

Υπολογιστική Χημεία Περιβάλλοντος

Βασικές έννοιες και εντολές

Contents

Linux- vi- ssh - ftp.....	2
Χρήση ηλεκτρονικών υπολογιστών σε απόσταση	2
Πρώτη επαφή με την Ανδρομέδα	8
Το λειτουργικό σύστημα Linux.....	12
Τα βασικά χαρακτηριστικά του Linux.....	12
Δομή του συστήματος αρχείων του Linux	12
Βασικές εντολές Linux	14
Είσοδος σε σύστημα Linux μέσω τερματικού Windows	18
Χρήση του VI.....	22
Αρχεία .bashrc	26
File Transfer Protocol, FTP.....	26
Διαγράμματα ροής και εισαγωγή στη fortran 90	29
Πρόγραμμα – Μεταγλωττιστής.....	29
Πως δημιουργούμε ένα πρόγραμμα ?	29
Ανάπτυξη αλγορίθμου - ΔΗΜΙΟΥΡΓΙΑ ΔΙΑΓΡΑΜΜΑΤΩΝ ΡΟΗΣ	31
Δημιουργία και επαλήθευση κώδικα –Δημιουργία εκτελέσιμου και χρήση του.....	32
Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ FORTRAN.....	33
Δομή προγράμματος	35
Εκφράσεις και εντολές	38
Τελεστές	38
Κύκλος επαναλήψεων DO	39
Επιλογή κατεύθυνσης ροής προγράμματος - ερώτηση IF	40
Υποπρόγραμμα (Υπορουτίνα) - CALL	42
ΣΥΝΑΡΤΗΣΗ.....	42
READ- WRITE – PRINT- FORMAT	44
OPEN – CLOSE.....	45
Άλλες εντολές για κατανόηση και πιθανή χρήση	46

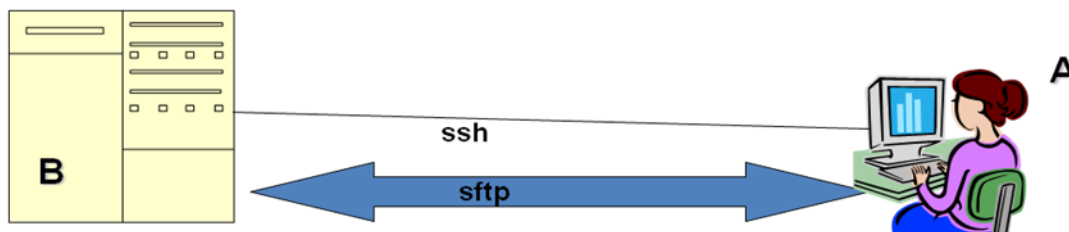
Linux- vi- ssh - ftp

Χρήση ηλεκτρονικών υπολογιστών σε απόσταση

Η σύνδεση σε ένα υπολογιστή εξ αποστάσεως προσφέρει μεταξύ άλλων

- 1- τη δυνατότητα ταυτόχρονης σύνδεσης πολλών χρηστών στο ίδιο μηχάνημα
- 2- τη προσβασιμότητα σε υπολογιστές που βρίσκονται σε μεγάλη απόσταση (άλλη πόλη ή χώρα).

Τρόποι σύνδεσης τους χρήστη από τον υπολογιστή A (τοπικό - client) στο μηχάνημα B (π.χ. υπερυπολογιστής, μονάδα υπολογιστών) που βρίσκεται σε απόσταση (πχ. στη διπλανή αίθουσα, σε άλλη πόλη, χώρα):



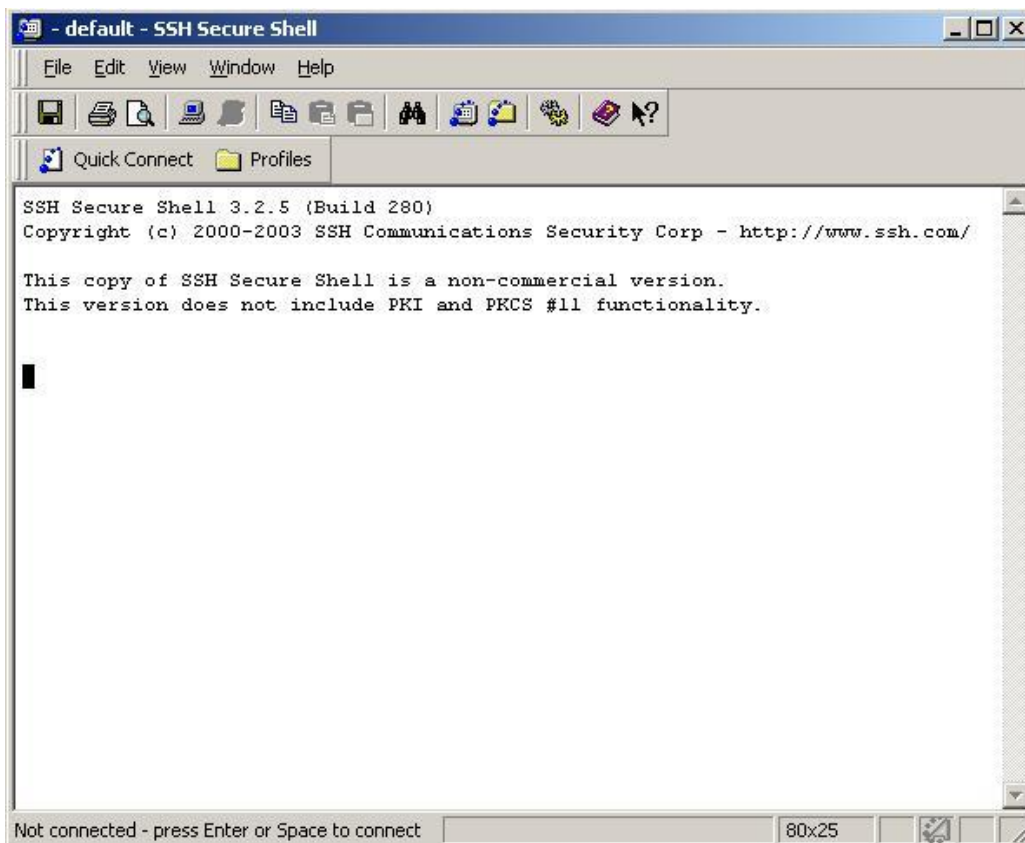
Υπάρχουν 2 τρόποι σύνδεσης που ο καθένας εξυπηρετεί και διαφορετικό σκοπό:

1. Για να **επιμεληθούμε αρχεία** (π.χ. για να γράψουμε ένα πρόγραμμα, να αλλάξουμε κάτι σε ένα πρόγραμμα) ή να κάνουμε υπολογισμούς στον απομακρυσμένο υπολογιστή, τότε χρειάζεται να συνδεθούμε **στον απομακρυσμένο υπολογιστή B** μέσω πρωτοκόλλου σύνδεσης (**telnet, ssh** ή **putty**) που **μετατρέπει το τοπικό μηχάνημα A (client) σε τερματικό** (απλή οθόνη) για τον υπολογιστή B (server).

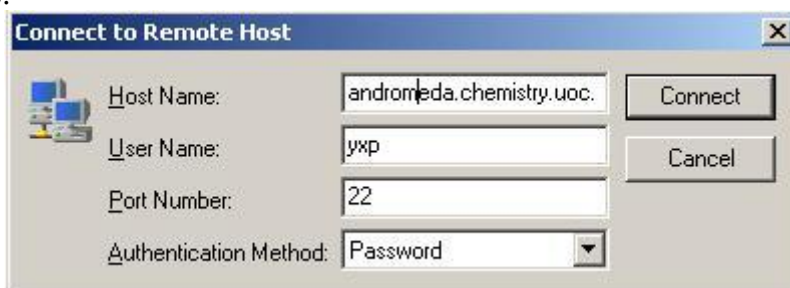
Η οθόνη που ανοίγουμε στη θέση A (client) είναι παράθυρο επικοινωνίας με τον υπολογιστή B (server) και ό,τι εντολή γράφουμε εκτελείται από τον B. Ο υπολογιστής A (client) στη διαδικασία αυτή είναι σαν να μην υπάρχει –λειτουργεί ως οθόνη του B (server).

Παράδειγμα για τη περίπτωση μας: Σύνδεση από ένα από τους ΗΥ της αίθουσας, που χρησιμοποιούν λειτουργικό σύστημα Windows (θέση A- client) με το linux υπολογιστή Andromeda (στη διπλανή αίθουσα- υπολογιστής B - server). Στον server θα δημιουργήσουμε τα υπολογιστικά προγράμματα σε fortran, θα τα μεταφράσουμε σε γλώσσα υπολογιστή (με τον compiler της fortran) δημιουργώντας ένα εκτελέσιμο από τον υπολογιστή αρχείο το οποίο θα του δώσουμε την εντολή και να εκτελέσει. Όλες αυτές οι ενέργειες απαιτούν να χρησιμοποιήσουμε τον client (A) ως τερματικό (οθόνη) για τον server. Αυτό γίνεται π.χ. μέσω του προγράμματος **ssh client**, το οποίο είναι εγκατεστημένο στον client (A), ως ακολούθως

Στην επιφάνεια εργασίας υπάρχει ένα εικονίδιο που λέγεται SSH secure Shell Client. Κάνοντας διπλό κλικ σε αυτό θα ανοίξει το παρακάτω παράθυρο.



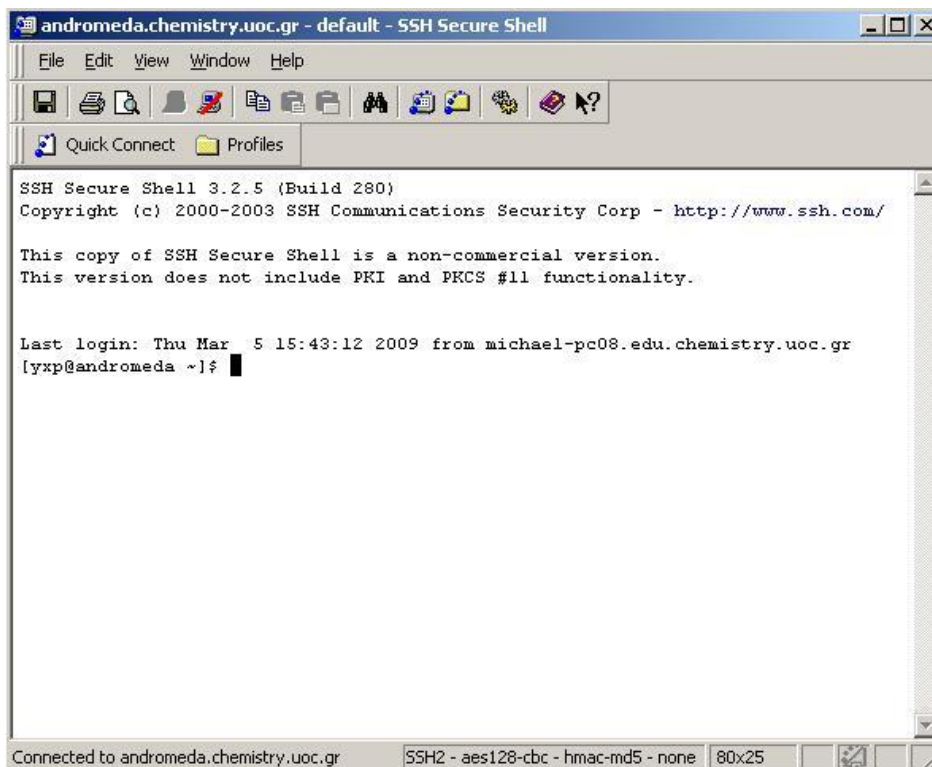
Πατώντας enter εμφανίζει ένα παράθυρο που ζητάει host (ο server πάνω στον οποίο θέλουμε να συνδεθούμε, στην περίπτωσή μας andromeda.chemistry.uoc.gr), το username (στην περίπτωσή μας yxp#), τον αριθμό της πόρτας μέσω της οποίας θα επικοινωνήσει (στην περίπτωση μας 22) και την μέθοδο αναγνώρισης (στην περίπτωσή μας password) όπως φαίνεται παρακάτω.



Συμπληρώνουμε τα στοιχεία και πατάμε Connect. Θα εμφανίσει ένα παράθυρο στο οποίο ζητάει τον κωδικό πρόσβασης.

