνονται, έτσι ώστε ακολουθώντας την αντίστροφη πορεία της αποσυμπίεσης, να μπορούμε να πάρουμε το αρχικό αρχείο.

Το Unix διαθέτει τις εντολές `gzip/gunzip` και `compress/uncompress` για συμπίεση/αποσυμπίεση αρχείων. Και τα δύο ζευγάρια εκτελούν την ίδια λειτουργία, απλά υλοποιούν διαφορετικό αλγόριθμο συμπίεσης/αποσυμπίεσης. Ο τρόπος χρήσης των εντολών φαίνεται παρακάτω:



```
Terminal
Window Edit Options Help
bash-2.05$ wc -c data.txt
69500 data.txt
bash-2.05$ gzip data.txt
bash-2.05$ wc -c data.txt.gz
6241 data.txt.gz
bash-2.05$ gunzip data.txt.gz
bash-2.05$ wc -c data.txt
69500 data.txt
bash-2.05$ compress data.txt
bash-2.05$ wc -c data.txt.Z
24314 data.txt.Z
bash-2.05$ uncompress data.txt.Z
bash-2.05$ wc -c data.txt
69500 data.txt
bash-2.05$ █
```

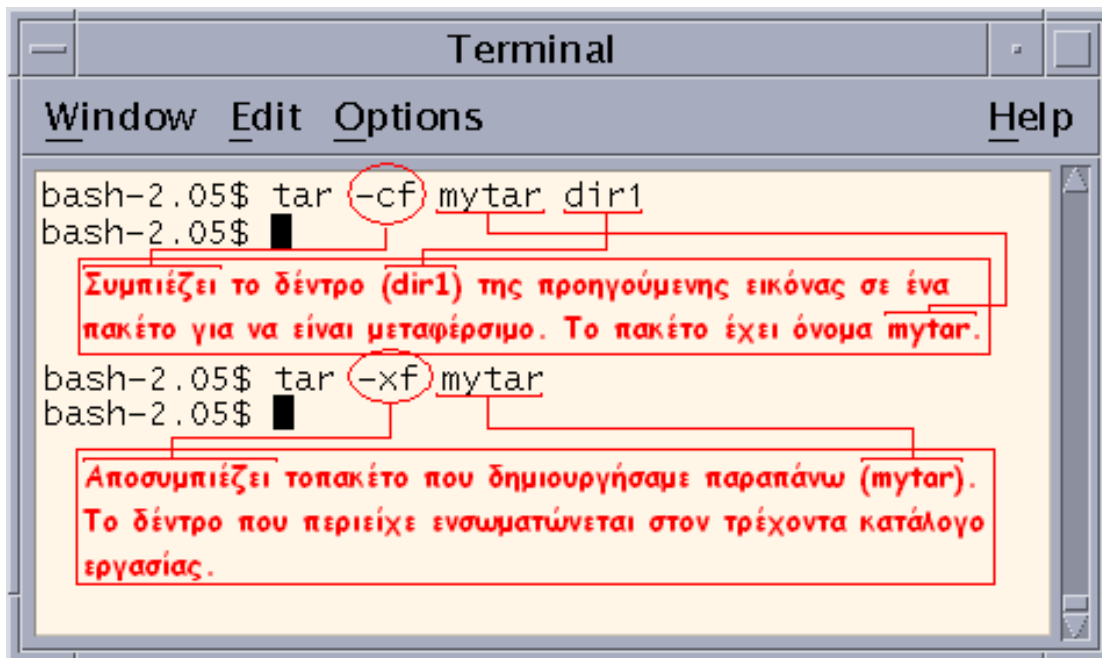
συμπίεση  
αποσυμπίεση  
συμπίεση  
αποσυμπίεση

Παρατηρήστε ότι η `gzip` προσθέτει την κατάληξη `.gz` στο τέλος του αρχείου ενώ η `compress` την κατάληξη `.Z` Και οι δύο εντολές ΔΙΑΓΡΑΦΟΥΝ το αρχικό αρχείο.

