

Στοιχεία Επικοινωνίας

Μακρής Χρήστος makri@ceid.upatras.gr

Κουνέλης Φώτιος kounelis@ceid.upatras.gr

Τσίρτση Αργυρώ tsirtsia@ceid.upatras.gr

Μπάρλος Παναγιώτης barlos@ceid.upatras.gr

Δημοπούλου Μερκουρία
dimoroulo@ceid.upatras.gr

Σαραβάνου Χριστίνα saravanou@ceid.upatras.gr

Χατζοπούλου Σοφία chatzopoul@ceid.upatras.gr

Κολωνίας Λεωνίδα kolonias@ceid.upatras.gr

Εισαγωγικό εργαστήριο C

- Γνωριμία με **Linux/Unix OS**
- Γνωριμία με το **Terminal**
- Γνωριμία με το **Diogenis**
- Γνωριμία με βασικές εντολές
- Γνωριμία με **pico**
- Το πρώτο μας πρόγραμμα σε C!
- Γνωριμία με το **gcc**

Για ποιό λόγο Unix OS (1)

- Σταθερά συστήματα
- Ασφαλή
- Εύκολα να τα διαχειριστείς και να τα συντηρήσεις
- Πολύ Καλή Απόδοση σε απαιτητικές εφαρμογές

Για ποιο λόγο Unix OS (2)

Χρησιμοποιούνται σχεδόν παντού:

- Υπολογιστικό κέντρο, web servers
- Routers
- Mac os x
- Linux

UNIX/LINUX systems

Χαρακτηριστικά:

- **Multitasking OS**
- **Multi-user connection**
- **Μικρά προγράμματα που συνεργάζονται άψογα**

Υπολογιστικό κέντρο ΤΜΗΥΠ:

Λειτουργικό σύστημα: CENTOS linux

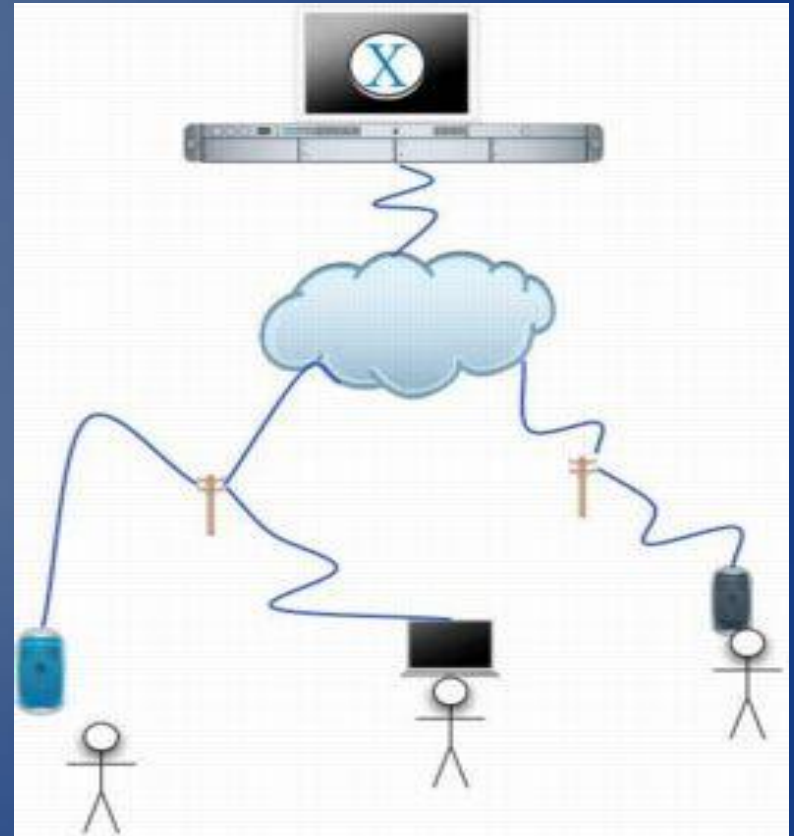
- **Παραπλήσιο με κάθε άλλο UNIX σύστημα!!!!**