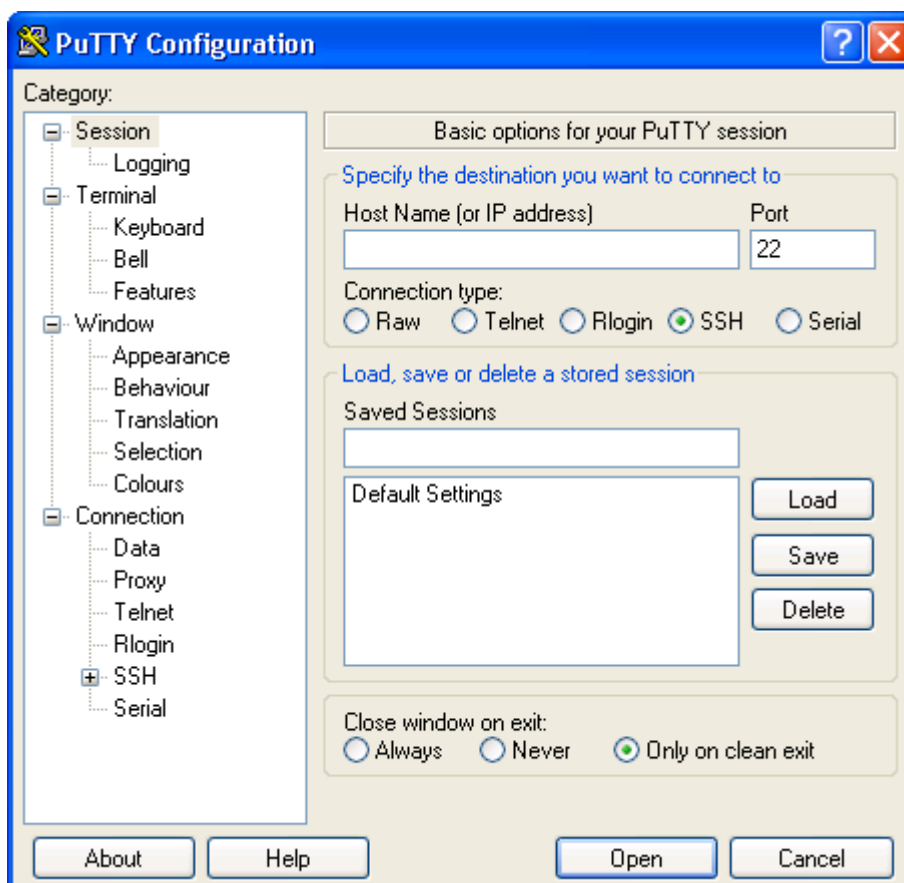


Συμπληρώνουμε τον κωδικό που μας έχει δοθεί και πατάμε OK ή enter. Τώρα έχουμε συνδεθεί στον server andromeda και το παράθυρο μοιάζει κάπως έτσι:



Σε αυτή τη φάση βρίσκεστε στον server (andromeda) σαν να καθόσασταν στο διπλά στον υπολογιστή, που όπως είπαμε μπορεί να είναι και σε άλλη χώρα.

Παρόμοια λειτουργεί και το **putty .exe** Ανοίγοντας το putty.exe θα εμφανιστεί το ακόλουθο παράθυρο.



Στο *Host Name (or IP address)* συμπληρώνουμε την διεύθυνση του υπολογιστή πάνω στον οποίο θέλουμε να συνδεθούμε. Στην περίπτωσή μας

andromeda.chemistry.uoc.gr

και πατάμε Enter. Θα εμφανιστεί το παράθυρο για το login του χρήστη.



Στο login as: δίνουμε το όνομα χρήστη και πατάμε Enter. Για το μάθημα χρησιμοποιούμε τον χρήστη yxp# (ο χρήστης που θα χρησιμοποιήσετε θα σας δοθεί την ώρα του μαθήματος)



Στη συνέχεια μας ρωτάει για τον κωδικό πρόσβασης του χρήστη στο σύστημα.

Προσοχή!!! Κατά την διάρκεια της πληκτρολόγησης του κωδικού πρόσβασης δεν εμφανίζονται χαρακτήρες στην οθόνη.

Δίνουμε τον κωδικό πρόσβασης και πατάμε Enter.

Να τονιστεί ξανά ότι έχει σημασία αν γράφουμε με μικρούς ή με κεφαλαίους χαρακτήρες.

Τώρα έχουμε ανοίξει ένα τερματικό σε σύστημα Linux. Αν δώσουμε την εντολή `pwd` θα απαντήσει:

```
/home/yxp#
```

πράγμα το οποίο σημαίνει ότι βρισκόμαστε στον υποφάκελο `yxp#` του φακέλου `home`. Προτείνεται για κάθε άσκηση να φτιάξετε ένα φάκελο μέσα στον οποίο θα δουλεύετε. Οπότε του δίνουμε την εντολή

```
mkdir foldername [Enter]
```

όπου `foldername` είναι το όνομα που θέλει ο κάθε ένας να δώσει στον φάκελο του. Στην συνέχεια με

```
cd foldername [Enter]
```

μπαίνουμε στον φάκελο μας.

Αν θέλουμε να αποσυνδεθούμε από το σύστημα γράφουμε `logout` ή πατάμε `Ctrl-d`.

2. Για να **μεταφέρουμε αρχεία από το ένα μηχάνημα στο άλλο** χωρίς να τα αλλάξουμε (π.χ. κατεβάζουμε από το διαδίκτυο μία εικόνα, ένα τραγούδι, μια βάση δεδομένων, κάποιες δορυφορικές πρατηρήσεις του όζοντος ή μεταφέρουμε στον υπολογιστή Α (π.χ. για να το τυπώσουμε) ένα πρόγραμμα που γράψαμε στον Β ή και ανάποδα. Κατά τη διαδικασία αυτή οι δύο υπολογιστές δρουν ανεξάρτητα και απλά ανοίγει μια διαύλος (**sftp** ή **ftp**) ανάμεσά τους.

Υπάρχουν διάφορα προγράμματα που ανοίγουν τέτοια δίαυλο 'File Transfer Protocol (FTP)' περισσότερο ή λιγότερο απλά στη χρήση. Μας ενδιαφέρει να ξέρουμε να χρησιμοποιούμε ένα πρόγραμμα που να υποστηρίζεται από όλους υπολογιστές, όπως είναι το `ftp` ή στη μορφή ασφαλείας `sftp` (secure ftp).

File Transfer Protocol, FTP

Το FTP είναι ένα πολύ χρήσιμο εργαλείο για την μεταφορά αρχείων μεταξύ υπολογιστών.

Η χρήση του μπορεί να γίνει είτε μέσω ειδικών προγραμμάτων, είτε μέσω του `command prompt` των Windows.

Command Prompt

Από το **Start** των Windows επιλέγουμε **Run** και στο παράθυρο που εμφανίζεται γράφουμε **cmd** (command prompt). Αυτή εντολή ανοίγει ένα **παράθυρο στον Α** που μοιάζει με την κονσόλα του Linux. Σε αυτή την οθόνη μπορούμε να δώσουμε στον Α εντολές παρόμοιες με αυτές της Linux όπως **sftp** ή **ftp**

Επίσης από αυτό το παράθυρο μπορούμε να περιηγηθούμε στα αρχεία μας με παρόμοιο τρόπο με αυτόν του Linux. Δηλαδή ισχύουν εντολές όπως το **cd**, το **mkdir** και άλλες. Η εντολή `ls` γίνεται `dir` στο `command prompt`.

Αν του δώσουμε την εντολή **ftp [host_name]** τότε προσπαθεί να συνδεθεί στον `host`. Στην περίπτωση μας ο `host` (B) είναι ο `andromeda.chemistry.uoc.gr` Μόλις εντοπισει τον `server` για συνδεθεί σε αυτόν ζητάει :

όνομα χρήστη (user):

κωδικό πρόσβασης (password):

Αν δώσουμε σωστά τα στοιχεία αυτά, τότε η κονσόλα γίνεται κονσόλα `ftp` και είναι κάπως έτσι:

ftp>

Έτσι έχουμε ανοίξει ένα δίαυλο επικοινωνίας μεταξύ των 2 μηχανημάτων για μεταφορά και διαχείριση αρχείων αλλά δεν μπορούμε να τα επεξεργαστούμε (π.χ. με VI). Ισχύουν οι εντολές περιήγησης του Linux (**cd**, **cd ..**, **pwd** κλπ) για να μετακινούμαστε μέσα στο σύστημα (B). Με την εντολή **dir** μπορούμε να δούμε τα περιεχόμενα του φακέλου που βρισκόμαστε.

Αν θέλουμε **να δούμε ή να περιηγηθούμε στους φακέλους του υπολογιστή στον οποίο δουλεύουμε** (τερματικό) χρησιμοποιούμε την εντολή **lcd** ή **lcd [folderpath]** αντίστοιχα όπου *folderpath* η πλήρης διαδρομή του φακέλου στον οποίο θέλουμε να πάμε, μέσα σε εισαγωγικά (π.χ. **lcd "C:\Documents and Settings\user\Desktop"**). Ο φάκελος στον οποίο βρισκόμαστε τοπικά είναι και ο φάκελος που είναι άμεσα συνδεδεμένος με τον φάκελο που βρισκόμαστε στο απομακρυσμένο μηχάνημα Linux.

Όταν βρούμε το αρχείο που θέλουμε να μεταφέρουμε από τον server (B) στον υπολογιστή μας (A), γράφουμε

get filename [Enter]

οπότε το αρχείο μεταφέρεται στον υπολογιστή μας στον φάκελο που έχουμε επιλέξει με το **lcd [folderpath]**

Για να μεταφέρουμε ένα αρχείο από τον υπολογιστή μας (A) στον server (B) χρησιμοποιούμε την εντολή **put**

put filename [Enter]

μεταφέρει το αρχείο **filename** από τον υπολογιστή μας στον server στον φάκελο που έχουμε επιλέξει με το **cd [folderpath]**.

Υπάρχουν και οι εντολές **mget** και **mput** που η διαφορά τους από τα απλά **get** και **put** είναι ότι ζητάει για επιβεβαίωση πριν τη μεταφορά.

mget * για να πάρουμε όλα τα αρχεία του φακέλου.

mput * για να βάλουμε όλα τα αρχεία του τοπικού φακέλου στον απομακρυσμένο.

και στις 2 παραπάνω περιπτώσεις μας ρωτάει ανά αρχείο αν θέλουμε να το μεταφέρουμε. Απαντάμε **y** (yes) ή **n** (no)

Επίσης μπορούμε να επιλέξουμε μερικά αρχεία π.χ. όλα όσα έχουν κατάληξη **.f90**
mget *.f90

Άρα παραπάνω το ***** χρησιμοποιείται για να δηλώσουμε οποιουδήποτε συνεχόμενους (χωρίς κενό ενδιάμεσα) χαρακτήρες του ονόματος ενός αρχείου.

Για την μεταφορά αρχείων με συγκεκριμένα format (word, excel κλπ), πριν την μεταφορά δίνουμε την εντολή **bin** (binary transfer).

Για να κλείσουμε την σύνδεση με τον server γράφουμε

bye ή **logout**

File Zilla

Στους υπολογιστές του εργαστηρίου είναι εγκατεστημένο το πρόγραμμα **filezilla** που χρησιμοποιείται για FTP και παρέχει γραφικό περιβάλλον.

Όταν ανοίξουμε το πρόγραμμα, στην πάνω μεριά της οθόνης μπορούμε να γράψουμε τον **host**, το όνομα χρήστη και τον κωδικό και να συνδεθούμε στον server.

Στην αριστερή μεριά της οθόνης βλέπουμε τους δίσκους και τα αρχεία που υπάρχουν στον υπολογιστή μας και στην δεξιά μεριά βλέπουμε τα αρχεία του server. Μπορούμε να περιηγηθούμε στους φακέλους ακριβώς όπως περιηγούμαστε σε φακέλους των Windows. Όταν βρούμε το αρχείο που θέλουμε να μεταφέρουμε, πατάμε πάνω του με το ποντίκι και το τραβάμε στον φάκελο που θέλουμε να το μεταφέρουμε.

Πρώτη επαφή με την Ανδρομέδα ...

Όταν καθίσετε μπροστά από τον υπολογιστή θα δείτε μια οθόνη που θα λέει ότι πρέπει να πατήσετε ctrl-alt-del για να μπορέσετε να συνδεθείτε.

Πατώντας το θα ζητήσει username και password. Αφού του δώσετε τα σωστά στοιχεία, **την πρώτη φορά που θα συνδεθείτε θα σας ζητηθεί να αλλάξετε τον κωδικό πρόσβασης σας.**

1. Σύνδεση με **ssh** στην `andromeda.chemistry.uoc.gr`
2. Αλλαγή κωδικού πρόσβασης στην `andromeda` (δείτε εντολές Linux `passwd`)
3. Αντιγραφή του αρχείου `testfile.f90` από τον χρήστη `yxp` `/home/yxp/testfile.f90`
4. Επεξεργασία του αρχείου ώστε να γράφει το κείμενο της παρακάτω σελίδας.
5. Μεταφορά του αρχείου με `ftp` στο τοπικό μηχάνημα Windows για εκτύπωση
6. Προσθήκη δυνατότητας πρόσβασης στον εκτυπωτή (σε περίπτωση που δεν έχουμε ήδη δυνατότητα): από το `start` → `printers and faxes`
Ανοίγει ένα παράθυρο που στην αριστερή πλευρά του έχει την επιλογή **Add a printer**. Πατώντας εκεί ανοίγει το παρακάτω παράθυρο διαλόγου

