



ΕΠΛ232 – Προγραμματιστικές Τεχνικές και Εργαλεία

Διάλεξη 20: Εργαλεία UNIX I (Κεφάλαια 1-4 & 7, DAS-2ED)

Δημήτρης Ζεϊναλιπούρ

<http://www.cs.ucy.ac.cy/courses/EPL232>



Περιεχόμενο Διάλεξης

- **Χειρισμός Καταλόγων/Αρχείων** (*mkdir, rmdir, cp, mv, rm*),
- **Εξέταση Αρχείων** (*file, head, tail, cat, more, wc, sort, uniq*),
- **Μετατροπή Αρχείων** (*iconv, unix2dos/dos2unix*)
- **Ρεύματα Εισόδου/Εξόδου** (*stdin, stdout, stderr*), Ανακατεύθυνση Αρχείων (*Redirection*), Διοχέτευση/Σωλήνωση (*Piping*).

Χειρισμός Καταλόγων/Αρχείων



- **Δημιουργία Καταλόγων**

- Εντολή ***mkdir*** *<directory>*

- *directory*: σχετικό ή απόλυτο μονοπάτι
- Αργότερα θα μιλήσουμε και για θέματα δικαιωμάτων στους καταλόγους που δημιουργούνται.

- **Διαγραφή Καταλόγων**

- Εντολή ***rmdir*** *<directory>*

- Ο κατάλογος **πρέπει να είναι κενός** (για να μην μένουν τα αρχεία σε αυτούς τους καταλόγους ορφανά), εναλλακτικά ***rm -rf <folder>***
 - Με εντολές διαγραφής είναι γενικά καλό να χρησιμοποιείται μόνο απλές επιλογές για να μειώσετε τα ενδεχόμενα λάθη.

- *-p, --parents* :Remove DIRECTORY and its ancestors. E.g., ``rmdir -p a/b/c'` is similar to ``rmdir a/b/c a/b a'`. Also `mkdir -p /a/b/c`

20-3

Χειρισμός Καταλόγων/Αρχείων



- **Αντιγραφή αρχείων**

- Εντολή **cp** (επιλογές **-i**, **-r**, **-p**)
- Δημιουργεί ένα αντίγραφο ενός αρχείου (ή καταλόγου)
- Χρήση εντολής: **cp** **<Source>** **<Destination>**
- Επιλογή **-i (interactive)**
 - Με αυτό τον τρόπο θα ερωτηθούμε για να επικυρώσουμε τη **υπερκάλυψη (override)** κάποιου destination αρχείου. (μπορεί να τοποθετηθεί στο .profile ως alias για να ενεργοποιείται πάντα ... θα δούμε αργότερα πως)

```
$ cp hello2.txt hello.txt
$ cp -i hello2.txt hello.txt
cp: overwrite `hello.txt'?
```
- Επιλογή **-r (recursive)**
 - *Αναδρομικά*, αντιγράφει όλα τα αρχεία και υποκαταλόγους
- Επιλογή **-p (permissions)**: διατηρεί i-node πεδία και τα δικαιώματα πρόσβασης (π.χ., cp web folder).

20-4

Χειρισμός Καταλόγων/Αρχείων



- **Μεταφορά αρχείων**

- Εντολή *mv* (επιλογή *-i*)

- Μεταφέρει ένα αρχείο από μια θέση, στην ιεραρχία του υποσυστήματος αρχείων, σε μια νέα

- Χρήση εντολής: *mv* <Source> <Destination>

- Επιλογή *-i*

- Σε ρωτά αν θέλεις να γράψει πάνω σ' ένα αρχείο που ήδη υπάρχει.