Unix OS philosophy

- Ιεραρχικό σύστημα αρχείων
- Πολλοί χρήστες μπορούν να συνδεθούν ταυτόχρονα
- Μπορούν να χρησιμοποιήσουν τους ίδιους υπολογιστικούς πόρους με διαφάνεια
- Μπορούν να σχηματίσουν groups, όπου μπορούν να διαμοιράζουν αρχεία
- Τα πρώτα unix συστήματα δεν είχαν γραφικό παραθυρικό περιβάλλον (gui) – Χρήση μόνο terminal

Multiuser Os

- Μοντέλο, που έλυσε πάρα πολλές δυσκολίες.
- Εξαιρετική χρήση σε όλα τα σύγχρονα λειτουργικά συστήματα ακόμη και σήμερα.



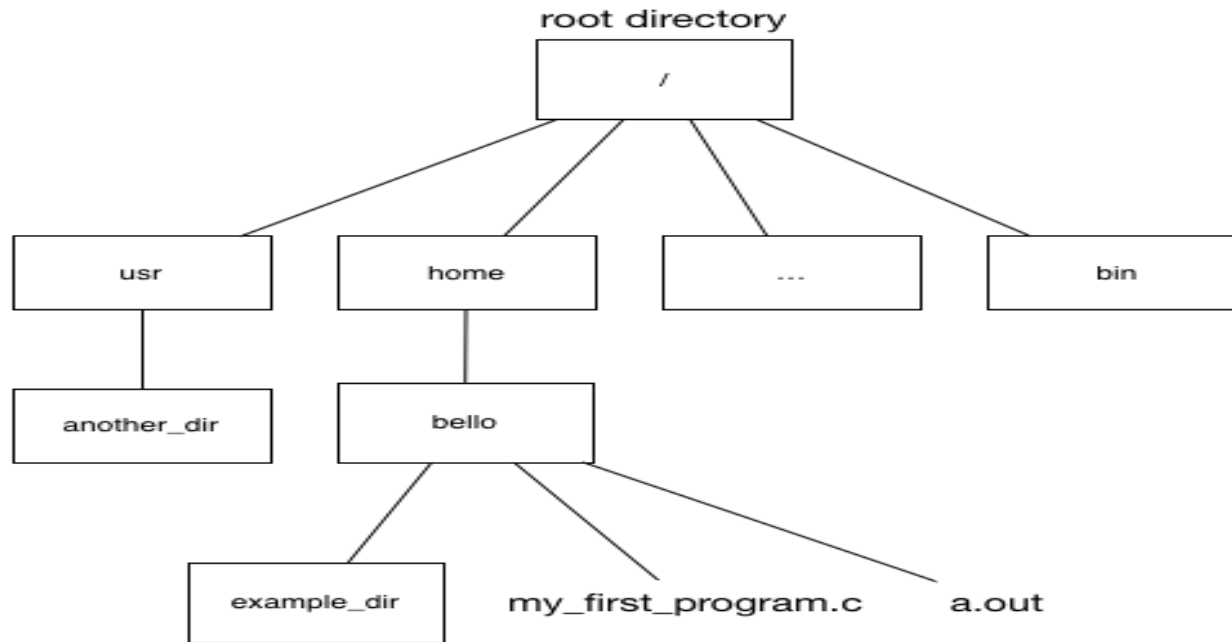
Ιεραρχικό σύστημα αρχείων (1)

- Το file system έχει δενδρική μορφή.
- Ολοι οι φάκελοι υπάρχουν κάτω απο τον αρχικό φάκελο ρίζα (root). Συμβολίζεται με /
- Κάθε υποφάκελος μπορεί να έχει δικούς του υποφακέλους.
- Τα αρχεία αποτελούν φύλλα του δέντρου, που σχηματίζεται.