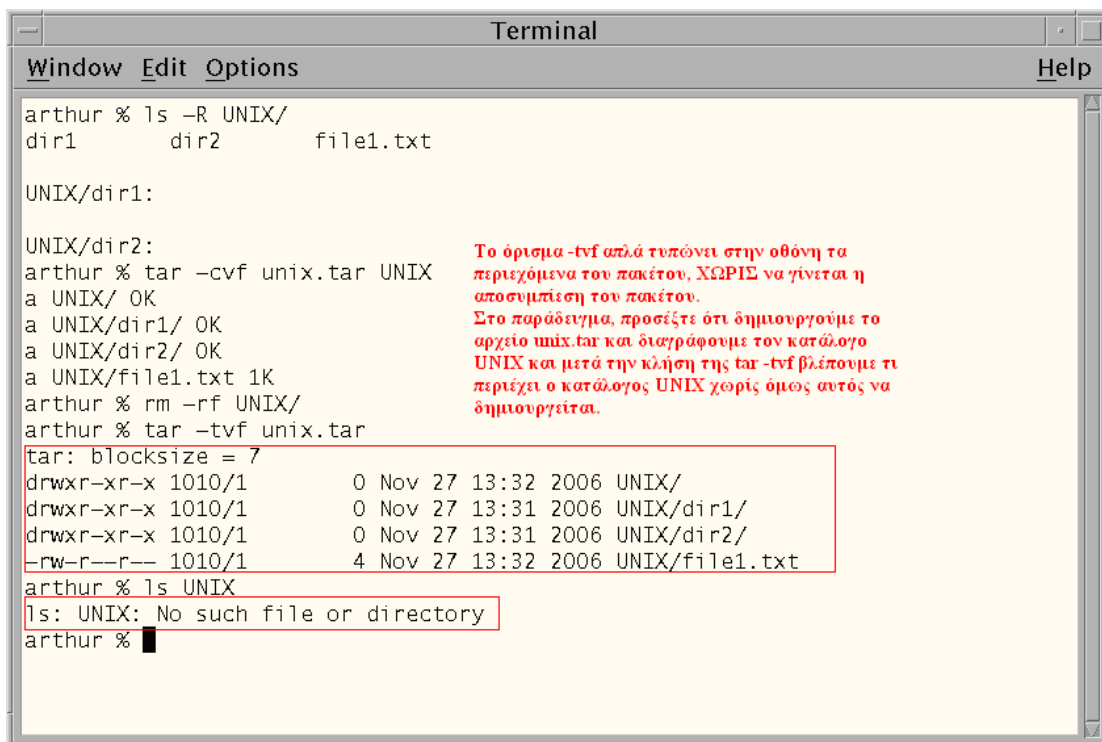
Εικόνα 50: Συμπίεση – αποσυμπίεση αρχείων.

Η εντολή `tar`

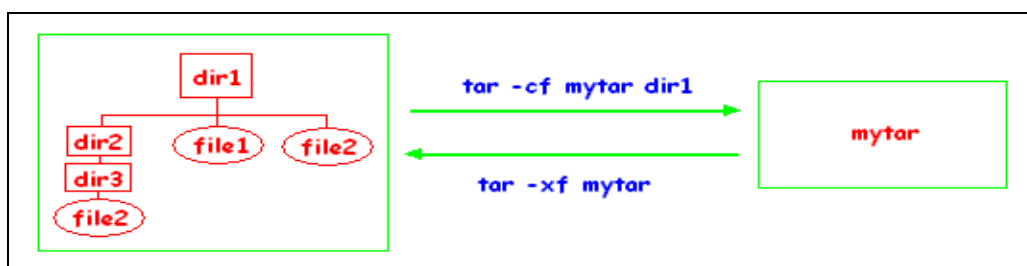
Συνεχίζοντας, υπάρχει ένα εργαλείο, η εντολή `tar`, η οποία και επιτρέπει να πακετάρουμε ένα δέντρο αρχείων, όπως το περιγράψαμε και σε προηγούμενα εργαστήρια. Το πακέτο αυτό είναι ένα αρχείο, το οποίο και μπορούμε να μεταφέρουμε οπουδήποτε, να το ξεπακετάρουμε και να πάρουμε πίσω το δέντρο όπως ακριβώς ήταν στην αρχή. Στις δύο εικόνες που ακολουθούν περιγράφεται η χρήση της εντολής `tar` και με ένα σχηματικό τρόπο απεικονίζονται τα αποτελέσματα της `tar`.