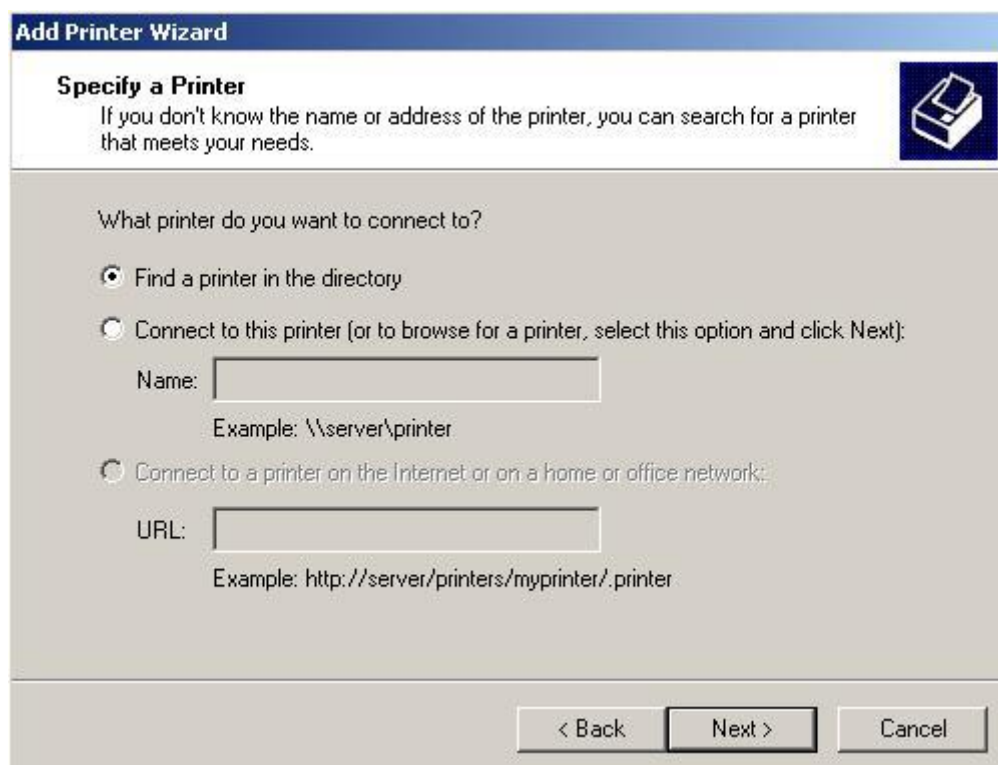


Στο οποίο πατάμε Next. Εμφανίζεται το επόμενο παράθυρο

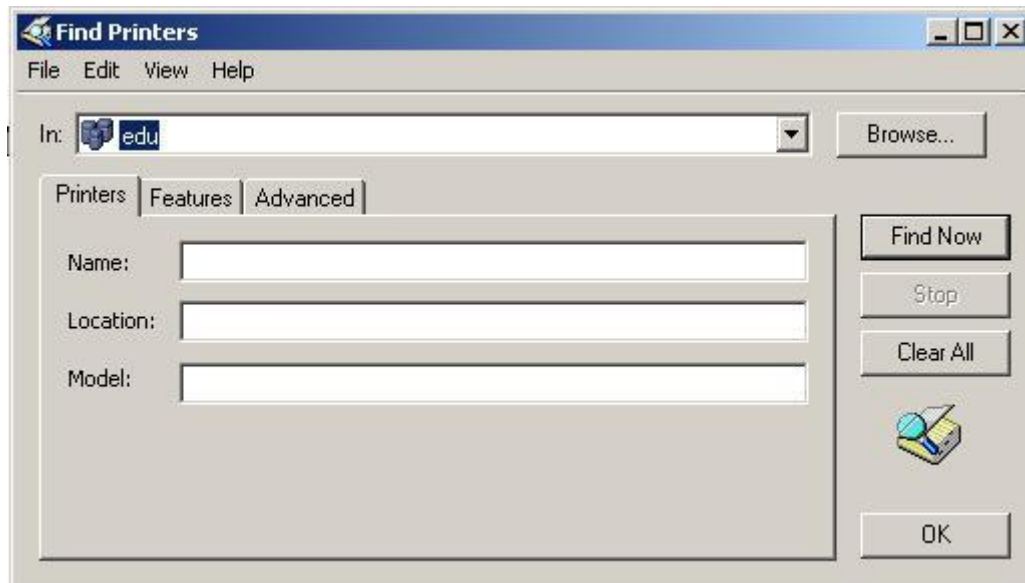


Επιλέγουμε **A network printer, or a printer attached to another computer** και πατάμε Next.

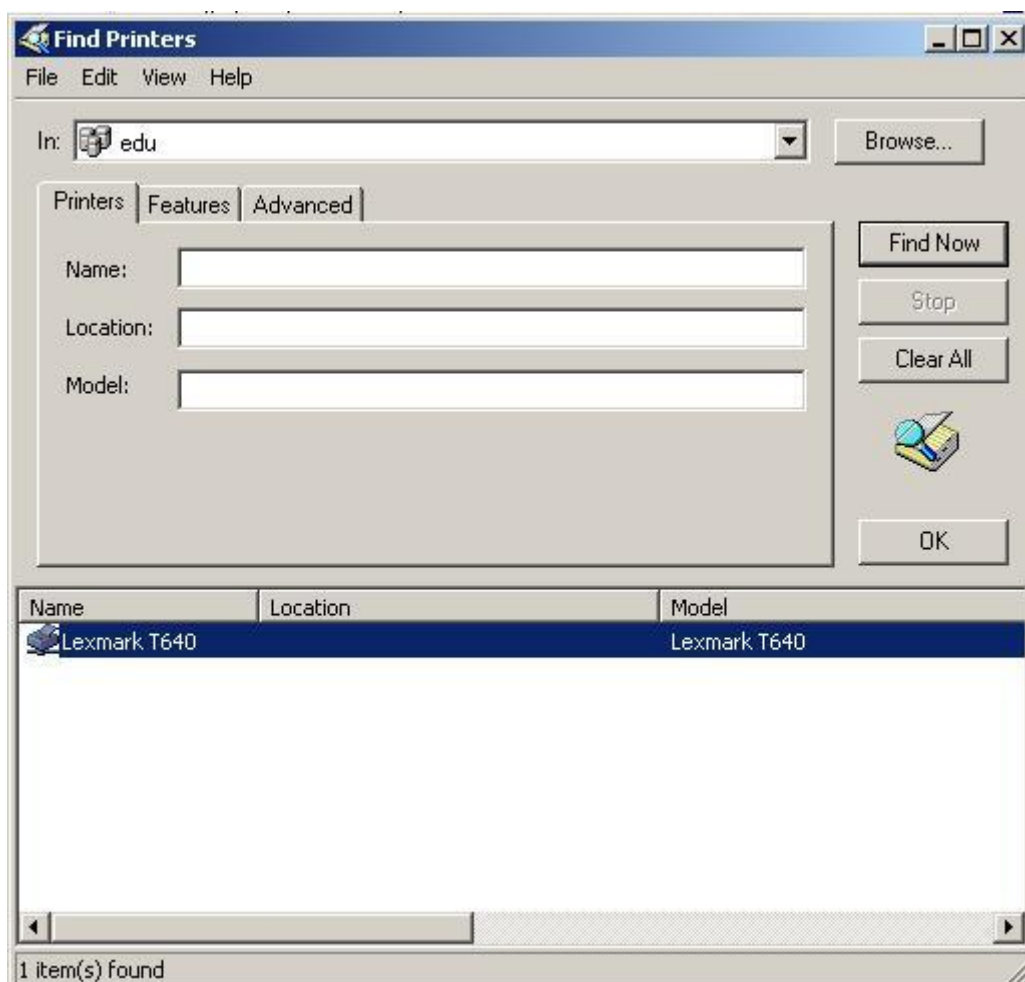
Στο επόμενο παράθυρο διαλόγου, που μοιάζει με το παρακάτω:



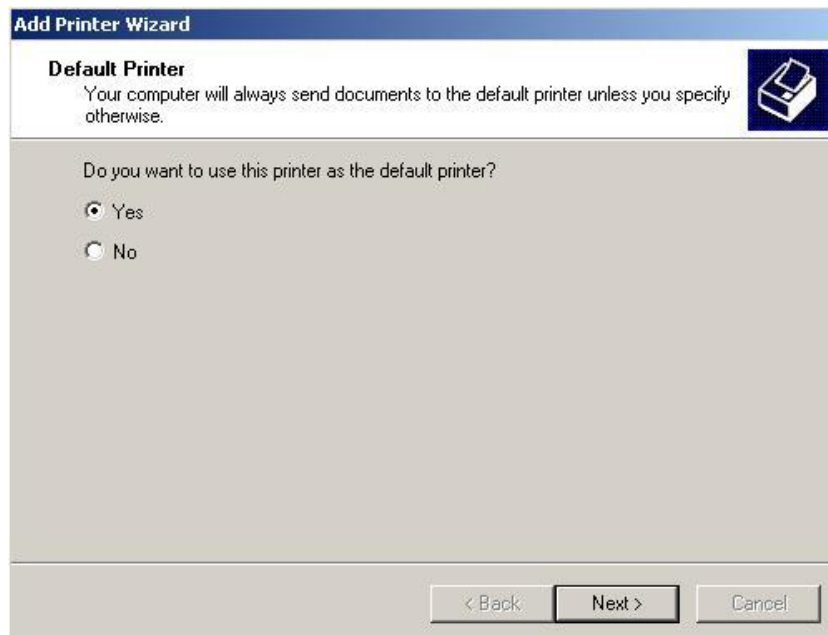
Επιλέγουμε **Find a printer in the directory** και πατάμε Next. Εμφανίζεται το παρακάτω παράθυρο.



Στο **In** επιλέγουμε το directory **edu** και πατάμε **Find Now**. Θα ξεκινήσει την αναζήτηση για εκτυπωτές και θα εμφανίσει τα αποτελέσματα κάπως έτσι:



Επιλέγουμε τον εκτυπωτή **Lexmark T640** όπως φαίνεται και στην παραπάνω εικόνα και πατάμε **OK**. Εμφανίζεται ένα παράθυρο σαν το παρακάτω:



Που μας ρωτάει αν θέλουμε να είναι αυτός ο προεπιλεγμένος εκτυπωτής. Επιλέγουμε **Yes** και πατάμε Next όπου και εμφανίζεται το τελικό παράθυρο διαλόγου εγκατάστασης του εκτυπωτή:



Επιλέγουμε **Finish** και τώρα μπορούμε να τυπώσουμε στον εκτυπωτή της αίθουσας.

ΠΡΟΣΟΧΗ!!! Αυτή η διαδικασία χρειάζεται να γίνει μόνο μια φορά. Από εκεί και πέρα ο εκτυπωτής είναι εγκατεστημένος στον λογαριασμό σας στο υπολογιστικό κέντρο και μπορείτε να τον χρησιμοποιείτε.

7. Εκτύπωση του αρχείου

Το λειτουργικό σύστημα Linux

<http://www.linux.org/forums/beginner-tutorials.53/>

<http://www.yolinux.com/TUTORIALS/LinuxTutorialHardware.html>

Το Linux είναι ένα ελεύθερο (Open Source) λειτουργικό σύστημα- κλώνος του UNIX για προσωπικούς υπολογιστές (Personal Computer, PC). Αρχικά αναπτύχθηκε από τον Linus Torvalds ο οποίος ξεκίνησε να εργάζεται στο Linux το 1991, όταν ήταν φοιτητής στο Πανεπιστήμιο του Ελσίνκι στη Φινλανδία. Ο Linus κυκλοφόρησε την αρχική έκδοση του Linux στο Internet ως ελεύθερο λογισμικό, ξεκινώντας με αυτή του την κίνηση άθελά του ένα από τα μεγαλύτερα φαινόμενα όλων των εποχών στην ανάπτυξη λογισμικού. Σήμερα το Linux υποστηρίζεται από χιλιάδες προγραμματιστές σε όλο τον κόσμο οι οποίοι συνεργάζονται μέσω του Internet. Επίσης υπάρχουν πολλές εταιρίες παροχής υπηρεσιών υποστήριξης του Linux. Για τους σκοπούς του μαθήματος θα χρησιμοποιηθεί η διανομή Red Hat Linux Enterprise Edition.

Τα βασικά χαρακτηριστικά του Linux.

Το Linux παρουσιάζει πολλά χαρακτηριστικά, τα σημαντικότερα από τα οποία συνοψίζονται παρακάτω:

- Ταυτόχρονη εκτέλεση πολλών διεργασιών (multitasking).
- Σύστημα πολλών χρηστών. Επιτρέπει την (ταυτόχρονη) χρήση του ίδιου υπολογιστή από πολλούς χρήστες.
- Δυνατότητα χρήσης κελύφους εντολών ή γραφικού περιβάλλοντος (στο μάθημα θα χρησιμοποιηθεί το κέλυφος εντολών)

Το Linux υποστηρίζει την ταυτόχρονη σύνδεση πολλών χρηστών στο σύστημα, ή πολλές ταυτόχρονες συνδέσεις του ίδιου χρήστη. Για την σύνδεση απαιτείται το *όνομα χρήστη (username ή login)* και ο *κωδικός πρόσβασης (password)*.

Δομή του συστήματος αρχείων του Linux

Στα Linux συστήματα τα πάντα θεωρούνται αρχεία. Όλο το υλικό (hardware) και το λογισμικό (software) υπάρχει στην μορφή αρχείων μέσα στον σκληρό δίσκο.

Χρησιμοποιείται δόμηση δένδρου (tree) με διαχωριστικό διακριτικό την κάθετο (slash, /).