Ιεραρχικό σύστημα αρχείων (2)

- Κάθε αρχείο και φάκελος έχει μια μοναδική διεύθυνση στο χώρο των αρχείων.
- Το `absolute path` σχηματίζεται από το μονοπάτι από τη ρίζα του δένρου μέχρι το φάκελο στον οποίο βρίσκεται το αρχείο μαζί με το όνομα του αρχείου.
- π.χ `/home/bello/test1.c` (`absolute path`)

Ιεραρχικό σύστημα αρχείων (3)



`/home/bello/my_first_program.c`

Absolute file path

Ιεραρχικό σύστημα αρχείων (4)

- Δεν είναι πάντα απαραίτητο να θυμόμαστε το `absolute path` του αρχείου
- Αν βρισκόμαστε σε κάποιο `directory` τότε όλα τα αρχεία, που βρίσκονται στο φάκελο αυτό μπορούν να αναφερθούν με το σχετικό τους όνομα
- Δηλαδή, αν βρίσκομαι ήδη στον φάκελο `/home` μπορώ να αναφερθώ στο `test1.c` με το όνομα του και μόνο.
- Ωστόσο όπου και αν βρίσκομαι, αν γνωρίζω το `absolute file path` ενός αρχείου μπορώ να αναφερθώ σε αυτό και να το “επεξεργαστώ”.

Δικαίωμα αρχείων-καταλογων

Σε κάθε unix συστημα υπάρχουν δικαιώματα για διάφορες κατηγορίες χρηστών.

Στο diogeni, οι περισσότεροι από εμάς είμαστε απλοί χρήστες και είμαστε ικανοί να τροποποιούμε αρχεία και καταλόγους, που εμείς έχουμε δημιουργήσει.

Οι διαχειριστές έχουν αυξημένες δυνατότητες μεταξύ των οποίων η διαχείριση λογαριασμών.

Γνωριμία με το diogeni



Γνωριμία με τον diogeni

- Ο diogenis είναι ο κεντρικός unix server του τμήματος μας (υπάρχει και ο zenon).
- Με τη χρήση του diogeni μπορούμε να κάνουμε πολλές εργασίες ως φοιτητές στο τμήμα μας.
- Είναι unix σύστημα, άρα ισχύουν όλα όσα αναφέραμε παραπάνω για τις βασικές αρχές λειτουργίας.
- Δυνατότητα απομακρυσμένης σύνδεσης μέσω ssh και μέσω του υπολογιστικού κέντρου.

Γνωριμία με το υπολογιστικό κέντρο

- Για να εισαχθούμε στο σύστημα CENTOS Linux πατάμε πάνω στο CENTOS Linux.
- Έπειτα, εισάγουμε τα Username και Password, που μας έδωσαν από τη γραμματεία:
 - Username: (π.χ. “kounelis”)
 - Πατάμε Enter και πληκτρολογούμε:
 - Password: (π.χ. “pass”)

Συνδεση με το diogeni

- Το επόμενο βήμα είναι η είσοδος μας στο απομακρυσμένο τερματικό του συστήματος.
- Ανοίγουμε το Terminal, με σκοπό να μπορέσουμε να δοκιμάσουμε βασικές εντολές και να γράψουμε το πρώτο C πρόγραμμα, ως εξής:
- Applications->Favorites-> Terminal
- Θα χρησιμοποιήσουμε το ssh για να συνδεθούμε στο diogeni

Χρήση ssh

- Πληκτρολογούμε
 - `ssh username@diogenis.ceid.upatras.gr`
- Έπειτα, μας ζητά το password μας και τέλος, μας ζητά επιβεβαίωση ότι θέλουμε να συνδεθούμε πληκτρολογούμε:
- `yes/no` αν θέλουμε/δε θέλουμε να συνδεθούμε .
- Αν πατήσουμε `yes`, θα είμαστε πλέον συνδεδεμένοι στον `diogeni`.
- Όταν συνδεόμαστε απο το υπολογιστικό κέντρο δεν θα μας ζητηθεί ο κωδικός.