- Χρησιμοποιείται και ως *μετονομασία αρχείων*

20-5

Χειρισμός Καταλόγων/Αρχείων



- Διαγραφή αρχείων
 - Εντολή *rm* (επιλογές *-i*, *-f*, *-r*)
 - Διαγράφει ένα αρχείο
 - Χρήση εντολής: *rm* <filename(s)>
 - Επιλογή *-i*
 - Επαλήθευση διαγραφής από χρήστη
 - Επιλογή *-f*
 - Υποχρεώνει (Force) τη διαγραφή του αρχείου
 - υπερισχύει της επιλογής *-i*
 - Επιλογή *-r*
 - *Αναδρομικά*, διαγράφει όλα τα αρχεία και υποκαταλόγους



Εξέταση Αρχείων

- Τύποι αρχείων
 - Διαφορετικά αρχεία έχουν διαφορετική χρήση
 - Πώς γνωρίζεις τι τύπου είναι το αρχείο;
- Εντολή *file*
 - αναγνώριση τύπων αρχείων
 - Βασίζεται στην **ανάλυση των πρώτων λίγων bytes του κειμένου** όπου περιέχεται μια **ακολουθία (magic number)** η οποία είναι μοναδική για κάθε τύπο εμπορικού αρχείου.



Εξέταση Αρχείων

```
bash-3.1$ file test/  
test/: directory  
bash-3.1$ file test/test1/test1.txt  
test/test1/test1.txt: ASCII text  
bash-3.1$ file test.c  
test.c: ASCII C program text  
bash-3.1$ file tes  
tes: executable (RISC System/6000  
    V3.1) or obj module not stripped  
bash-3.1$
```


Εξέταση Αρχείων

Μηχανισμός Αναγνώρισης



[http://en.wikipedia.org/wiki/Magic_number_\(programming\)](http://en.wikipedia.org/wiki/Magic_number_(programming))

- [Compiled Java](#) class files ([bytecode](#)) start with 0xCAFEBAFE on [big-endian](#) systems.
- [GIF](#) image files have the [ASCII](#) code for 'GIF89a' (0x474946383961) or 'GIF87a' (0x474946383761)
- [JPEG](#) image files begin with 0xFFD8FF, and JPEG/[JFIF](#) files contain the [ASCII](#) code for 'JFIF' (0x4A464946) or JPEG/[EXIF](#) files contain the [ASCII](#) code for 'Exif' (0x45786966) beginning at byte 6 in the file, followed by more [metadata](#) about the file.
- [PNG](#) image files begin with an 8-[byte](#) signature which identifies the file as a PNG file and allows immediate detection of some common file-transfer problems: \211 P N G \r \n \032 \n (0x89504e470d0a1a0a)
- Standard [MIDI](#) music files have the [ASCII](#) code for 'MThd' (0x4D546864) followed by more metadata about the file.
- Unix script files usually start with a [shebang](#), '#!' (0x2321, or 0x2123 on [little-endian](#) processors) followed by the path to an interpreter.
- [PostScript](#) files and programs start with '%!' (0x2521).
- [PDF](#) files start with '%PDF'.
- Old [MS-DOS](#) .exe files and the newer [Microsoft Windows PE](#) (Portable Executable) .exe files start with the [ASCII](#) string 'MZ' (0x4D5A), the initials of the designer of the file format, [Mark Zbikowski](#). The definition allows 'ZM' as well but it is quite uncommon.
- The [Berkeley Fast File System](#) superblock format is identified as either 0x19540119 or 0x011954 depending on version; both represent the birthday of author [Marshall Kirk McKusick](#).
- Executables for the [Game Boy](#) and [Game Boy Advance](#) handheld video game systems have a 48-byte or 156-byte magic number, respectively, at a fixed spot in the header. This magic number encodes a bitmap of the [Nintendo](#) logo.
- [Zip](#) files begin with 'PK', the initials of [Phil Katz](#), author of [DOS](#) compression utility [PKZIP](#).
- Old [Fat binaries](#) (containing code for both [68K](#) processors and [PowerPC](#) processors) on Classic [Mac OS](#) contained the [ASCII](#) code for 'Joy!' (0x4A6F7921) as a prefix.
- [TIFF](#) files begin with either "II" or "MM" depending on the byte order (II for Intel, or little endian, MM for Motorola, or big endian), followed by 0x2A00 or 0x002A (decimal [42](#) as a 2-byte integer in Intel or Motorola byte ordering).
- [Unicode](#) text files, encoded with [UTF-8](#), often start with \xEF\xBB\xBF (i»¿ in a [Latin-1](#) text editor), the UTF-8 encoding of the [Byte Order Mark](#). If they are encoded with [UTF-16](#), they often start with the [Byte Order Mark](#) coded as \xFE\xFF or \xFF\xFE depending on [endianness](#).