Η δομή των αρχείων αυτών στον δίσκο έχει ως εξής:

/	είναι ο αρχικός (βασικός) φάκελος που περιέχει το λειτουργικό σύστημα και όλα όσα χρειάζεται ο υπολογιστής για να ξεκινήσει.
/usr	φάκελος ο οποίος περιέχει όλες τις εντολές, τις βιβλιοθήκες (libraries) και γενικά όλα όσα δεν αλλάζουν κατά την κανονική χρήση και λειτουργία του συστήματος.
/var	φάκελος που περιέχει αρχεία που αλλάζουν κατά την κανονική λειτουργία του συστήματος, όπως για παράδειγμα βάσεις δεδομένων κ.α.
/home	φάκελος ο οποίος περιέχει τα αρχεία των χρηστών του συστήματος, όπως για παράδειγμα ρυθμίσεις, δεδομένα που αποθηκεύουν οι χρήστες, e-mail κ.α.
/proc	φάκελος που περιέχει εικονικά αρχεία. Ουσιαστικά ένα αρχείο που βρίσκεται σε αυτό το φάκελο δεν είναι αποθηκευμένο στο δίσκο ούτε καταλαμβάνει χώρο σε αυτόν. Όταν βλέπουμε ένα αρχείο από αυτό το φάκελο στην πραγματικότητα βλέπουμε πληροφορίες που βρίσκονται στην μνήμη RAM του υπολογιστή.
/bin	εκτελέσιμα αρχεία που χρειάζονται κατά την εκκίνηση και ίσως χρησιμοποιηθούν από τους χρήστες του συστήματος. Όλες οι βασικές εντολές Linux που περιγράφονται στο επόμενο μέρος χρειάζονται το αντίστοιχο εκτελέσιμο
/sbin	εκτελέσιμα αρχεία που δεν προορίζονται για χρήση από τους χρήστες του συστήματος.
/etc	αρχεία που ρυθμίζουν το λειτουργικό σύστημα
/root	το home directory (αρχικός φάκελος) του διαχειριστή του συστήματος (ο διαχειριστής ονομάζεται super user, su ή root)
/dev	αρχεία συσκευών. Στα Linux συστήματα, όπως είπαμε και παραπάνω, ακόμα και το υλικό εμφανίζεται ως αρχεία, τα οποία τοποθετούνται μέσα σε αυτό το φάκελο.
/mnt	το σημείο στο οποίο «συνδέονται» αφαιρούμενοι δίσκοι (cdrom, DVD-ROM, floppy, δικτυακοί δίσκοι κ.α.). Συνήθως περιέχει ένα φάκελο για κάθε συσκευή που είναι συνδεδεμένη στο σύστημα.
/lib	βιβλιοθήκες για την λειτουργία προγραμμάτων.
/boot	αρχεία που χρησιμοποιούνται για την εκκίνηση του συστήματος. Συνήθως περιέχει και τον πυρήνα (kernel) του συστήματος.

/opt	στον φάκελο αυτό αποθηκεύονται οι διάφορες εφαρμογές που εγκαθίστανται στο σύστημα.
/tmp	φάκελος για την αποθήκευση προσωρινών αρχείων που χρειάζεται το σύστημα για την λειτουργία του. Καθαρίζεται αυτόματα από το σύστημα όταν αυτό χρειάζεται
/lost+found	αρχεία που ανακτήθηκαν κατά την επιδιόρθωση του συστήματος (σε περίπτωση βλάβης)

Οι φάκελοι στο Linux εμφανίζονται με χρώμα μπλέ και τα εκτελέσιμα αρχεία με ανοικτό πράσινο, προγράμματα με κόκκινο ή πράσινο.

<http://www.cyberciti.biz/tips/where-is-color-of-ls-command-defined.html>

Βασικές εντολές Linux

Στα Linux συστήματα, βασική εφαρμογή είναι το shell (κέλυφος). Το πιο διαδεδομένο κέλυφος είναι το bash (Bourne Again Shell) που είναι μια δωρεάν έκδοση του κελύφους Bourne.

Μέσω του κελύφους γίνεται η «επικοινωνία» του χρήστη με τον υπολογιστή. Οι εντολές δίνονται υπό την μορφή:

Command -flag argument

όπου command είναι η εντολή, argument δείχνει που θα εφαρμοστεί η εντολή και flag είναι κάποιες περαιτέρω ιδιότητες που μπορεί ο χρήστης να δώσει στην κάθε εντολή.

Προσοχή!!! Τα πάντα στο Linux είναι case sensitive, δηλαδή έχει σημασία αν κάτι είναι γραμμένο με μικρούς ή με κεφαλαίους χαρακτήρες.

Οι βασικές εντολές που χρειάζονται για την επικοινωνία αυτή είναι:

cd [change **d**irectory] Χρησιμοποιείται για την περιήγηση ανάμεσα στους φακέλους στο σύστημα.

Γενικά για την περιήγηση στους φακέλους μας υπάρχουν 2 ορισμοί πολύ χρήσιμοι.

- είναι ο φάκελος στον οποίο βρισκόμαστε
- είναι ο αμέσως προηγούμενος φάκελος.

Οπότε αν θέλουμε να πάμε ένα φάκελο πίσω γράφουμε

cd .. [Enter]

chfn change finger information

finger [username]

w

who

finger

cal [**calendar**] εμφανίζει το ημερολόγιο του τρέχοντος μήνα.

cal year εμφανίζει το ημερολόγιο του χρόνου που ζητήθηκε.

cat [**concatenate**] εμφανίζει στην οθόνη τα περιεχόμενα κάποιου αρχείου. Παράδειγμα το `cat filename` θα εμφανίσει στην οθόνη το αρχείο `filename`.

cat -v εμφανίζει και τους μη εκτυπώσιμους χαρακτήρες του αρχείου που ζητείται να εμφανιστεί.

chgrp [**change group**] αλλάζει την ομάδα ιδιοκτησίας του αρχείου.

chown [**change owner**] αλλάζει τον ιδιοκτήτη του αρχείου.

chmod [**change mode**] αλλάζει την άδεια χρήσης του αρχείου.

cpm [**compare**] συγκρίνει 2 αρχεία και εμφανίζει την πρώτη διαφορά που βρίσκει.

diff a b>c συγκρίνει το αρχείο `a` με το αρχείο `b` και καταγράφει τις διαφορές τους στο αρχείο `c`.

cp [**copy**] χρησιμοποιείται για την αντιγραφή ενός αρχείου.

cp -r χρησιμοποιείται για την αντιγραφή φακέλου.

mv [**move**] χρησιμοποιείται για την μετακίνηση ενός αρχείου ή ενός φακέλου καθώς και για την μετονομασία.

find ψάχνει στον δίσκο για αρχεία που να πληρούν τα κριτήρια που του δίνουμε.

file [filename] μας δείχνει τι τύπο αρχείου είναι το `[filename]`

grep	[g lobal r egular e xpression and p rint] ψάχνει μέσα στα αρχεία για τη ζητούμενη έκφραση.
logname	εμφανίζει το username του χρήστη.
ls	[l ist] εμφανίζει τα περιεχόμενα του τρέχοντα φακέλου.
ls -a	[l ist -a ll] εμφανίζει όλα τα περιεχόμενα του φακέλου, ακόμα και τα κρυφά αρχεία.
ls -l	[l ist -l ong list] εμφανίζει τα περιεχόμενα του φακέλου με πληροφορίες δίπλα από το κάθε ένα.
ls -la	συνδυασμός των προηγούμενων 2.
ls -ltr , ls -lSr, ls -lX, ls -lXr	
ls -- help	
ln	δημιουργεί ένα σύνδεσμο symlink με ένα αρχείο test η εντολή είναι ln -s test symlink
more [filename]	δειχνει το αρχείο [filename] για να σταματήσει πατάμε το q
less [filename]	δειχνει το αρχείο [filename] και μπορούμε να μετακινηθούμε πάνω κάτω.
passwd	αλλάζει τον κωδικό πρόσβασης του χρήστη.
ps	[p rocess] δείχνει στην οθόνη τις διεργασίες που τρέχουν εκείνη τη στιγμή.
pwd	[p rint w orking d irectory] εμφανίζει στην οθόνη την διαδρομή για το φάκελο που βρίσκεται ο χρήστης.
rm	[r emove] χρησιμοποιείται για διαγραφή αρχείου.
rm -r	χρησιμοποιείται για διαγραφή φακέλου.
rm -i	χρησιμοποιείται για την διαγραφή αρχείου ή αρχείων. Διαφέρει στο rm στο ότι ζητάει επιβεβαίωση για την διαγραφή.
tail	εμφανίζει στην οθόνη τις τελευταίες γραμμές ενός αρχείου.
tail -f	το ίδιο με πριν, μόνο που σε περίπτωση που το αρχείο αλλάζει, εμφανίζει και τις αλλαγές.
logout	αποσυνδέει το χρήστη από το σύστημα.
clear	καθαρίζει την οθόνη από προηγούμενες εντολές και μηνύματα.

man	[man ual] προσφέρει βοήθεια για την χρήση εντολών.
Man [command]	για τη συγκεκριμένη εντολή
mkdir	[make directory] δημιουργεί υποφάκελους.
rmdir [dirname]/	σβήνει τον φάκελο αρκεί νάνε άδειος
./[filename]	εκτέλεση του εκτελέσιμου αρχείου filename.
who	εμφανίζει στην οθόνη τους χρήστες που είναι συνδεδεμένοι εκείνη τη στιγμή στο σύστημα.
Whereis XX	δείχνει που βρίσκεται το XX στον υπολογιστή (full path)
finger	όπως πριν αλλά με περισσότερες λεπτομέρειες για τον χρήστη.
ps -u [username]	εμφανίζει στην οθόνη όλες τις λειτουργίες που τρέχει ο συγκεκριμένος χρήστης.
kill [process#]	σταματάει («σκοτώνει») την διεργασία με αριθμό #.
top	δείχνει τη λίστα των διεργασιών που τρέχουν στον ΗΥ
touch [a file name]	δημιουργεί το αρχείο αν δεν υπάρχει ή επικαιροποιεί τη χρήση του.
echo	επαναλαμβάνει/γράφει αυτό που έπεται
date	γράφει την ημερομηνία

Πέρα από τις εντολές, υπάρχουν και συντομεύσεις του πληκτρολογίου που μπορεί ο χρήστης να χρησιμοποιεί για διάφορες λειτουργίες. Μερικές από αυτές είναι:

Ctrl – D	τέλος αρχείου ή σύνδεσης.
Ctrl – C	σταματά την εντολή που τρέχει εκείνη τη στιγμή.
Ctrl –]	αντιστοιχεί στο πλήκτρο ESC.
Ctrl – Z	θέτει την εντολή που τρέχει, σε διαθεσιμότητα.
Ctrl – U	σβήνει την γραμμή εντολών.
Ctrl – H	σβήνει τον χαρακτήρα αριστερά του κέρσορα.

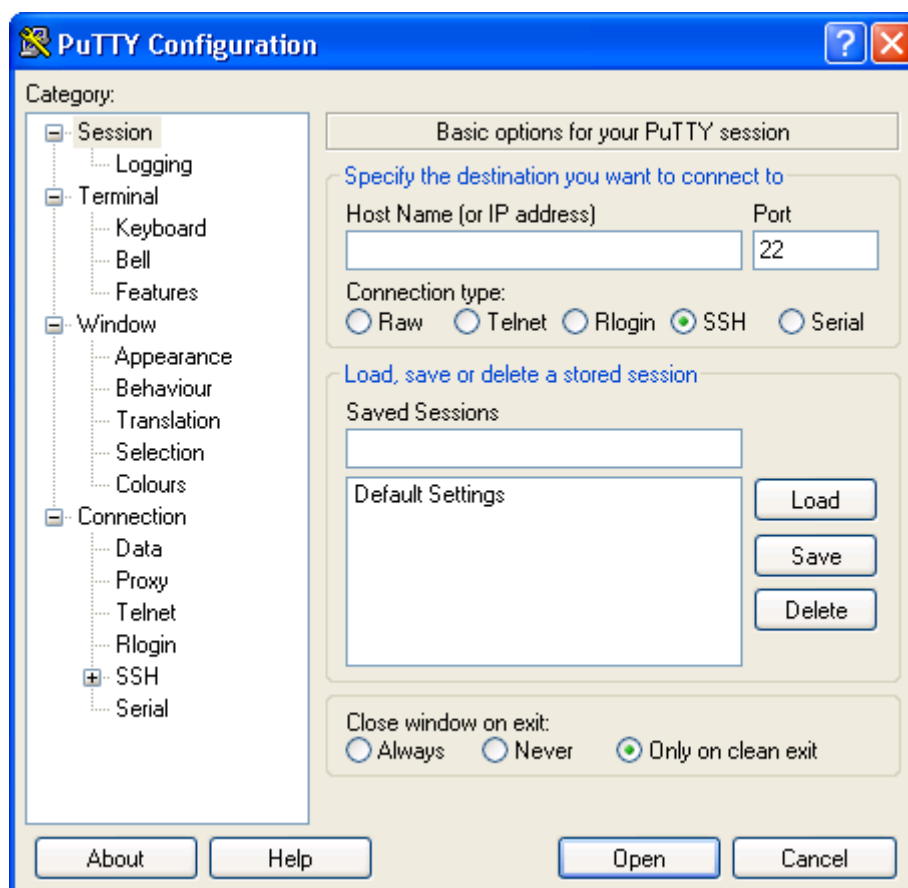
Είσοδος σε σύστημα Linux μέσω τερματικού Windows

Για τις ανάγκες του μαθήματος υπάρχει ένας υπολογιστής με σύστημα Linux πάνω στον οποίο θα συνδέονται όλοι οι χρήστες μέσω τερματικού από υπολογιστές με σύστημα Windows.

Για να γίνει αυτό χρειαζόμαστε κάποιο πρόγραμμα secure shell client όπως για παράδειγμα το putty.exe ή το SSH Secure Shell.

Στο παράδειγμα θα χρησιμοποιηθεί το putty.exe.