Παράδειγμα χρήσης ssh

```
kounelis@diogenis:~  
File Edit View Search Terminal Help  
[kounelis@ad-fe ~]$ ssh kounelis@diogenis.ceid.upatras.gr  
Last login: Thu Oct 27 14:02:38 2016 from 10.1.0.175  
Welcome to DIOPENIS.ceid.upatras.gr  
[kounelis@diogenis ~]$
```

server name

home directory (~)

username

Απομακρυσμένη είσοδος

Welcome to DIOGENIS.ceid.upatras.gr

Αν βλέπουμε αυτό το μήνυμα είμαστε συνδεδεμένοι στο Diogenis και έτοιμοι να γράψουμε βασικές εντολές!

- Οι εντολές είναι μικρά προγραμματάκια
- Η γενική μορφή μιας εντολής
- `command_name -some_flags argument1 ..argumentN`
- Το όνομα κάθε εντολής προέρχεται συνήθως από την λειτουργία, που υλοποιεί.
- Πχ `cd` -> **change directory**
- Αν εκτελεστεί σωστά μια εντολή συνήθως δεν εμφανίζεται κάποιο μήνυμα επιτυχίας. Αν δεν εκτελεστεί τότε θα εμφανιστεί διαγνωστικό μήνυμα

Βασικές εντολές (1)

Ας δούμε μερικές εντολές:

- `mkdir tests` Δημιουργούμε έναν υποκατάλογο με το όνομα `tests`
- `cd tests` Αλλάζουμε υποκατάλογο και μπαίνουμε στον υποκατάλογο `tests`
- `cd ..` Γυρνάμε στον αρχικό υποκατάλογο(1 βήμα πίσω)
- `pwd` Εμφανίζεται ο υποκατάλογος στον οποίο εργαζόμαστε
- `rmdir tests` Οριστική διαγραφή του υποκαταλόγου με το όνομα `tests` αν είναι κενός!
- `ls/ ls-l` Περιεχόμενα υποκαταλόγου/Περιεχόμενα καταλόγου σε αναλυτική λίστα

Βασικές εντολές (2)

- `mv test.c tests` Μετακινούμε το αρχείο με το όνομα `test.c` στο φάκελο με το όνομα `tests`
- `cp test.c tests/test2.c` Αντιγράφει το αρχείο με το όνομα `test.c` στο φάκελο με το όνομα `tests/` αρχείο `test2.c`
- `rm test2.c` Διαγράφουμε το αρχείο με το όνομα `test2.c`

Σε πολλά unix συστήματα δέν μας ζητείται επιβεβαίωση όταν διαγράφουμε κάτι. Απαιτείται ιδιαίτερη προσοχή.

Αν θέλουμε να μας ζητάει πάντα επιβεβαίωση εκτελούμε

- `rm -i somefile.c`

Βοηθητικές εντολές

- quota Παρατηρούμε πόσα από τα 200.000 διαθέσιμα Kbytes χρησιμοποιούμε π.χ:137072/ 2000000
- man cdΣυνδυάζοντας το man με μία εντολή παίρνουμε πληροφορίες για την εντολή
- chmod 711 afile Αλλάζουμε τα δικαιώματα χρήσης του αρχείου afile
- date Βλέπουμε την τρέχουσα ημερομηνία
- cal 11 2015 Θα δούμε το ημερολόγιο Νοεμβρίου 2015!