Εξέταση Αρχείων

- Εντολή *head* (επιλογή *-n*)
 - Εμφάνιση αρχικών γραμμών αρχείων
 - 10 γραμμές εξ' ορισμού
 - Προσφέρει ένα γρήγορο τρόπο να δεις εάν είναι αυτό το αρχείο που γυρεύεις.
 - Χρήση: *head -n filename*
 - Τυπώνει τις πρώτες *n* γραμμές



Εξέταση Αρχείων

```
bash-3.1$ head -8 test-math.c
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
int main()
```

```
{
```

```
    int a, b;
```

```
bash-3.1$
```



Εξέταση Αρχείων

- Εντολή *tail* (επιλογή *-n*)
 - Εμφάνιση τελικών γραμμών αρχείων
 - 10 γραμμές εξ' ορισμού
 - Προσφέρει ένα γρήγορο τρόπο να δεις εάν είναι αυτό το αρχείο που γυρεύεις από το τέλος.
 - Χρήση: *tail -n filename*
 - Τυπώνει τις τελευταίες *n* γραμμές



Εξέταση Αρχείων

```
bash-3.1$ tail -7 test-math.c
```

```
c = sin(0);
```

```
printf("a= %d\nb= %d\nc= %f\n", a, b, c);
```

```
return 0;
```

```
}
```

```
bash-3.1$
```



Εξέταση Αρχείων

- Εντολή *cat* (επιλογές *-s*, *-n*)
 - *cat* → *concatenate* (συνένωση)
 - εμφάνιση πολλών αρχείων, το ένα μετά το άλλο
 - *cat* *<filename1>* *<filename2>* ...
 - Επιλογή *-s*
 - Συμπύσσει (*suppress*) πολλαπλές κενές γραμμές σε μια μόνο
 - Επιλογή *-n*
 - προσθέτει αρίθμηση γραμμών στην έξοδο (όχι στο ίδιο το αρχείο).



Εξέταση Αρχείων

```
bash-3.1$ cat -ns test-math.c
 1  #include <stdio.h>
 2  #include <math.h>
 3
 4  int main()
 5  {
 6
 7      int a, b;
 8      float c;
 9
10      a = -2;
11
12      b = abs(a);
13
14      c = sin(0);
15
16      printf("a= %d\nb= %d\nc= %f\n", a,b, c);
17
18      return 0;
19  }
bash-3.1$
```



Εξέταση Αρχείων

- Εάν το αρχείο είναι αρκετά μεγάλο, θα «κυλίσει» στην οθόνη πολύ γρήγορα για να το διαβάσουμε.
- Για να διαβάσουμε το αρχείο μπορούμε να εκτελέσουμε μια σωλήνωση με την εντολή `more`
 - `cat a b c | more` (θα την δούμε σε πολύ λίγο)
- Εάν θέλουμε να σταματήσουμε την εντολή τότε στέλνουμε το σήμα
 - *Ctrl-C (interrupt, sig2), το οποίο θα σταματήσει την εκτέλεση της εντολής.*



Εξέταση Αρχείων

- Εντολή *more* (επιλογές *-s*, *-d*)
 - Εμφάνιση του αρχείου σελίδα-σελίδα στην οθόνη
 - Χρήση: *more* <filename>
 - Επιλογή *-s* (*suppress*)
 - Συμπύσσει (suppress) πολλαπλές κενές γραμμές σε μια μόνο
 - Επιλογή *-d*
 - εμφανίζει φιλικά μηνύματα υπαγόρευσης στο τέρμα της κάθε σελίδας
 - --More--[Press space to continue, 'q' to quit.]