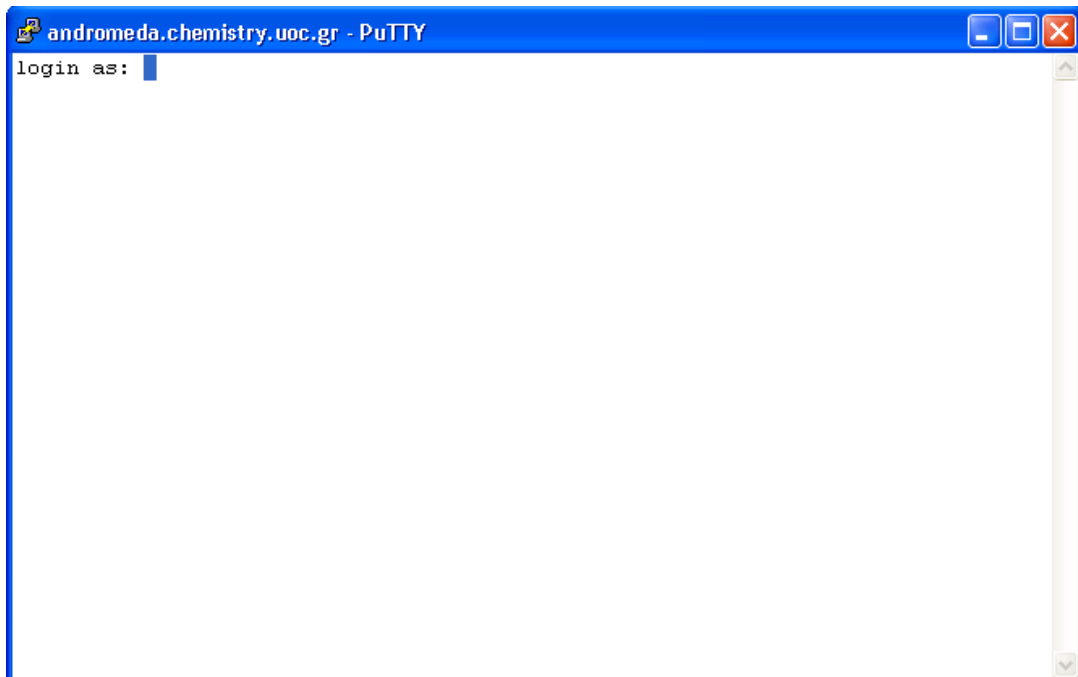
Ανοίγοντας το putty.exe θα εμφανιστεί το ακόλουθο παράθυρο.



Στο *Host Name (or IP address)* συμπληρώνουμε την διεύθυνση του υπολογιστή πάνω στον οποίο θέλουμε να συνδεθούμε. Στην περίπτωσή μας

andromeda.chemistry.uoc.gr

και πατάμε Enter. Θα εμφανιστεί το παράθυρο για το login του χρήστη.



Στο login as: δίνουμε το όνομα χρήστη και πατάμε Enter. Για το μάθημα χρησιμοποιούμε τον χρήστη yxp# (ο χρήστης που θα χρησιμοποιήσετε θα σας δοθεί την ώρα του μαθήματος)



Στη συνέχεια μας ρωτάει για τον κωδικό πρόσβασης του χρήστη στο σύστημα.

Προσοχή!!! Κατά την διάρκεια της πληκτρολόγησης του κωδικού πρόσβασης δεν εμφανίζονται χαρακτήρες στην οθόνη.

Δίνουμε τον κωδικό πρόσβασης και πατάμε Enter.

Να τονιστεί ξανά ότι έχει σημασία αν γράφουμε με μικρούς ή με κεφαλαίους χαρακτήρες.

Τώρα έχουμε ανοίξει ένα τερματικό σε σύστημα Linux. Αν δώσουμε την εντολή `pwd` θα απαντήσει:

```
/home/yxp#
```

πράγμα το οποίο σημαίνει ότι βρισκόμαστε στον υποφάκελο `yxp#` του φακέλου `home`. Προτείνεται για κάθε άσκηση να φτιάξετε ένα φάκελο μέσα στον οποίο θα δουλεύετε. Οπότε του δίνουμε την εντολή

```
mkdir foldername [Enter]
```

όπου `foldername` είναι το όνομα που θέλει ο κάθε ένας να δώσει στον φάκελο του. Στην συνέχεια με

```
cd foldername [Enter]
```

μπαίνουμε στον φάκελο μας.

Αν θέλουμε να αποσυνδεθούμε από το σύστημα γράφουμε `logout` ή πατάμε `Ctrl-d`.

Αν θέλουμε να σβήσουμε τον υπολογιστή (μόνο ο `administrator`) δίνουμε την εντολή

```
shutdown -h now
```

```
shutdown -h 13:05
```

```
shutdown -h +5
```

(+5 minutes)

οπότε το μηχάνημα στέλνει το μήνυμα

Linux is going for system halt NOW

Και μετά κλείνει όλα τα προγράμματα που τρέχει, όταν είναι έτοιμο για να το σβήσουμε δίνει το μήνυμα

System halted

Οπότε και μπορούμε να το αποσυνδέσουμε από το ηλεκτρικό.

Ο επεξεργαστής κειμένου VI

Ο επεξεργαστής κειμένου VI ξεκίνησε να αναπτύσσεται γύρω στο 1976 από τον Bill Joy, ο οποίος τότε ήταν μεταπτυχιακός φοιτητής στο Πανεπιστήμιο της Καλιφόρνια, στο Μπέρκλεϊ. Αργότερα, ο Joy βοήθησε στην ίδρυση της Sun Microsystems στην οποία και έγινε επιστημονικός υπεύθυνος.

Ο VI βασίστηκε στον ήδη υπάρχοντα τότε επεξεργαστή, τον ed ο οποίος επεξεργαζόταν μια γραμμή κάθε φορά. Ο Joy αρχικά έφτιαξε τον ex, ένα επεξεργαστή κειμένου σαν τον ed, με πολλές περισσότερες δυνατότητες. Αργότερα άρχισε να αναπτύσσει τον VI σαν “visual interface” του ex. Ουσιαστικά το VI επιτρέπει να βλέπουμε το κείμενο σε όλη την οθόνη και όχι μια γραμμή κάθε φορά, γεγονός που του δίνει και το όνομα του.

Ο VI παραμένει ακόμα και σήμερα ιδιαίτερα διαδεδομένος, παρά την ανάπτυξη πολλών επεξεργαστών κειμένου με γραφικό περιβάλλον, που είναι πολύ ευκολότεροι στην χρήση για αρχάριους.

Χρήση του VI

Το VI έχει 2 βασικές λειτουργίες, την λειτουργία εντολών (command mode) και την λειτουργία εισαγωγής (insert mode). Όταν βρισκόμαστε σε command mode κάθε πλήκτρο του πληκτρολογίου είναι και μία εντολή. Για να γράψουμε κείμενο πρέπει να μπούμε σε insert mode.

Για να ξεκινήσουμε ένα καινούριο κείμενο με vi πρέπει στην γραμμή εντολών του Linux να γράψουμε

```
vi filename
```

όπου filename είναι το όνομα που θέλουμε να δώσουμε στο αρχείο. Έτσι ανοίγει το vi και δημιουργείται ένα αρχείο με το όνομα filename.

Μόλις ανοίγει το vi είναι σε command mode. Οπότε δεν μπορούμε αμέσως να γράψουμε κείμενο.

Κάποιες από τις εντολές του είναι οι παρακάτω.

Εντολές κίνησης του κέρσορα.

h	κίνηση προς τα αριστερά.
j	κίνηση προς τα κάτω.
k	κίνηση προς τα πάνω
l	κίνηση προς τα δεξιά.
[Enter]	μετάβαση στην αρχή της επόμενης γραμμής.
\$	μετάβαση στην τελευταία στήλη της γραμμής που βρισκόμαστε.
0	μετάβαση στην πρώτη στήλη της γραμμής που βρισκόμαστε.
^	μετάβαση στην πρώτη μη κενή στήλη στην γραμμή που βρισκόμαστε.
w	μετάβαση στην αρχή της επόμενης λέξης ή στο επόμενο σημείο στίξης.
W	μετάβαση μετά το επόμενο κενό.
b	μετάβαση στην αρχή της προηγούμενης λέξης ή σημείο στίξης.
B	μετάβαση στην αρχή της προηγούμενης λέξης –αγνοεί σημεία στίξης.

e	μετάβαση στο τέλος της επόμενης λέξης ή σημείο στίξης.
E	μετάβαση στο τέλος της επόμενης λέξης –αγνοεί σημεία στίξης.
H	μετάβαση στην αρχή της οθόνης.
M	μετάβαση στην μέση της οθόνης.
L	μετάβαση στο τέλος της οθόνης.

Μετακινήσεις οθόνης

G	μετάβαση στην τελευταία γραμμή του αρχείου.
#G	μετάβαση στην γραμμή #.
z+	μετακίνηση της συγκεκριμένης γραμμής στην πάνω μεριά της οθόνης.
z	μετακίνηση της συγκεκριμένης γραμμής στην μέση της οθόνης.
z-	μετακίνηση της συγκεκριμένης γραμμής στην κάτω μεριά της οθόνης.
Ctrl-F	μετακίνηση μία οθόνη κάτω.
Ctrl-B	μετακίνηση μία οθόνη πάνω.
Ctrl-D	μετακίνηση μισή οθόνη κάτω.
Ctrl-U	μετακίνηση μισή οθόνη πάνω.

Εντολές εισαγωγής κειμένου.

Με τις ακόλουθες εντολές μπαίνουμε σε insert mode ώστε να γράψουμε ή να αντικαταστήσουμε κείμενο.

r	αντικατάσταση του χαρακτήρα κάτω από τον κέρσορα με τον επόμενο χαρακτήρα που πληκρολογείται.
R	αντικαθιστά χαρακτήρες μέχρι να πατήσουμε Esc.
i	ξεκινάει την εισαγωγή του κειμένου από τον χαρακτήρα που βρίσκεται ακριβώς κάτω από τον κέρσορα.
I	ξεκινά την εισαγωγή του κειμένου από την αρχή της γραμμής στην οποία βρίσκεται ο κέρσορας.
a	ξεκινά την εισαγωγή κειμένου από τον επόμενο χαρακτήρα από εκεί που βρίσκεται ο κέρσορας.
A	ξεκινά την εισαγωγή του κειμένου από το τέλος της τρέχουσας γραμμής.

- o εισάγει μια καινούρια γραμμή κάτω από την γραμμή που βρίσκεται ο κέρσορας και ξεκινά την εισαγωγή κειμένου από εκεί.
- O εισάγει μια καινούρια γραμμή πάνω από την γραμμή που βρίσκεται ο κέρσορας και ξεκινά την εισαγωγή του κειμένου από εκεί.
- s ανταλλάσσει το γράμμα κάτω από τον κέρσορα με τον επόμενο χαρακτήρα που θα πατήσουμε. Μετά την αλλαγή ο editor συνεχίζει να βρίσκεται σε insert mode.

Όταν είμαστε σε insert mode και για να επανέλθουμε σε command mode πατάμε Esc.

Εντολές διαγραφής.

- x διαγραφή του χαρακτήρα που βρίσκεται κάτω από τον κέρσορα.
- X διαγραφή του χαρακτήρα που βρίσκεται πριν από τον κέρσορα.
- D διαγραφή από το σημείο που βρίσκεται ο κέρσορας μέχρι το τέλος της γραμμής.
- dd διαγραφή της γραμμής που βρίσκεται ο κέρσορας.
- dw διαγραφή της λέξης πάνω στην οποία βρίσκεται ο κέρσορας.
- db διαγραφή της λέξης πριν τον κέρσορα.
- #dd διαγράφει # γραμμές και τις αποθηκεύει προσωρινά στην μνήμη μέχρι να δοθεί εντολή επικόλλησης.
- d# διαγράφει #+1 γραμμές και τις αποθηκεύει προσωρινά στην μνήμη μέχρι να δοθεί εντολή επικόλλησης.

Εντολή αντιγραφής

- yy (yank)'αντιγράφει' την γραμμή στην οποία βρίσκεται ο κέρσορας. Η γραμμή αυτή μπορεί να επικολληθεί με την εντολή p (put). Μπορεί να χρησιμοποιηθεί και με ακέραιο αριθμό μπροστά από την εντολή για αντιγραφή παραπάνω από μιας γραμμών.

Εντολή εισαγωγής

Η εντολή εισαγωγής εισάγει στο κείμενο είτε κείμενο που έχει μόλις διαγραφεί είτε κείμενο που έχει αντιγραφεί.

P εισαγωγή πριν από τον κέρσορα.

p εισαγωγή μετά από τον κέρσορα.

Εντολές εύρεσης

? βρίσκει λέξη ή κείμενο ψάχνοντας προς τα πίσω.

/ βρίσκει λέξη ή κείμενο ψάχνοντας προς τα μπροστά.

f βρίσκει ένα χαρακτήρα στην γραμμή που βρίσκεται ο κέρσορας ψάχνοντας από τον κέρσορα και μετά.

F βρίσκει ένα χαρακτήρα στην γραμμή που βρίσκεται ο κέρσορας ψάχνοντας από τον κέρσορα και πίσω.

t βρίσκει ένα χαρακτήρα στην γραμμή που βρίσκεται ο κέρσορας ψάχνοντας από τον κέρσορα και μετά και σταματάει ένα χαρακτήρα πριν από τον ζητούμενο.

T βρίσκει ένα χαρακτήρα στην γραμμή που βρίσκεται ο κέρσορας ψάχνοντας από τον κέρσορα και μετά και σταματάει ένα χαρακτήρα πριν από τον ζητούμενο.

Διάφορες εντολές

. επαναλαμβάνει την τελευταία εντολή.

u αναιρεί την τελευταία εντολή.

xr ανταλλάσσει τον πρώτο χαρακτήρα με τον δεύτερο.

Μπορούν να χρησιμοποιηθούν και οι εντολές που ίσχυαν στον κειμενογράφο ex. Ο τρόπος εισαγωγής τους είναι πατώντας “:” όσο βρισκόμαστε σε command mode. Για να τερματίσουμε την εισαγωγή της εντολής πατάμε Enter ή Esc. Κάποιες πολύ χρήσιμες εντολές είναι οι

:w σώζει τις αλλαγές που έχουν γίνει στο κείμενο.

:q αυτή είναι η εντολή εξόδου από το vi.

:q! έξοδος από το vi χωρίς αποθήκευση των αλλαγών που έχουν γίνει.