Εκτέλεση εντολών (1)

mkdir tests

ls

cd tests

cd tests

cd ..

ls

pwd

rmdir tests

ls-l

Εκτέλεση εντολών (2)

```
mkdir test2
```

```
pico test1.c (Ctrl+O Save, Ctrl+X Exit)
```

```
pico test2.c
```

```
mv test1.c test2
```

```
cd test2
```

```
ls
```

```
mv test1.c myfile.c
```

```
cd ..
```

```
cp test2.c test2
```


Εκτέλεση εντολών (3)

```
cd test2
```

```
ls
```

```
cp test2.c myfile2.c
```

```
ls
```

```
cd ..
```

```
rmdir test2
```

```
cd test2
```

```
pico afile
```

```
rm myfile2.c
```

```
ls
```

Εκτέλεση εντολών (4)

```
rm *.c
```

```
ls -l
```

```
chmod 711 afile
```

```
rm afile
```

```
cd ..
```

```
rmdir test2
```

```
less afile (read only)
```

Filezilla

Για τη μεταφορά αρχείων από τον diogenis στο δικό μας προσωπικό υπολογιστή.

Εγκαθιστούμε το filezilla client

Για να το χρησιμοποιήσουμε το filezilla:

host: diogenis.ceid.upatras.gr

username: "kounelis"

password: "pass"

port:22

Tutorial: https://wiki.filezilla-project.org/FileZilla_Client_Tutorial_%28en%29

Εκτύπωση

Για να εκτυπώσουμε στο σύστημα πατάμε **File-> Print** και στέλνουμε την εργασία στο **CC-All-Printers CC**.

Για να δούμε την ουρά εκτύπωσης γράφουμε την εντολή: **lpq**

Για να δούμε πόσες διαθέσιμες σελίδες έχουμε να τυπώσουμε χρησιμοποιούμε την εντολή: **pages**

PICO (1)

Ανοίγουμε τον pico editor γράφοντας στο terminal `pico bfile.c`

- CTRL+O **Save**
- CTRL+X **Exit**
- CTRL+J **Justify**
- CTRL+R **Read**
- CTRL+W **Search**

PICO (2)

- CTRL+K **Cut text**
- CTRL+U **Uncut text**
- CTRL+C **Cursor text**
- CTRL+T **To spell**

Editors

Υπάρχει δυνατότητα χρήσης κι άλλων text editors, όπως:

- Vi
- Vim
- Mined

Συστήνεται να χρησιμοποιείται ο pico ως πιο φιλικός προς το χρήστη.

Test.c

Θα δοκιμάσουμε το test.c

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

```
int main(void){
```

```
printf("Hello World! This is our first C program!  
  \n");
```

```
return 0;
```

```
}
```


GCC

Για τη μεταγλώττιση του C κώδικά μας χρησιμοποιούμε τον μεταγλωττιστή (compiler):

```
gcc -o output filename.c
```

Αν θέλουμε να ξέρουμε περισσότερα για τα λάθη μας κατά τη μεταγλώττιση χρησιμοποιούμε:

```
gcc -Wall -o test test.c
```

Παράδειγμα GCC

Μεταγλωττίστε:

```
gcc -o test test.c
```

Τρέξτε το πρόγραμμά σας:

```
./test
```

Λάθος- Test.c

Θα δοκιμάσουμε το test.c

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

```
int main(void){
```

```
printf("Hello World! This is our first C program!  
  \n")
```

```
}
```

Τι εκτυπώνεται όταν κάνουμε ΤΩΡΑ gcc;

Τι μάθαμε σήμερα

- Μάθαμε για την φιλοσοφία των unix συστημάτων και των τρόπων, που λειτουργούν.
- Γνωρίσαμε βασικές εντολές που μπορούμε να εκτελέσουμε σε συστήματα unix.
- Μάθαμε πώς αποθηκεύονται τα αρχεία σε ένα unix σύστημα.
- Περιηγηθήκαμε στο file system με χρήση βασικών εντολών.
- Δημιουργήσαμε φακέλους και αρχεία.

Τι μάθαμε σήμερα(2)

- Δημιουργήσαμε ένα απλό πρόγραμμα σε c και το εκτελέσαμε.
- Μάθαμε πώς να συνδεόμαστε στον diogenis και πώς να μεταφέρουμε αρχεία από τον προσωπικό μας υπολογιστή σε αυτό.