Εικόνα 51: Η εντολή tar.



Εικόνα 52: Η εντολή tar (συνέχεια).



Εικόνα 53: Σχηματική απεικόνιση της tar στο παράδειγμα της εικόνας 4.

## 2.7 Δικαιώματα χρήσης αρχείων – chmod

Το Unix υποστηρίζει ένα σύστημα δικαιωμάτων για την προστασία των αρχείων. Τα δικαιώματα ουσιαστικά είναι ο μηχανισμός με τον οποίο το Unix καθορίζει τι μπορεί να κάνει ο οποιοσδήποτε σε ένα αρχείο. Όταν λέμε «ο οποιοσδήποτε», εννοούμε τρεις κατηγορίες χρηστών:

1. ο ίδιος ο χρήστης
2. οι υπόλοιποι χρήστες που ανήκουν στην ίδια ομάδα εργασίας με τον χρήστη
3. οι χρήστες των υπολοίπων ομάδων

Τα δικαιώματα που υποστηρίζονται είναι τριών ειδών:

1. *ανάγνωσης (read)*. Αν σε ένα αρχείο έχει παραχωρηθεί αυτό το δικαίωμα, τότε η αντίστοιχη κατηγορία μπορεί να δει τα περιεχόμενα του αρχείου MONO, αλλά δεν μπορεί να τα αλλάξει.
2. *εγγραφής (write)*. Η παραχώρηση αυτού του δικαιώματος σημαίνει ότι η αντίστοιχη κατηγορία μπορεί να τροποποιήσει τα δεδομένα.
3. *εκτέλεσης (execute)*. Δικαίωμα εκτέλεσης σημαίνει ότι το αρχείο μπορεί να εκτελεστεί, αν είναι εκτελέσιμο. Στην περίπτωση καταλόγων η παραχώρηση του δικαιώματος αυτού είναι απαραίτητη αν θέλουμε να επιτρέπουμε σε κάποιον να μπορεί να εισέλθει στον κατάλογο (με την εντολή cd).

Για κάθε μία από τις προαναφερθείσες κατηγορίες χρηστών θα πρέπει να ορίζεται χωριστά τι δικαιώματα θα έχουν επί ενός αρχείου. Όπως είναι αυτονόητο, τα δικαιώματα τους θα είναι συνδυασμός των παραπάνω.

### Η εντολή chmod

Το βοήθημα που διαθέτει το Unix για τον προσδιορισμό δικαιωμάτων είναι η εντολή chmod. Η λειτουργία της δίνεται στις παρακάτω εικόνες.