Εξέταση Αρχείων

- Κινήσεις μέσα στο “ψευδό-κέλυφος” εκτέλεσης της *more*:
 - ***spacebar***
 - κυλά το αρχείο στην επόμενη σελίδα
 - ***return***
 - κινείται κατά μια γραμμή
 - ***n return***
 - εμφανίζει τις επόμενες *n* γραμμές
 - ***q***
 - τερματισμός της εντολής *more* χωρίς να έχει τελειώσει



Εξέταση Αρχείων

```
bash-3.1$ more -d test.c
```

```
#include<stdio.h>
```

```
#define TAB '\t'
```

```
#define SPACE ' '
```

```
#define NLN '\n'
```

```
#define BACKSP '\b'
```

```
int check_char(int flug);
```

```
void main(void)
```

```
{ int ch1,flug=-1,counter=0,n_lines=0,total_ch=0;
```

```
float average;
```

```
printf("\nType anything you like and '@' to QUIT\n");
```

```
ch1=getchar();
```

```
while(ch1!='@')
```

```
{ flug=check_char(ch1);
```

```
if (flug==1) {
```

```
if (ch1==NLN) {
```

```
total_ch++; n_lines++;
```

```
printf("\nNumber of characters in this line = %d\n",counter);
```

```
--More--(44%)[Press space to continue, 'q' to quit.]
```



Εξέταση Αρχείων

- Εντολή **wc** (επιλογές **-l**, **-w**, **-c**)
 - Εκθέτει τον αριθμό των **γραμμών (line)**, **λέξεων (word)** και **χαρακτήρων (char)** ενός αρχείου
 - Χρήση: **wc** *<filename>*

```
bash-3.1$ wc test-math.c  
20 30 174 test-math.c
```

```
bash-3.1$
```



Εξέταση Αρχείων

– Επιλογή *-l*

- εκθέτει μόνο τον αριθμό των γραμμών

```
bash-3.1$ wc -l test-math.c  
20 test-math.c
```

– Επιλογή *-w*

- εκθέτει μόνο τον αριθμό των λέξεων

```
bash-3.1$ wc -w test-math.c  
30 test-math.c
```

– Επιλογή *-c*

- εκθέτει μόνο τον αριθμό των χαρακτήρων

```
bash-3.1$ wc -c test-math.c  
174 test-math.c
```



Εξέταση Αρχείων

- Εντολή **sort** (επιλογές *-r*, *-n*, *-k*)
 - Ταξινόμηση των γραμμών αρχείου σε αλφαβητική σειρά
 - Χρήση: **sort** [επιλογές] <filename>
 - Είναι χρήσιμη για αρχεία δεδομένων ή έξοδο από προγράμματα



Εξέταση Αρχείων

- Επιλογή ***-r (reverse)***
 - ταξινόμηση σε φθίνουσα σειρά (larges=>smallest)
- Επιλογή ***-n (sort numeric)***
 - ταξινόμηση με αριθμητική σειρά
- Επιλογή ***-k [number]***
 - ταξινόμηση με βάση τη στήλη *#number*
 - *Ακολουθεί παράδειγμα*



Εξέταση Αρχείων

```
bash-3.1$ cat test-sort.txt
```

```
john      32      london
```

```
jean      7       paris
```

```
marco    21      rome
```

```
anna     18      nicosia
```

```
antonio  58      madrid
```

```
peter    4       chicago
```

```
luis     1       geneva
```

```
filip    18      brussels
```




Εξέταση Αρχείων

```
bash-3.1$ sort -r test-sort.txt
```

```
peter      4      chicago  
marco     21      rome  
luis      1      geneva  
john     32      london  
jean      7      paris  
filip    18      brussels  
antonio  58      madrid  
anna     18      nicosia
```



Εξέταση Αρχείων

```
bash-3.1$ sort -nk 2 test-sort.txt  
luis      1      geneva  
peter    4      chicago  
jean     7      paris  
anna    18     nicosia  
filip   18     brussels  
marco   21     rome  
john    32     london  
antonio 58     madrid
```



Εξέταση Αρχείων

```
bash-3.1$ sort -k 3 test-sort.txt  
filip      18      brussels  
peter      4       chicago  
luis       1       geneva  
john       32      london  
antonio    58      madrid  
anna       18      nicosia  
jean       7       paris  
marco     21      rome
```