Χρήση διογένη για τις ασκήσεις

- Για τις ασκήσεις μάς έχουμε τις εξής απλές επιλογές:
 - 1) Να συνδεθούμε στο diogeni, να γράψουμε το πρόγραμμα μας με χρήση κάποιου editor (pico) να το κάνουμε compile και να το εκτελέσουμε.
 - 2) Να γράψουμε το πρόγραμμα μας στον προσωπικό μας υπολογιστή στο σπίτι, να το μεταφέρουμε μέσω filezilla (winscp, sftp) στον διογένη και να τα εκτελέσουμε όπως πριν.
- Προσοχή μεταφέρουμε τα αρχεία .c και όχι τα executables. Και κάνουμε compile στον diogeni

Προγράμματα (ενδεικτικά)

- Προγράμματα για απομακρυσμένη σύνδεση στον διογένη:

Windows

putty

Os X

terminal(default) via ssh
lterm2 via ssh

Linux

Terminal(default) via ssh

Προγράμματα (ενδεικτικά)

- Επεξεργαστές κειμένου:

Windows

notepad

notepad++

Sublime

Os X

Textmate (gui)

Macvim (gui)

Xcode (gui , ide)

Vim, vi (terminal)

Linux

Gedit (gui)

Vim, vi, pico (terminal)

Sublime

Προγράμματα (ενδεικτικά)

- Προγράμματα μεταφοράς αρχείων από και προς τον διογένη:

Windows

Winscp
filezilla

Os X

Filezilla (gui)
Transmit (gui)
sftp, ftp (terminal)

Linux

Filezilla
sftp , ftp

2ο Εργαστήριο

Εισαγωγή στην C

Δομή ενός προγράμματος C(1)

```
Αρχεία header που θέλουμε να συμπεριλάβουμε  
τύπος-επιστρεφόμενης-τιμής όνομα-συνάρτησης  
(λίστα-παραμέτρων){  
    Κορμός-συνάρτησης(εντολές)  
}
```

Δομή ενός προγράμματος C(2)

```
#include<stdio.h>
int main(void){
    printf("Hello World!");

    return 0;
}
```

Δομή ενός προγράμματος C(3)

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



Header αρχείο

```
int main(void){
```

```
    printf("Hello World!");
```

```
    return 0;
```

```
}
```

Δομή ενός προγράμματος C(4)

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



Header αρχείο

```
int main(void){
```



Η συνάρτηση main()

```
    printf("Hello World!");
```

```
    return 0;
```

```
}
```

Δομή ενός προγράμματος C(5)

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



Header αρχείο

```
int main(void){
```



Η συνάρτηση main()

```
    printf("Hello World!");
```



Το πρόγραμμα μας

```
    return 0;
```

```
}
```

To Include

`#include<stdio.h>`

- Το αρχείο Header είναι ένα αρχείο με συγκεκριμένες δηλώσεις μεταβλητών και συναρτήσεων.
- Η ντιρεκτίβα **#include** λέει στον προεπεξεργαστή να διαβάσει ένα άλλο αρχείο (στην προκειμένη περίπτωση το `stdio.h`) και το συμπεριλάβει στο πρόγραμμα (**include**)

Η συνάρτηση main()

```
int main(void)
```

- Η συνάρτηση main είναι η βασική συνάρτηση του προγράμματος σας και η πρώτη που θα “τρέξει”
- Αρχικά το int προσδιορίζει τον τύπο μεταβλητής που θα επιστέψει η συνάρτηση (main)
- Μέσα στην παρένθεση της main προσδιορίζουμε τον τύπο των ορισμάτων.

Στην περίπτωση μας δεν έχουμε ορίσματα γι αυτό και έχουμε βάλει void.