:wq σώζει τις αλλαγές και βγαίνει από το vi.

- :# μετάβαση στην γραμμή #.
- :\$ μετάβαση στο τέλος του κειμένου.

Αρχεία .bashrc

Περιέχουν τις προσωπικές μας προτιμήσεις π.χ.

```
# my personal aliases
alias cp='cp -v -i'
alias rm='rm -i'
alias mv='mv -i'
```

Η πρώτη γραμμή που αρχίζει με το # είναι σχόλιο, οι επόμενες ορίζουν συντομεύσεις εντολών (alias).”

Αμα αλλάξουμε κάτι στο αρχείο αυτό για να το ενεργοποιήσουμε χωρίς να χρειαστεί να από/επανα συνδεθούμε στον ΗΥ γράφουμε:

```
Source .bashrc
```

File Transfer Protocol, FTP

Το FTP είναι ένα πολύ χρήσιμο εργαλείο για την μεταφορά αρχείων μεταξύ υπολογιστών.

Η χρήση του μπορεί να γίνει είτε μέσω ειδικών προγραμμάτων, είτε μέσω του command prompt των Windows.

Command Prompt

Για να ανοίξουμε ένα command prompt πατάμε Start-> Run και στο παράθυρο που εμφανίζεται γράφουμε cmd.

Ανοίγει ένα παράθυρο που μοιάζει με το κέλυφος του Linux. Από αυτό το παράθυρο μπορούμε να περιηγηθούμε στα αρχεία μας με παρόμοιο τρόπο με αυτόν του Linux. Δηλαδή ισχύουν εντολές όπως το cd, το mkdir και άλλες. Η εντολή ls γίνεται dir στο command prompt.

Αν του δώσουμε την εντολή ftp [host] τότε προσπαθεί να συνδεθεί στον host. Στην περίπτωση μας ο host είναι ο andromeda.chemistry.uoc.gr

Μόλις συνδεθεί στον server ζητάει όνομα χρήστη και αμέσως μετά κωδικό πρόσβασης. Αν δώσουμε σωστά τα στοιχεία αυτά, τότε η κονσόλα γίνεται κονσόλα ftp και είναι κάπως έτσι:

```
ftp>
```

Έτσι έχουμε ανοίξει ένα δίαυλο επικοινωνίας μεταξύ των 2 μηχανημάτων για μεταφορά και διαχείριση αρχείων αλλά δεν μπορούμε να τα επεξεργαστούμε (π.χ. με VI). Ισχύουν οι εντολές περιήγησης του Linux (cd, cd .., pwd κλπ) για να μετακινούμαστε μέσα στο σύστημα. Με την εντολή dir μπορούμε να δούμε τα περιεχόμενα του φακέλου που βρισκόμαστε.

Αν θέλουμε **να δούμε ή να περιηγηθούμε στους φακέλους του υπολογιστή στον οποίο δουλεύουμε** (τερματικό) χρησιμοποιούμε την εντολή lcd ή lcd [folderpath] αντίστοιχα όπου folderpath η πλήρης διαδρομή του φακέλου στον οποίο θέλουμε να πάμε, μέσα σε εισαγωγικά (π.χ. lcd "C:\Documents and Settings\user\Desktop"). Ο φάκελος στον οποίο βρισκόμαστε τοπικά είναι και ο φάκελος που είναι άμεσα συνδεδεμένος με τον φάκελο που βρισκόμαστε στο απομακρυσμένο μηχάνημα Linux.

Όταν βρούμε το αρχείο που θέλουμε να μεταφέρουμε στον υπολογιστή μας, γράφουμε

```
get filename [Enter]
```

οπότε το αρχείο μεταφέρεται στον υπολογιστή μας στον φάκελο που έχουμε επιλέξει με το lcd [folderpath]

Για να μεταφέρουμε ένα αρχείο από τον υπολογιστή μας στον server χρησιμοποιούμε την εντολή put

```
put filename [Enter]
```

μεταφέρει το αρχείο filename από τον υπολογιστή μας στον server στον φάκελο που έχουμε επιλέξει με το `cd [folderpath]`.

Υπάρχουν και οι εντολές `mget` και `mput` που η διαφορά τους από τα απλά `get` και `put` είναι ότι ζητάει για επιβεβαίωση πριν τη μεταφορά.

`mget *` για να πάρουμε όλα τα αρχεία του φακέλου.

`mput *` για να βάλουμε όλα τα αρχεία του τοπικού φακέλου στον απομακρυσμένο.

και στις 2 παραπάνω περιπτώσεις μας ρωτάει ανά αρχείο αν θέλουμε να το μεταφέρουμε. Απαντάμε `y` (yes) ή `n` (no)

επίσης μπορούμε να επιλέξουμε μερικά αρχεία π.χ. όλα όσα έχουν κατάληξη `.f90`

```
mget *.f90
```

Για την μεταφορά αρχείων με συγκεκριμένα format (word, excel κλπ), πριν την μεταφορά δίνουμε την εντολή `bin` (binary transfer).

Για να κλείσουμε την σύνδεση με τον server γράφουμε

```
bye
```

```
ή logout.
```

File Zilla

Στους υπολογιστές του εργαστηρίου είναι εγκατεστημένο το πρόγραμμα filezilla που χρησιμοποιείται για FTP και παρέχει γραφικό περιβάλλον.

Όταν ανοίξουμε το πρόγραμμα, στην πάνω μεριά της οθόνης μπορούμε να γράψουμε τον host, το όνομα χρήστη και τον κωδικό και να συνδεθούμε στον server.

Στην αριστερή μεριά της οθόνης βλέπουμε τους δίσκους και τα αρχεία που υπάρχουν στον υπολογιστή μας και στην δεξιά μεριά βλέπουμε τα αρχεία του server. Μπορούμε να περιηγηθούμε στους φακέλους ακριβώς όπως περιηγούμαστε σε φακέλους των Windows. Όταν βρούμε το αρχείο που θέλουμε να μεταφέρουμε, πατάμε πάνω του με το ποντίκι και το τραβάμε στον φάκελο που θέλουμε να το μεταφέρουμε.

Διαγράμματα ροής και εισαγωγή στη fortran 90

Πρόγραμμα – Μεταγλωττιστής

Τα αριθμητικά προγράμματα επιτρέπουν

- Περίπλοκους υπολογισμούς (π.χ. προσομοιώσεις διασποράς και χημείας ρύπων στο περιβάλλον)
- Οργάνωση δεδομένων και παρουσίασής τους
- Αυτοματοποίηση και έλεγχο μηχανημάτων και καταγραφή δεδομένων σε πραγματικό χρόνο.

Ο προγραμματισμός απαιτεί **δομημένη σκέψη** – αναγκαία για την επίλυση προβλημάτων και χωρίς τον ΗΥ. Ο ΗΥ απλά επιταχύνει την επίλυση.

Πρόγραμμα (ή λογισμικό) είναι μια **σειρά εντολών** που πρέπει να ακολουθήσει ο ΗΥ για την επίλυση ενός προβλήματος, μεταφορά δεδομένων, επεξεργασία τους και παρουσίασή τους. Το πρόγραμμα γράφεται σε μία γλώσσα προγραμματισμού. **Γλώσσα προγραμματισμού** είναι ένα **σύνολο κανόνων** που πρέπει να εφαρμοστούν κατά τη συγγραφή ενός προγράμματος.

Λειτουργικό σύστημα είναι το πρόγραμμα που επιτρέπει τον έλεγχο του υπολογιστή σαν σύνολο.

Compiler: Μεταγλωττιστής προγραμμάτων είναι ένα εκτελέσιμο πρόγραμμα, διαφορετικό για κάθε γλώσσα προγραμματισμού και για κάθε ΗΥ που βασίζεται σε διαφορετικό λειτουργικό σύστημα ή μικροεπεξεργαστή, που **μετατρέπει σε γλώσσα μηχανής** τον κώδικα που έχει γραφτεί στην εν λόγω γλώσσα προγραμματισμού. Έτσι δημιουργείται το **εκτελέσιμο** πρόγραμμα (executable), το οποίο **εκτελούμε** στον υπολογιστή.

Πως δημιουργούμε ένα πρόγραμμα ?

1. *Κατάστρωση προβλήματος* (άσκηση)
(τι ξέρουμε ≡ δεδομένα εισόδου –input data,
τι δεν ξέρουμε και θέλουμε να υπολογίσουμε ≡ δεδομένα εξόδου - output)
2. *Ανάπτυξη αλγόριθμου* (f) – ΔΗΜΙΟΥΡΓΙΑ ΔΙΑΓΡΑΜΜΑΤΩΝ ΡΟΗΣ
(σειρά υπολογιστικών βημάτων που θα συσχετίσουν τα δεδομένα εισόδου με τη λύση του προβλήματος δηλ. με τα δεδομένα εξόδου

$$Y=f(x)$$

Περιγραφή των υπολογισμών με διάγραμμα ροής

Δηλαδή γράφουμε μία - μία τις πράξεις (διεργασίες) που πρέπει να κάνουμε για να υπολογίσουμε το f(x).

Οι πράξεις αυτές (υπολογιστικά βήματα) μπορούν να εκτελούνται διαδοχικά, ή να επαναλαμβάνονται ή να εκτελούνται επιλεκτικά με βάση κάποιας συνθήκης) δηλαδή

- να ακολουθούν η μία την άλλη

- να γίνονται υπό συνθήκες
 - να επαναλαμβάνονται –είτε για ένα συγκεκριμένο αριθμό επαναλήψεων που τον ορίζουμε – είτε μέχρι να ικανοποιηθεί μια συνθήκη (πχ. Συνθήκη σύγκλισης σε επαναληπτική μέθοδο)
3. Μετατροπή του αλγόριθμου σε πρόγραμμα –ΔΗΜΙΟΥΡΓΙΑ ΚΩΔΙΚΑ
(ακολουθώντας τους κανόνες της γλώσσας προγραμματισμού)
 4. Διόρθωση σφαλμάτων – ΔΗΜΙΟΥΡΓΙΑ ΕΚΤΕΛΕΣΙΜΟΥ

Για να πάρουμε τα αποτελέσματα του προγράμματος –δεδομένα εξόδου - πρέπει να δώσουμε εντολή στον ΗΥ να τρέξει (εκτελέσει) το εκτελέσιμο αρχείο.

Παράδειγμα προγράμματος

```

program try1
implicit none
real :: x1, x2, x3, l
integer :: i, k
real,dimension(3) :: a, T
real,dimension(31) :: s
data T /230., 270., 298./
open(unit=20,file='data')
do i=1,3
a(i)=tanh(318.8/T(i))
print*, a(i)
do k=1,31
l=200.+(k-1)*5.
x1=7.34*exp(-68.6*a(i)*((log(227.6/l))**2))
x2=43.5*exp(-123.6*a(i)*((log(372.5/l))**2))
x3=11.2*exp(-84.8*a(i)*((log(442.4/l))**2))
s(k)=(a(i)**0.5)*(x1+x2+x3)
write(20,*) l, s(k)
enddo
enddo
close(20)
end program

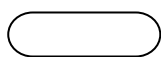
```

Ανάπτυξη αλγορίθμου - ΔΗΜΙΟΥΡΓΙΑ ΔΙΑΓΡΑΜΜΑΤΩΝ ΡΟΗΣ

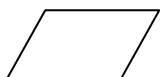
Η σωστή επιλογή/δημιουργία του αλγορίθμου αποτελεί το κλειδί για την επιτυχία ενός κώδικα. Η επιτυχία του κώδικα κρίνεται τόσο από την **ορθότητα των αποτελεσμάτων** ανεξαρτήτως δεδομένων εισόδου, τη **συντομία της λογικής διεργασίας επίλυσης** του προβλήματος, την **οργανωμένη παρουσίαση** των αποτελεσμάτων, όσο και από την **ταχύτητα εκτέλεσης** του προγράμματος.

Η αναπαράσταση των υπολογιστικών βημάτων ενός αλγορίθμου με διάγραμμα ροής μας επιτρέπει τη σύντομη και περιεκτική περιγραφή τους και διευκολύνει τη δημιουργία του κώδικα (προγράμματος) ανεξαρτήτως της χρησιμοποιούμενης γλώσσας προγραμματισμού. Τα διαγράμματα ροής επιτρέπουν να ορίσουμε τα δεδομένα εισόδου, τις πράξεις που πρέπει να εκτελεστούν είτε διαδοχικά είτε επαναληπτικά είτε υπό συγκεκριμένες προϋποθέσεις (π.χ. μέχρι να ικανοποιηθεί μια συνθήκη όπως η συνθήκη σύγκλισης σε επαναληπτική μέθοδο).

Για την απεικόνιση αλγορίθμων με **διαγράμματα ροής** χρησιμοποιούνται **γεωμετρικά σχήματα** μερικά – τα πιο ευρέως χρησιμοποιούμενα παρατίθενται παρακάτω:



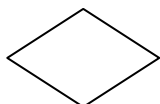
Αρχή / τέλος



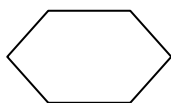
Είσοδος/έξοδος (δεδομένων)



Επεξεργασία (διαδοχικές πράξεις)



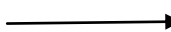
Απόφαση (επιλογή κατεύθυνσης ροής με βάση τη συνθήκη που αναγράφεται – σωστό ή λάθος)



Προετοιμασία (επικεφαλίδα επανάληψης)



Σύνδεση πολλών γραμμών ροής



Γραμμή ροής

Δημιουργία και επαλήθευση κώδικα – Δημιουργία εκτελέσιμου και χρήση του

Από τη στιγμή λοιπόν που έχουμε δημιουργήσει το διάγραμμα ροής που επιτρέπει τη ΣΩΣΤΗ επίλυση του προβλήματος, είμαστε έτοιμοι να γράψουμε τον αντίστοιχο κώδικα, μετατρέποντας τον αλγόριθμο σε πρόγραμμα.