Εξέταση Αρχείων

- Εντολή **uniq** (επιλογή -c)
 - Αφαιρεί *συνεχόμενα* διπλά αντίγραφα γραμμών (**διπλότυπα**)
 - Γιατί πρέπει να είναι *συνεχόμενα*;
 - Χρήση: **uniq** [επιλογές] <filename>
 - Επιλογή **-c (count)**
 - Εκθέτει κάθε εμφάνιση με τον αριθμό των *διπλότυπων* γραμμών.
 - Επιλογή **-u (unique)**
 - *Τυπώνει μόνο τις γραμμές που εμφανίζονται μια φορά.*



Εξέταση Αρχείων

```
bash-3.1$ cat test-uniq.test
```

```
Dan
```

```
Dan
```

```
Dan
```

```
Athena
```

```
Ryu
```

```
Ryu
```

```
Athena
```

```
Nick
```

```
bash-3.1$ uniq test-uniq.test
```

```
Dan
```

```
Athena
```

```
Ryu
```

```
Athena
```

```
Nick
```

```
bash-3.1$ uniq -c test-uniq.test
```

```
3 Dan
```

```
1 Athena
```

```
2 Ryu
```

```
1 Athena
```

```
1 Nick
```

Μετατροπή Συνόλου Χαρακτήρων (Character Set)



- Εντολή *iconv -f <from_enc> -t <to_enc>*
 - Μετατρέπει το character set ενός αρχείου από το ένα **from** πρότυπο στο **to** πρότυπο.

- *Εξαιρετικά χρήσιμο για ανάλυση παλαιών αρχείων που δεν είναι UTF8 (Unicode)!, π.χ.,*

- *iconv -f ISO-8859-1 -t UTF-8 file.txt*

Μετατροπή από παλαιά κωδικοποίηση West-European #7 (ISO/IEC 8859-7) σε Unicode.

Μετατροπή Συνόλου Χαρακτήρων (Character Set)



```
# Παρουσίαση Προβλήματος!  
$ cat names-8859-7.txt  
1 1 ????? 6652 334686 342632 350579  
2 2 ????? 3850 192168 198306 204445  
3 1 ???????? 3320 165291 171007 176724  
4 3 ?????????? 2770 137441 142678 147915
```

Εάν ανοίγατε το αρχείο με κειμενογράφο ή web browser που υποστηρίζει αυτό το charset τότε τα ? θα μετατρέπονταν σε αναγνώσιμους χαρακτήρες. Αλλά για μετατροπή μεγάλων αρχείων αυτή η προσέγγιση δεν είναι αποδοτική

```
# Εύρεση του Τύπου Συνόλου Χαρακτήρων
```

```
$ file names-8859-7.txt
```

```
names-8859-7.txt: ISO-8859 text, with CRLF line terminators
```

```
# Μετατροπή του Τύπου Συνόλου Χαρακτήρων
```

```
$ iconv -f ISO-8859-7 -t UTF-8 names-8859-7.txt > names-UTF8.txt
```

```
$ cat names-UTF8.txt
```

```
1 1 ΜΑΡΙΑ 6652 334686 342632 350579  
2 2 ΕΛΕΝΗ 3850 192168 198306 204445  
3 1 ΓΕΩΡΓΙΟΣ 3320 165291 171007 176724  
4 3 ΑΙΚΑΤΕΡΙΝΗ 2770 137441 142678 147915
```

20-31

Μετατροπή Συνόλου Χαρακτήρων (Character Set)



```
# Παρουσίαση υποστηριζόμενων συνόλων χαρακτήρων  
$ iconv -l
```

```
ANSI_X3.4-1968 ANSI_X3.4-1986 ASCII CP367 IBM367 ISO-IR-6 ISO646-US  
ISO_646.IRV:1991 US US-ASCII CSASCII  
UTF-8  
UTF-8-MAC UTF8-MAC  
ISO-10646-UCS-2 UCS-2 CSUNICODE  
UCS-2BE UNICODE-1-1 UNICODEBIG CSUNICODE11  
UCS-2LE UNICODELITTLE  
ISO-10646-UCS-4 UCS-4 CSUCS4  
UCS-4BE  
UCS-4LE  
UTF-16  
UTF-16BE  
UTF-16LE  
UTF-32  
UTF-32BE  
UTF-32LE  
ISO-8859-16 ISO-IR-226 ISO8859-16 ISO_8859-16 ISO_8859-16:2001
```