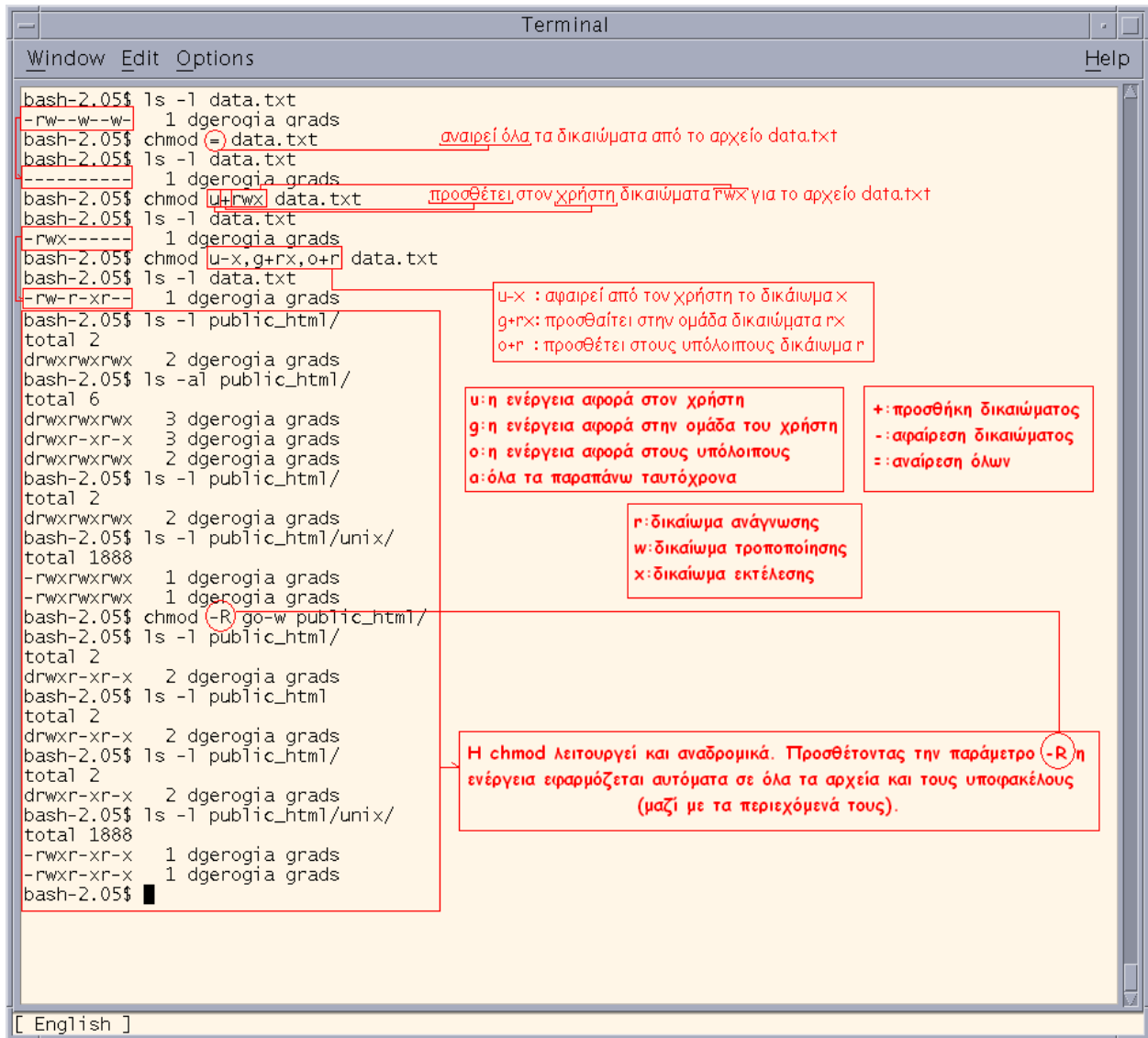
```
Terminal
Window Edit Options Help
bash-2.05$ ls -l data.txt
-rw-r--r--  1 dgerogia grads          4 Noé  22 13:03 data.txt
bash-2.05$ chmod 755 data.txt
bash-2.05$ ls -l data.txt
-rwxr-xr-x  1 dgerogia grads          4 Noé  22 13:03 data.txt
bash-2.05$ chmod 600 data.txt
bash-2.05$ ls -l data.txt
-rw-----  1 dgerogia grads          4 Noé  22 13:03 data.txt
bash-2.05$ █
```

Στην chmod αντί για τα αλφαριθμητικά (u,g,o,a κ.τ.λ.) μπορούμε να καθορίσουμε δικαιώματα με αριθμητικές τιμές που κωδικοποιούνται στα επιθυμητά δικαιώματα που θέλουμε να δώσουμε σε κάποιο αρχείο. Οι αριθμητικές τιμές είναι τριψήφιες, όπως φαίνεται και παραπάνω. Το πρώτο ψηφίο αναφέρεται στον χρήστη, το δεύτερο στην ομάδα του χρήστη και το τρίτο στους υπόλοιπους. Για τον υπολογισμό των ψηφίων:

1. Σε κάθε δικαίωμα αντιστοιχά μία τιμή: r=4, w=2, x=1  
Το μηδέν (0) σημαίνει ανάρτηση δικαιωμάτων (αντίστοιχο με το =)
2. Για κάθε δικαίωμα που θέλω να εκχωρήσω, προσθέτω την αντίστοιχη τιμή  
π.χ. για να εκχωρήσω r,w: (r=4)+(w=2)=6
3. Με τον τρόπο αυτό φτιάχνω και τα 3 ψηφία  
π.χ 6 στην πρώτη θέση σημαίνει rw για τον χρήστη κ.τ.λ., όπως έχουμε πει και παραπάνω για τη σημασία των θέσεων.