Σωστή λειτουργία του προγράμματος δεν σημαίνει ορθότητα αποτελεσμάτων – αυτή όπως προαναφέραμε απαιτεί σωστή επίλυση του προβλήματος δηλαδή σωστό αλγόριθμο / διάγραμμα ροής (προϋγούμενο βήμα). Παρακάτω περιγράφονται εντολές και κανόνες για τη δημιουργία κώδικα σε fortran90.

Ο κώδικας αυτός πρέπει κατόπιν να **ελεγχθεί** για τυχόν συντακτικά λάθη και να **μεταγλωττιστεί** σε γλώσσα μηχανής (εκτελέσιμο αρχείο).

Για τα προγράμματα γραμμένα σε fortran 90 αυτή η διαδικασία γίνεται με τον compiler f90 με την εντολή **f90** που συντάσσεται ως ακολούθως:

f90 [name].f90

οπότε αν δεν υπάρχουν λάθη δημιουργεί εκτελέσιμο με το όνομα **a.out**

Ή

f90 -o [name].exe [name].f90

οπότε το εκτελέσιμο που δημιουργείται ονομάζεται **[name].exe**

Για να τρέξει το πρόγραμμα πρέπει να ζητήσουμε στον ΗΥ να εκτελέσει το [name].exe και αυτό γίνεται με την εντολή

./[name].exe

ή

./a.out

Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ FORTRAN

Το αλφάβητο (όλα με λατινικούς χαρακτήρες – **δε διαφοροποιεί μικρά με κεφαλαία** – πράγμα που γίνεται στη LINUX)

Λέξεις → εκφράσεις (από το χρήστη)

Εντολές fortran (προκαθορισμένες λέξεις με καθορισμένη σημασία και σύνταξη π.χ

DO IF READ PRINT)

προτάσεις συνδυασμός εντολών με τη βοήθεια κανόνων

εκφράσεις συνδυασμός λέξεων με τη βοήθεια κανόνων

ΠΙΝΑΚΑΣ 3.2 (Από Ματαράς και Κουτελιέρης – fortran 90/95 για επιστήμονες και μηχανικούς)

Λέξεις		παράδειγμα	
Εντολές	Δεσμευμένες λέξεις της fortran που χρησιμοποιούνται για να δημιουργήσουμε προτάσεις που καθορίζουν το περιβάλλον του προγράμματος, ελέγχουν τον τρόπο εκτέλεσης, και επικοινωνούν αμφίδρομα με τον χρήστη.	DO IF READ	
Μεταβλητές	Λέξεις που δημιουργούμε για να αναπαραστήσουμε φυσικά μεγέθη που <u>αλλάζουν τιμή</u> κατά την διάρκεια επίλυσης του προβλήματος μας.	TEMP για την θερμοκρασία	
Σταθερές	Επώνυμες: Λέξεις που δημιουργούμε για να αναπαραστήσουμε συνήθως φυσικά μεγέθη που <u>δεν αλλάζουν τιμή</u> κατά την διάρκεια επίλυσης του προβλήματος μας.	pi η σταθερά n=3,14	
	Ρητές: τιμές	3, 1.24, 5000	
Τελεστές	Εκτελούν πράξεις μεταξύ Δεδομένων.	Ανάθεσης	=
		Αριθμητικοί	** , * , / , + , -
		αλφαριθμητικοί	//
		Λογικοί	.AND. , .NOT. , .OR. , .EQV. , .NEQV.
		Σύγκρισης	<= , < , == , /= , >= , > .LT. , .LE. , .EQ. , .NEQ. , .GE. , .GT.
Ετικέτες	Αριθμοί, με 1 έως 5 ψηφία, που σηματοδοτούν την θέση μιας έκφρασης.	100, 20	
Διαχωριστές	Εμπεριέχουν ή διαχωρίζουν τα στοιχεία μιας λίστας λέξεων.	π.χ / , / , () , (//)	

Ελεύθερη σύνταξη γραμμένη σε αρχεία με κατάληξη .f90 (π.χ. askisi1.f90)

1. Ο κώδικας γράφεται σε γραμμές μέχρι 132 χαρακτήρες
2. Τα κενά χωρίζουν λέξεις μεταξύ τους – πολλά συνεχόμενα κενά διαβάζονται ως ένα κενό
3. ! στην αρχή μιας γραμμής τότε όλη η γραμμή θεωρείται σχόλιο και δεν εκτελείται
4. ! σε κάποιο άλλο σημείο μιας γραμμής – ότι ακολουθεί το ! θεωρείτε σχόλιο
5. Η γραμμή μπορεί να αρχίζει από οποιαδήποτε θέση (στήλη)
6. Μπορεί να περιέχει μόνο μια ετικέτα στην αρχή της
7. Μια έκφραση συνεχίζει στην επόμενη γραμμή αν στο τέλος της γραμμής έχει &
8. Μια γραμμή μπορεί να περιέχει πολλές προτάσεις ή εκφράσεις αρκεί να χωρίζονται με ;

Δομή προγράμματος

Program [name] (αρχή προγράμματος)

IMPLICIT NONE

(λέμε ότι οποιαδήποτε παράμετρος ή μεταβλητή χρησιμοποιηθεί στο πρόγραμμα πρέπει να έχει οριστεί στην αρχή του υποπρογράμματος αλλιώς μας βγάζει λάθος ο μεταγλωττιστής κατά τη διαδικασία ελέγχου σφαλμάτων για τη δημιουργία του εκτελέσιμου)

! ΔΗΛΩΣΕΙΣ μεταβλητών , παραμέτρων

INTEGER :: A, B ! A input B output

REAL ::

REAL, DIMENSION(8):: ακολουθούν πραγματικοί αριθμοί σε πίνακες 8 στοιχείων

LOGICAL::

CHARACTER ::

COMPLEX ::

REAL, PARAMETER :: pi=3.14

ΣΤΟ ΤΕΛΟΣ ΤΩΝ ΔΗΛΩΣΕΩΝ Μπορούμε να βάλουμε δεδομένα εισόδου π.χ.:

DATA x,y,z/1,2,3/ !αντιστοιχεί στο x=1, y=2, z=3

DATA x,y,z,w/2*1,2,3/ ! αντιστοιχεί σε x=1, y=1, z=2, w=3

DATA (x(i),i=1,3)/2*5,3/ !! αντιστοιχεί σε x(1)=5, x(2)=5, x(3)=3

! ΕΚΦΡΑΣΕΙΣ - ΥΠΟΛΟΓΙΣΜΟΙ

Πχ Διάφορες πράξεις – προσοχή στην αριστερή μεριά πάντα μόνο μία μεταβλητή

PRINT *

WRITE *

end [name] (τέλος προγράμματος) (ΑΠΑΡΑΙΤΗΤΗ ΕΝΤΟΛΗ)

Web site με πληροφορίες <http://h18009.www1.hp.com/fortran/docs/>

IMPLICIT type (a[, a]...)[, type (a[, a]...)]...

IMPLICIT NONE

type

Is a data type specifier (CHARACTER*(*) is not allowed).

! ΔΗΛΩΣΕΙΣ μεταβλητών , παραμέτρων

INTEGER

INTEGER([KIND=]*n*)

INTEGER(*n*)

n

Is kind 1, 2, 4, or 8.

REAL

REAL([KIND=]*n*)

REAL(*n*)

DOUBLE PRECISION

n

Is kind 4, 8, or 16. Kind 16 is only available on OpenVMS, Tru64 UNIX, and Linux systems.

COMPLEX

COMPLEX([KIND=]*n*)

COMPLEX(*n*)

DOUBLE COMPLEX

n

Is kind 4, 8, or 16. Kind 16 is only available on OpenVMS, Tru64 UNIX, and Linux systems.

LOGICAL

LOGICAL([KIND=]*n*)

LOGICAL(*n*)

n

Is kind 1, 2, 4, or 8.

If a kind parameter is specified, the logical constant has the kind specified. If no kind parameter is specified, the kind of the constant is *default logical*.

Logical Constants

A *logical constant* represents only the logical values true or false, and takes one of the following forms:

.TRUE.[*k*]

.FALSE.[*k*]

k

Is the optional kind parameter: 1 for LOGICAL(1), 2 for LOGICAL(2), 4 for LOGICAL(4), or 8 for LOGICAL(8). It must be preceded by an underscore (`_`).

CHARACTER

CHARACTER([KIND=]*n*)

CHARACTER([LEN=]*len*)

CHARACTER([LEN=]*len*,[KIND=]*n*)

CHARACTER(KIND=*n*,[LEN=]*len*)

CHARACTER**len*[,]

n

Is kind 1.

CHARACTER*20 TITLE

COMMON *[/[cname]/] var-list* [[,] *[/[cname]/ var-list]*...

cname

Is the name of the common block. The name can be omitted for blank common (//).

var-list

Is a list of variable names, separated by commas.

INTEGER A(20)

REAL Y(20)

COMMON /QUANTA/ A, Y

DATA *var-list /c-list*[[,] *var-list /c-list*]/...

var-list

Is a list of variables or implied-do lists, separated by commas.

ΕΝΤΟΛΗ PARAMETER

Η εντολή PARAMETER τοποθετείται στο τμήμα δηλώσεων του προγράμματος (στην αρχή) και έχει τη γενική μορφή

Δήλωση τύπου, PARAMETER :: όνομα σταθεράς = τιμή, ...
--

Σημ: Οποιαδήποτε σταθερά δηλώνεται με την ιδιότητα ή την εντολή PARAMETER **δεν μπορεί να αλλάξει κατά την διάρκεια του προγράμματος.**

ΕΝΤΟΛΗ DIMENSION –δίασταση (θέσεις πίνακα)

DIMENSION A(10,10)

DATA A/100*1.0/ ! initialization by name

```
DATA A(1,1), A(10,1), A(3,3) /2*2.5, 2.0/ ! initialization by
element
```

```
DATA ((A(I,J), I=1,5,2), J=1,5) /15*1.0/ ! initialization by
implied-do list
```

Στην FORTRAN 90 για την δήλωση ενός πίνακα πχ A(10,10,5) ανάλογα με τον τύπο των στοιχείων του (πχ, REAL, INTEGER, ...) γράφουμε :

REAL (ή INTEGER, ή ...), DIMENSION (10,10,5) :: A

Σημείωση: Η εντολή DATA είναι μη εκτελέσιμη και πρέπει να εμφανίζεται μετά τις δηλώσεις των δεδομένων και πριν από οποιαδήποτε εκτελέσιμη εντολή. Η διαφορά της από την PARAMETER είναι ότι οι μεταβλητές που παίρνουν αρχική τιμή από την DATA παραμένουν μεταβλητές. Παρόλα αυτά η εντολή DATA δεν μπορεί να χρησιμοποιηθεί δεύτερη φορά για να μεταβάλει την τιμή ενός πίνακα.

Εκφράσεις και εντολές

Τελεστές

προτεραιότητα πράξεων - **αριθμητικοί τελεστές**

```
**  ύψωση σε δύναμη
*   Πολλαπλασιασμός
/   διαίρεση
+   πρόσθεση
-   αφαίρεση
```

Ξέρουμε ότι, οι πολλαπλασιασμοί και διαιρέσεις είναι πράξεις ισοδύναμης μεταξύ τους προτεραιότητας και μεγαλύτερης προτεραιότητας από ότι οι προσθέσεις και αφαιρέσεις .

Η σειρά εκτέλεσης στις πράξεις ίδιας προτεραιότητας είναι από αριστερά προς τα δεξιά

Εξαίρεση αποτελεί ο αριθμητικός τελεστής ** (δύναμη), καθόσον διαδοχικοί τελεστές ** εκτελούνται από τα δεξιά προς τα αριστερά.

Αν θέλουμε να αλλάξουμε τη σειρά εκτέλεσης των πράξεων χρησιμοποιούμε παρενθέσεις.

Επίση και οι λογικοί τελεστές εκτελούνται από αριστερά προς τα δεξιά

Λογικοί τελεστές:

.NOT.

.AND.

.OR.

.EQV.

.NEQV.

Χρήση A[τελεστής]B

Τελεστές συσχέτισης A με B

==	.EQ.	A.EQ.B ή A=B
/=	.NE.	
<	.LT.	
<=	.LE.	
>=	.GE.	
>	.GT.	

Κύκλος επαναλήψεων DO

```

DO I = 1, N                                ! Block DO
    TOTAL = TOTAL + B(I)
ENDDO

DO 20 I = 1, N                               ! Nonblock DO
20 TOTAL = TOTAL + B(I)

```

DO ... CONTINUE

```

DO 30 I = 1, N

    DO 30 J = 1 + I, N

        RESULT(I,J) = 1.0 / REAL(I + J)
        RESULT2(I,J)=RESULT(I,J)+1.0

30 CONTINUE

```

CONTINUE

```

    DO 150 I = 1, 40
40  Y = Y + 1
    Z = COS(Y)
    PRINT *, Z
    IF (Y .LT. 30) GO TO 150
    GO TO 40
150 CONTINUE

```

Επιλογή κατεύθυνσης ροής προγράμματος - ερώτηση IF

Αν μια συνθήκη ικανοποιείται ή όχι

The IF construct takes the following form:

```
[name:] IF (expr) THEN
```

```
block
```

```
[ELSE IF (expr) THEN [name]
```

```
block]...
```

```
[ELSE [name]
```

```
block]
```

```
END IF [name]
```

```
name
```

Is the name of the IF construct.

```
expr
```

Is a scalar logical expression enclosed in parentheses.

```
block
```

Is a sequence of zero or more statements or constructs.