Διαχείριση Αρχείων XML / JSON (xmllint, jq)



- Στην εποχή των ανοικτών δεδομένων (Open Data) διατίθενται πλέον στον ιστό σωρεία δεδομένων προς κατανάλωση, π.χ.,
 - π.χ., δεδομένα κλινικών δοκιμών από το <https://clinicaltrials.gov/> διαθέτει δεδομένα σε XML
 - Wikidata.org διαθέτει μια XML έκδοση της Wikipedia σε XML.
 - Οι πλείστες Web 2.0 υπηρεσίες (π.χ., Google, FB, Twitter, κτλ.) παρέχουν JSON APIs τα οποία επιτρέπουν την προσπέλαση σε JSON (lightweight XML) δεδομένα σε συνεχόμενη βάση
- Τι είδους εργαλεία χρειαζόμαστε για να επεξεργαστούμε γρήγορα τέτοια δεδομένα;

Διαχείριση Αρχείων XML / JSON (xmllint, jq)



```
# Παρουσίαση περιεχομένου XML  
$ xmllint --format 3178056.xml
```

```
<ref id="B72">  
  <label>72</label>  
  <element-citation publication-type="journal">  
    <person-group person-group-type="author">  
      <name>  
        <surname>Price</surname>  
        <given-names>MN</given-names>  
      </name>  
      <name>  
        <surname>Dehal</surname>  
        <given-names>PS</given-names>  
      </name>  
      <name>  
        <surname>Arkin</surname>  
        <given-names>AP</given-names>  
      </name>
```

Διαχείριση Αρχείων XML / JSON (xmllint, jq)



Ανάκτηση και μορφοποίηση περιεχομένου JSON

\$ **curl -s**

```
'http://api.nytimes.com/svc/elections/us/v3/finances/2008/president/tals.json?api-key=super-secret' | jq '.' | head
```

```
{  
  "results": [  
    {  
      "candidate_id": "P80003338",  
      "date_coverage_from": "2007-01-01",  
      "date_coverage_to": "2008-11-24",  
      "candidate_name": "Obama, Barack",  
      "name": "Barack Obama",  
      "party": "D",
```

Επιπλέον Εργαλεία για Data Science:

- **json2csv - convert JSON to CSV | xml2json - convert XML to JSON**
- **csvkit - suite of utilities for converting to and working with CSV**

20-35



Ρεύματα Εισόδου/Εξόδου

- Όταν εκτελέσουμε ένα πρόγραμμα ανοίγουν τρία προσωπικά αρχεία για αυτό το αρχείο τα οποία είναι ως ακολούθως:
- ***stdin***
 - Είναι το προκαθορισμένο ρεύμα για **παραλαβή εισόδου**, όταν ένα πρόγραμμα τρέχει.
 - default: είσοδος από το πληκτρολόγιο
- ***stdout***
 - Είναι το προκαθορισμένο ρεύμα για **αποστολή εξόδου**, όταν ένα πρόγραμμα τρέχει.
 - default: έξοδος στην οθόνη
- ***stderr***
 - Είναι το προκαθορισμένο ρεύμα για αποστολή εξόδου **διαγνωστικών μηνυμάτων**, όταν ένα πρόγραμμα τρέχει.
 - default: έξοδος – μηνύματα λάθους στην οθόνη

Ανακατεύθυνση Αρχείων



- Χρήση ειδικών τελεστών του UNIX για διάβασμα (*read*), γράψιμο (*write*) και επισύναψη (*append*) δεδομένων από/σε αρχείο.
 - < **ανακατεύθυνση εισόδου** από αρχείο
το αρχείο πρέπει να υπάρχει.
 - > **ανακατεύθυνση εξόδου** σε αρχείο
αν δεν υπάρχει το αρχείο δημιουργείται,
αλλιώς γράφεται πάνω από το υπάρχων.
 - >> **ανακατεύθυνση εξόδου και
επισύναψη** των δεδομένων στο
υπάρχων αρχείο (*append*)
αν δεν υπάρχει το αρχείο δημιουργείται,
αλλιώς συνεχίζει στο υπάρχων.