Το πρόγραμμα μας

- Μέσα στην `main` γράφουμε το πρόγραμμα που θέλουμε.
- Ένα πρόγραμμα αποτελείτε από διάφορες εντολές, όπως `printf("...")`, `scanf("...")`, αναθέσεις και δηλώσεις μεταβλητών,...

Δηλώσεις και τύποι μεταβλητών

τύπος όνομα-μεταβλητής;

π.χ. `int number;`

Τύποι μεταβλητών

- `int` – ακέραιος προσημασμένος αριθμός
- `char` – χαρακτήρας
- `float` – αριθμός κινητής υποδιαστολής
- `double` – αριθμός κινητής υποδιαστολής, διπλής ακρίβειας
- `void` – απουσία τιμής

Ανάθεση μεταβλητών

Όνομα-μεταβλητής = τιμή;

π.χ. `number = 10;`

- Ο τύπος της τιμής πρέπει να ταιριάζει με τον τύπο της μεταβλητής!!!

Είσοδος και έξοδος

Έξοδος στην οθόνη

- `printf("αλφαριθμητικό");`
- `printf("αλφαριθμητικό %τύπος-μεταβλητής", μεταβλητή);`

Είσοδος από το πληκτρολόγιο

- `scanf("%τύπος-μεταβλητής", &μεταβλητή);`

Το όρισμα %τύπος-μεταβλητής

<code>int</code>	<code>%d</code>
<code>char</code>	<code>%c</code>
<code>float</code>	<code>%f</code>
<code>double</code>	<code>%lf</code>

Εκτέλεση αριθμητικών υπολογισμών

<u>Τελεστής</u>	<u>Σημασία</u>
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
%	Υπολοιπο διαίρεσης ακεραίων

example1.c

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

```
int main (void){
```

```
    printf ("Hello World !!! \n");
```

```
    printf ("This is my first program using C language \n");
```

```
    return 0;
```

```
}
```

example2.c

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

```
int main (void){
```

```
    int number;
```

```
    number = 100;
```

```
    printf ("The value is %d ", number);
```

```
    return 0;
```

```
}
```


example3.c

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

```
int main (void) {
```

```
    //use comments!!
```

```
    //use the "tab" key!!
```

```
    char ch;    //we use %c to print variables of type char
```

```
    int i;      //we use %d to print variables of type int
```

```
    float f;    //we use %f to print variables of type float
```

```
    ch = 'X';
```

```
    i = 10;
```

```
    f = 20.1;
```

```
    printf (" ch variable is %c \n", ch );
```

```
    printf (" i variable is %d \n", i );
```

```
    printf (" f variable is %f \n", f );
```

```
    return 0;
```

```
}
```

example4.c

```
#include <stdio.h>

int main (void) {

    //use comments!!

    //use the "tab" key!!

    int number;          //we use %d to print variables of type int
    float f;             //we use %f to print variables of type float

    printf ( " Enter an integer: \n " );
    scanf ( " %d ", &number );    //assign value of type int from keyboard
    printf ( " Enter a floating point number: \n " );
    scanf ( " %f ", &f );        //assign value of type float from keyboard
    printf ( " First value you typed: %d \n", number );
    printf ( " Second value you typed: %f \n", f );
    return 0;
}
```

example5.c

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

```
#include <math.h> //Don't forget that !!!
```

```
int main (void) {
```

```
    double riza; //we use %f to print variables of type double
```

```
    riza = sqrt(10.0) ;
```

```
    printf (" H riza tou 10 einai: %f \n", riza );
```

```
    return 0;
```

```
}
```

example6.c

```
#include <stdio.h>
#include <math.h> //Actually we don't need that now
int main (void) {
    //In this program we compute the Area of a box
    int a, b; //you can define variables of the same type in one line
    printf ( " Enter length of box: \n" );
    scanf( "%d", &a );
    printf ( " Enter width of box: \n" );
    scanf( "%d", &b );
    // Compute AND Print the Area
    printf ( " Area is: %d \n", a * b ); //you can compute inside of printf

    return 0;
}
```