Σύνταξη και αντίστοιχη ροή ελέγχου σε εντολές IF

Construct	Flow of Control
IF (e) THEN block END IF	
IF (e) THEN block ₁ ELSE block ₂ END IF	
IF (e ₁) THEN block ₁ ELSE IF (e ₂) THEN block ₂ END IF	
IF (e ₁) THEN block ₁ ELSE IF (e ₂) THEN block ₂ ELSE IF (e ₃) THEN block ₃ ELSE block ₄ END IF	

ZK-0617-GE

The IF statement takes the following form:

IF (*expr*) *stmt*

expr

Is a scalar logical expression enclosed in parentheses.

stmt

Is any complete, unlabeled, executable Fortran statement

IF (J.GT.4 .OR. J.LT.1) **GO TO** 250

DO I=1,10

```

DO J=1,5
  IF (TABLE(I,J) .GE. 100.) THEN
    FORCHR(J) = FBIG
  ELSE IF (TABLE(I,J) .GT. 0.1) THEN
    FORCHR(J) = FMED
  ELSE
    FORCHR(J) = FSML
  END IF
END DO
END DO

```

Υποπρόγραμμα (Υπορουτίνα) - CALL

CALL *sub* [[*a-arg* [, *a-arg*]...]]

sub

Is the name of the subroutine subprogram or other external procedure, or a dummy argument associated with a subroutine subprogram or other external procedure.

The following table shows the statements that define these subprograms, and how control is transferred to the subprogram:

Subprogram	Defining Statements	Control Transfer Method
Function	FUNCTION or ENTRY	Function reference ¹
Subroutine	SUBROUTINE or ENTRY	CALL statement ²
Statement function	Statement function definition	Function reference

¹ A function can also be invoked by a defined operation (see [Section 8.9.4](#)).

² A subroutine can also be invoked by a defined assignment (see [Section 8.9.5](#)).

Ένα υποπρόγραμμα καλείται στο κυρίως πρόγραμμα με την εντολή `call`.

ΣΥΝΑΡΤΗΣΗ

```

FUNCTION ROOT(A)
  X = 1.0
  DO
    EX = EXP(X)
    EMINX = 1./EX
    ROOT = X - ((EX+EMINX) *.5+COS(X)-A) / ((EX-EMINX) *.5-SIN(X))
  
```

```

      IF (ABS((X-ROOT)/ROOT) .LT. 1E-6) RETURN
      X = ROOT
    END DO
  END

```

In the preceding example, the following formula is calculated repeatedly until the difference between X_i and X_{i+1} is less than $1.0E-6$:

$$X_{i+1} = X_i - \frac{\cosh(X_i) + \cos(X_i) - A}{\sinh(X_i) - \sin(X_i)}$$

STATEMENT FUNCTIONS

fun ([*d-arg* [, *d-arg*]...]) = *expr*

fun

Is the name of the statement function.

d-arg

Is a dummy argument. A dummy argument can appear only once in any list of dummy arguments, and its scope is local to the statement function.

expr

Is a scalar expression defining the computation to be performed.

CONTAINS

Μιά συνάρτηση (**function**) γράφεται στο τέλος του προγράμματος πριν το END εισαγόμενη εκεί με την εντολή **CONTAINS** π.χ.

```
PROGRAM main
```

```
  IMPLICIT NONE
```

```
  REAL :: a,b,c
```

```
  .
```

```
  .
```

```
  .
```

```
  mainsum=add
```

```
  .
```

```
  .
```

```
  .
```

```
CONTAINS
```

```
  FUNCTION add()
```

```
    IMPLICIT NONE
```

```
    REAL :: add !a,b,c,defined in `main`
```

```
    add=a+b+c
```

```
  END FUNCTION add
```

```
END PROGRAM main
```

Μια συνάρτηση χρησιμοποιείται στο κυρίως πρόγραμμα με το όνομά της

Μεταβλητές που ορίζονται στο κύριο πρόγραμμα παραμένουν ορισμένες και στις εσωτερικές διεργασίες εκτός αν ξαναοριστούν εκεί. Καλό είναι να ξαναορίζουμε όλες τις μεταβλητές σε κάθε υποπρόγραμμα για αποφυγή λαθών.

READ- WRITE – PRINT- FORMAT

print *, X ! γράφει στην οθόνη ή στο αρχείο εξόδου στη θέση μνήμης 6 τη τιμή του X

write (unit_number, n_format) X

γράφει στο αρχείο που έχουμε ανοίξει στη θέση μνήμης του HY 'unit' με τον τρόπο που υποδεικνύουμε στο FORMAT που αντιστοιχεί στον αριθμό n, τη τιμή του X

Αν αντί για αριθμό unit βάλουμε * τότε γράφει στη θέση 6 (default write unit - οθόνη)

Το unit μπορεί να πάρει τιμές από 1 ως 99, εκτός 5

Αν αντί για αριθμό n που χαρακτηρίζει την μορφή με την οποία θέλουμε να γράφουμε τα δεδομένα μας βάλουμε *, το πρόγραμμα γράφει τα δεδομένα στη σειρά ως δεκαδικούς αριθμούς.

read (unit_number, n_format) X

διαβάζει από το αρχείο που έχουμε ανοίξει στη θέση μνήμης του HY 'unit_number' με τον τρόπο που υποδεικνύουμε στο FORMAT που αντιστοιχεί στον αριθμό n_format, τη τιμή του X

Αν αντί για αριθμό unit βάλουμε * τότε διαβάζει από τη θέση 5 (default read unit - οθόνη)

Το unit μπορεί να πάρει τιμές από 1 ως 99, εκτός 6

Αν αντί για αριθμό n που χαρακτηρίζει την μορφή με την οποία θέλουμε να διαβάσουμε τα δεδομένα μας βάλουμε *, το πρόγραμμα διαβάζει θεωρώντας ότι τα δεδομένα χωρίζονται μεταξύ τους με ένα κενό

n FORMAT(...)

Η εντολή FORMAT συνοδεύει τις εντολές READ και WRITE (ή PRINT) και καθορίζει τον τρόπο με τον οποίο διαβάζεται ή τυπώνεται ένα σύνολο δεδομένων. Η εντολή FORMAT συντάσσεται με τον παρακάτω τρόπο:

n FORMAT(...)

Όπου n είναι ο αριθμός της εντολής στην οποία αναφέρεται το READ ή το WRITE. Το όρισμα της εντολής εξαρτάται από τον τύπο των δεδομένων που θέλουμε να χρησιμοποιήσουμε:

α) Αν είναι κενό διάστημα, τότε το όρισμα είναι της μορφής:

wX

όπου w ο αριθμός των θέσεων που θέλουμε να διαβάσουμε ή να τυπώσουμε και X σύμβολο που υποδηλώνει κενό διάστημα. Έτσι αν θέλουμε να διαβάσουμε 8 κενές θέσεις από το αρχείο 1 θα γράψουμε:

READ(1,100) KENO
100 FORMAT(8X)

β) Αν είναι ακέραιος αριθμός το όρισμα είναι της μορφής :

/w

Αν θέλουμε να γράψουμε ένα τριψήφιο ακέραιο αριθμό στο αρχείο 1 θα γράψουμε στο πρόγραμμά μας:

WRITE(1,200) INT
200 FORMAT(I3)

γ) Για πραγματικούς αριθμούς το όρισμα είναι της μορφής:

Fw.d

Στην περίπτωση αυτή **d** είναι ο αριθμός των δεκαδικών ψηφίων, ενώ **w** ο συνολικός αριθμός των ψηφίων, συμπεριλαμβανομένης και της τελείας. Έτσι, αν θέλουμε να διαβάσουμε από το αρχείο 1 τον αριθμό 33.5678, θα γράψουμε:

```
      READ(1,101) REAL
101   FORMAT( F7.4 )
```

δ) Για αριθμούς σε εκθετική μορφή το όρισμα είναι της μορφής:

Ew.d

Εδώ d είναι ο αριθμός των ψηφίων του εκθέτη και w είναι πάλι ο συνολικός αριθμός των ψηφίων, μόνο που συμπεριλαμβάνεται η υποδιαστολή, η βάση του εκθέτη και το πρόσημο αυτού. Έτσι, αν θέλουμε να διαβάσουμε τον αριθμό 0.45E-04 στο αρχείο 30 θα γράψουμε:

```
      WRITE(30,300) EXPON
300   FORMAT( E8.2 )
```

Ε) Αν θέλουμε να γράψουμε χαρακτήρες π.χ. λέξη, πρόταση, τίτλο πίνακα το όρισμα είναι της μορφής:

```
      WRITE(2,400) CHARACTER
400   FORMAT(A160)
```

Το format μπορούμε να το εισαγάγουμε και μέσα στην εντολή WRITE ή READ αντικαθιστώντας τον αριθμό του format με '()' με το αναλυτικό περιεχόμενο του format μέσα στη παρένθεση

Π.χ.

```
      WRITE(2,'(A160)') CHARACTER
```

OPEN – CLOSE

Για να ανοίξουμε/ φορτώσουμε ένα αρχείο στη θέση μνήμης unit του υπολογιστή χρησιμοποιούμε την εντολή OPEN και για να ελευθερώσουμε τη θέση μνήμης (κλείσουμε το αρχείο) την εντολή CLOSE

```
open([unitnumber],file=[filename],status='old', form='formatted')
```

```
read([unitnumber],'(a160)')xlabelx
```

```
close([unitnumber])
```

Αμα κλείσουμε ένα αρχείο ελευθερώνουμε τη θέση μνήμης του ΗΥ στην οποία μπορούμε να ανοίξουμε ένα άλλο αρχείο για γράψιμο ή διάβασμα δεδομένων

ΠΡΟΣΟΧΗ - 1 → Το **όνομα του αρχείου** [filename] επειδή είναι αυτό που είχε το αρχείο στο λειτουργικό linux πρέπει να γράφεται ακριβώς όπως το βλέπουμε όταν κάνουμε

ls -la

στο σύστημα linux (άρα με κεφαλαίους και μικρούς χαρακτήρες όπου αυτοί εμφανίζονται).

ΠΡΟΣΟΧΗ -2 → Δεν πρέπει να έχουμε ταυτόχρονα ανοικτά δύο αρχεία στην ίδια θέση μνήμης!

Άλλες εντολές για κατανόηση και πιθανή χρήση

INCLUDE [filename]

Για να συμπεριλάβουμε στο πρόγραμμα κομμάτια κώδικα που είναι σε άλλο αρχείο, το [filename].

USE [modulename]

Το **module** / άρθρωμα περιέχεται σε αρχείο filename.f90 και είναι της μορφής

MODULE [modulename]

Δηλώσεις μεταβλητών και παραμέτρων

CONTAINS

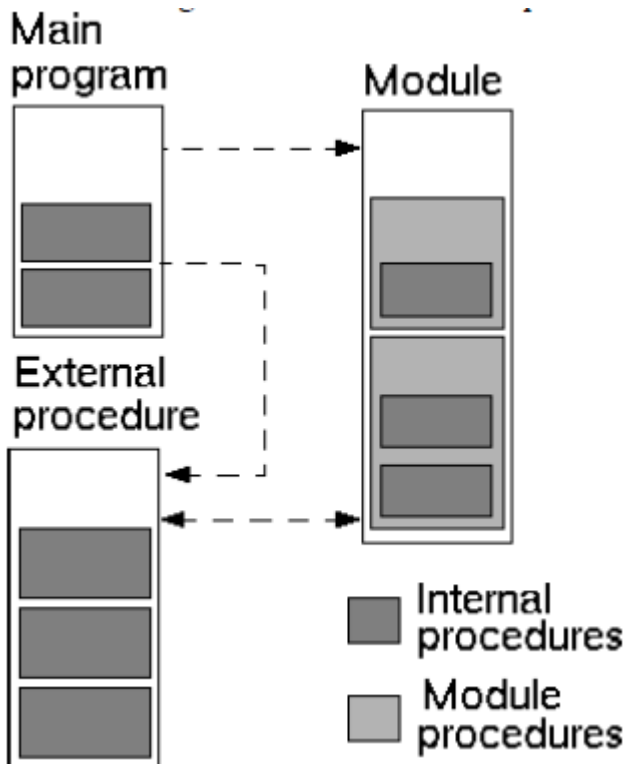
Διαδικασίες αρθρώματος

END MODULE [modulename]

CASE

GO TO

STOP



```

PROGRAM [name]

[specification statements]

[executable statements]

...

[CONTAINS

internal procedures]

END [PROGRAM [name]]

```

Περαιτέρω διάβασμα:

1. Προγραμματισμός FORTRAN 90/95 για επιστήμονες και μηχανικούς – Δ. Σ. Ματαράς & Φ. Α Κουτελιέρης, Εκδόσεις Τζιόλα, Θεσσαλονίκη
2. Σημειώσεις Σταμάτη Σταματιάδη (TETY) Εισαγωγή στη γλώσσα προγραμματισμού Fortran 95
3. Σημειώσεις Introduction to Fortran Nikos Daskalakis
4. Εγχειρίδιο fortran 90 στο διαδίκτυο (introductiontoFortran90.pdf στην Andromeda στο /home/γxp)

καθώς και στο

http://www.pcc.qub.ac.uk/tec/courses/f90/stu-notes/F90_notesMIF_2.html

παραδείγματα μικρών απλών προγραμμάτων

```
PROGRAM nothing
! does nothing
END PROGRAM nothing
```

```
PROGRAM hi
! display a message
WRITE(*,*) 'Hello World!'
END PROGRAM hi
```