Ανακατεύθυνση Αρχείων



- Είναι χρήσιμη η ανακατεύθυνση;
 - Όποτε θέλεις να σώσεις την έξοδο ενός προγράμματος.
 - Όποτε θέλεις να αποφύγεις να γράφεις την είσοδο συνέχεια.
 - Όποτε θέλεις να απομονώσεις διαγνωστικά μηνύματα (λάθους).
- Σημείωση στην ανακατεύθυνση:
 - 0 → stdin
 - 1 → stdout
 - 2 → stderr

Ανακατεύθυνση Αρχείων



```
bash-3.1$ ls test/ /bin/DoesNotExist
ls: /bin/DoesNotExist: No such file or directory
test/:
ErrorOutput HardLinkToFile1.txt
  SymbLinkToFile2.txt test1 test2
bash-3.1$ ls test/ /bin/DoesNotExist 2>
  test/ErrorOutput
test/:
ErrorOutput HardLinkToFile1.txt
  SymbLinkToFile2.txt test1 test2
bash-3.1$ cat test/ErrorOutput
ls: /bin/DoesNotExist: No such file or directory
```

```
./foo < InputFile 1>> OutputFile 2>> ErrorFile
```



Διοχέτευση/Σωλήνωση

- **Διοχέτευση (*pipe*)** επιτρέπει τη σύνδεση της εξόδου ενός προγράμματος στην είσοδο ενός άλλου.

– Επιτυγχάνεται με τη χρήση του ειδικού συμβόλου |

– *Παραδείγματα:*

```
bash-3.1$ ls test
```

```
HardLinkToFile1.txt
```

```
SymbLinkToFile2.txt
```

```
test1
```

```
test2
```

```
bash-3.1$ ls test | wc
```

```
4          4          52
```

Εκθέτει τον αριθμό των γραμμών, λέξεων και χαρακτήρων των περιεχομένων του καταλόγου *test*

20-40



Διοχέτευση/Σωλήνωση

```
bash-3.1$ cat test-uniq.test
```

Dan

Dan

**Παράδειγμα ταξινόμησης αρχείου και μετά
αφαίρεσης των διπλότυπων**

Dan

Athena

Ryu

Ryu

Athena

Nick

```
bash-3.1$ sort test-uniq.test | uniq
```

Athena

Dan

Nick

Ryu

```
bash-3.1$ sort test-uniq.test | uniq -c
```

```
  2 Athena
```

```
  3 Dan
```

```
  1 Nick
```

```
  2 Ryu
```



Διοχέτευση/Σωλήνωση

- Μπορούμε να έχουμε περισσότερες από μια διοχέτευση σε κάθε εντολή.
- Πάντοτε διαβάζουμε την εντολή από αριστερά στα δεξιά → η έξοδος κυλά από αριστερά στα δεξιά

```
bash-3.1$ sort test-uniq.test | uniq | wc  
4 4 20
```

Ταξινομεί το αρχείο, μετά αφαιρεί τα διπλά αντίγραφα και στο τέλος εκθέτει τον αριθμό των γραμμών, λέξεων και χαρακτήρων

Ανακατεύθυνση και Διοχέτευση



- Μπορούμε να έχουμε ταυτόχρονα ανακατεύθυνση αρχείων και διοχέτευση.

```
bash-3.1$ sort < test-uniq.test 2> ErrorOutput | uniq 1> Output
```

```
bash-3.1$ cat ErrorOutput
```

```
bash-3.1$ cat Output
```

Athena

Dan

Nick

Ryu