[ English ]

Εικόνα 54: Η εντολή chmod – Παράδειγμα 1.



Εικόνα 55: Η εντολή chmod – Παράδειγμα 1.

## 2.8 Ανακατεύθυνση εισόδου /εξόδου

Πολλές από τις εντολές του Unix είτε περιμένουν από τον χρήστη να πληκτρολογήσει κάποια δεδομένα ως είσοδο, είτε τυπώνουν το αποτέλεσμα στην οθόνη. Τα δύο αυτά μέσα εισόδου (= πληκτρολόγιο) και εξόδου (=οθόνη) ονομάζονται αντίστοιχα πρότυπη είσοδος και πρότυπη έξοδος. Υπάρχει όμως η δυνατότητα να καθορίσουμε διαφορετικά μέσα που αντιστοιχούν είτε στην πρότυπη έξοδο, είτε στην πρότυπη είσοδο, είτε και στα δύο. Η τακτική αυτή ονομάζεται ανακατεύθυνση εισόδου, εξόδου, εισόδου-εξόδου αντίστοιχα. Ουσιαστικά με την ανακατεύθυνση ορίζουμε ως μέσο εισόδου ή εξόδου ένα αρχείο. Έτσι στην περίπτωση της ανακατεύθυνσης εισόδου, το Unix αντί να περιμένει το χρήστη να πληκτρολογήσει δεδομένα, διαβάζει τα δεδομένα του αρχείου στο οποίο ανακατευθύνουμε την είσοδο και με αυτά τροφοδοτεί το πρόγραμμα το οποίο περιμένει είσοδο. Η διαδικασία είναι ίδια με το να πληκτρολογούσε ο χρήστης τα δεδομένα του αρχείου όταν θα έτρεχε κάποιο πρόγραμμα. Αντίστοιχα στην ανακατεύθυνση εξόδου, τα δεδομένα αντί να τυπωθούν στην οθόνη, αποθηκεύονται στο αρχείο στο οποίο ανακατευθύνουμε την έξοδο. Η ανακατεύθυνση εισόδου γίνεται με τον τελεστή < και της εξόδου με > ή >>, όπως φαίνεται στο παράδειγμα στις εικόνες που ακολουθούν. Το test είναι ένα πρόγραμμα που έχουμε φτιάξει και απλά περιμένει από τον χρήστη να γράψει δύο λέξεις με το πληκτρολόγιο για να τις αποθηκεύσει στο αρχείο data.txt

```
Terminal
Window Edit Options Help
bash-2.05$ ./test
123
abc
bash-2.05$ cat data.txt
Line 1 is: "123"
Line 2 is: "abc"
bash-2.05$ cat input
UNIX
HTML
bash-2.05$ ./test < input
bash-2.05$ cat data.txt
Line 1 is: "UNIX"
Line 2 is: "HTML"
bash-2.05$
```

Το πρόγραμμα test διαβάζει τα δεδομένα που πληκτρολογεί ο χρήστης και τα αποθηκεύει σε ένα αρχείο όπως φαίνεται δίπλα:  
Line ... is: "...ότι έγραψε ο χρήστης..."

Τα δεδομένα του αρχείου input περνάνε ως είσοδο στη πρόγραμμα test.

Ο τελεστής < είναι η ανακατεύθυνση της εισόδου. Αυτό σημαίνει ότι αν ένα πρόγραμμα περιμένει από το χρήστη είσοδο από το πληκρολόγιο, τότε στην ανακατεύθυνση θα πάρει ως είσοδο τα δεδομένα ενός αρχείου σαν να τα πληκτρολογούσε ο χρήστης.

Εικόνα 57:Ανακατεύθυνση εισόδου.



```
Terminal
Window Edit Options Help
bash-2.05$ cal 11 2006
November 2006
S M Tu W Th F S
      1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30

bash-2.05$ cal 11 2006 > data.txt
bash-2.05$ cat data.txt
November 2006
S M Tu W Th F S
      1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30

bash-2.05$ cal 11 2007 >> data.txt
bash-2.05$ cat data.txt
November 2006
S M Tu W Th F S
      1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30

November 2007
S M Tu W Th F S
      1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30

bash-2.05$ cal 11 2007 > data.txt
bash-2.05$ cat data.txt
November 2007
S M Tu W Th F S
      1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30

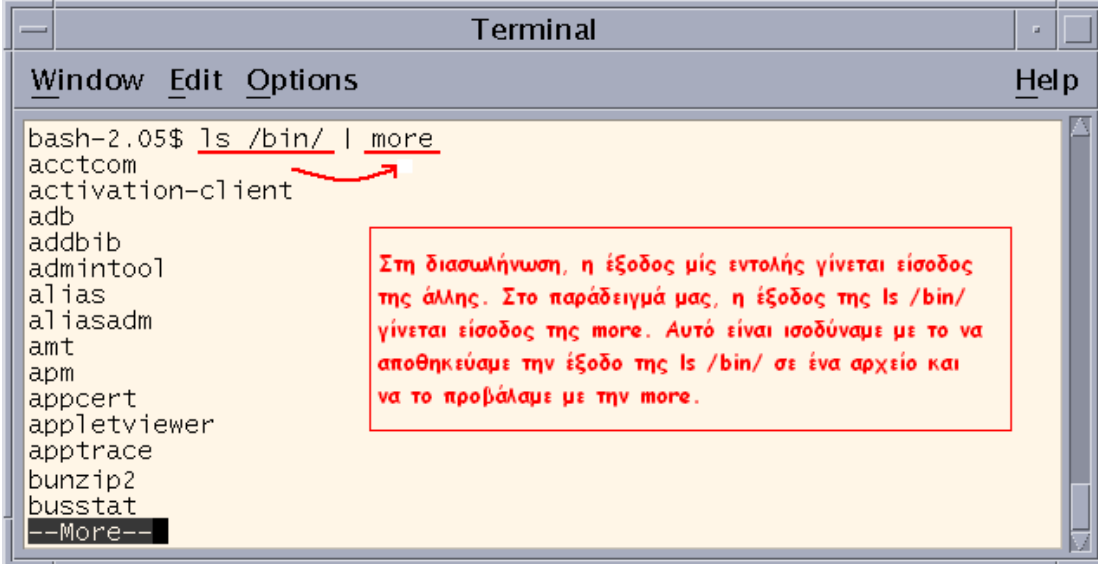
bash-2.05$ █
```

**Ανακατεύθυνση εξόδου με > και >>.**  
**Με > τα δεδομένα που θα τύπωνε στην οθόνη η εντολή αντιγράφονται σε αρχείο, αντικαθιστώντας το περιεχόμενο.**  
**Με >> τα δεδομένα που θα τύπωνε στην οθόνη η εντολή αντιγράφονται στο τέλος του αρχείου, διατηρώντας το περιεχόμενο.**

Εικόνα 58:Ανακατεύθυνση εξόδου.

## 2.9 Διασωλήνωση εντολών

Θα κλείσουμε τον οδηγό με την επεξήγηση της έννοιας της διασωλήνωσης εντολών. Ουσιαστικά η διασωλήνωση συνοψίζεται στο να μπορούμε να εκτελούμε σειριακά (δηλαδή τη μία μετά την άλλη) εντολές έτσι ώστε η έξοδος της μίας, αντί να τυπώνεται στην οθόνη, να περνά ως είσοδος της επόμενης κ.ο.κ. Η έννοια της διασωλήνωσης περικλείει και την ανακατεύθυνση εισόδου.



```
Terminal
Window Edit Options Help
bash-2.05$ ls /bin/ | more
acctcom
activation-client
adb
addbib
admintool
alias
aliasadm
amt
apm
appcert
appletviewer
apptrace
bunzip2
busstat
--More--
```

Στη διασωλήνωση, η έξοδος μίας εντολής γίνεται είσοδος της άλλης. Στο παράδειγμά μας, η έξοδος της `ls /bin/` γίνεται είσοδος της `more`. Αυτό είναι ισοδύναμο με το να αποθηκεύαμε την έξοδο της `ls /bin/` σε ένα αρχείο και να το προβάλαμε με την `more`.

Εικόνα 59: Διασωλήνωση δύο εντολών.