

lab0

Σκοπός του εισαγωγικού εργαστηρίου είναι να εξοικειωθείτε με το περιβάλλον συγγραφής και εκτέλεσης προγραμμάτων C που θα χρησιμοποιείτε στο εργαστήριο καθώς και με το περιβάλλον υποβολής και αυτόματης βαθμολόγησης εργασιών. Στο τέλος του εργαστηρίου θα πρέπει να έχετε ολοκληρώσει τα εξής:

- Εξάσκηση σε βασικές εντολές του περιβάλλοντος γραμμής εντολών του Linux.
- Συγγραφή ενός απλού προγράμματος C
- Επιτυχής μεταγλώττιση ενός προγράμματος C, χωρίς warnings
- Επιτυχής εκτέλεση ενός προγράμματος
- Σύνδεση στο περιβάλλον autolab κι επιτυχής υποβολή του κώδικα σε αυτό

Πριν έρθετε στο εισαγωγικό εργαστήριο:

- Πρέπει να έχετε μελετήσει και καταλάβει το περιεχόμενο της ενότητας "Εισαγωγικές έννοιες".
- Πρέπει να έχετε διαβάσει τουλάχιστον μία φορά την ενότητα "Εξάσκηση στο εργαστήριο".
- Συνίσταται να έχετε ολοκληρώσει την εγγραφή σας στο autolab.

Κατά τη διάρκεια του εισαγωγικού εργαστηρίου:

- Διαβάστε την εκφώνηση και ολοκληρώστε τις ασκήσεις που εμφανίζονται με **σιέλ φόντο**.

Μετά το εισαγωγικό εργαστήριο:

- Εάν αισθάνεστε πως έχετε ελλείψεις, μπορείτε να πάτε στο εργαστήριο 205 ή στο 206 σε ώρες που δε γίνονται μαθήματα και να κάνετε επιπλέον εξάσκηση. Περιμένουμε ότι από το lab1 και μετά θα γνωρίζετε ότι καλύφθηκε στο lab0.

Εισαγωγικές έννοιες

Το λειτουργικό σύστημα Linux

Το λειτουργικό σύστημα είναι ένα λογισμικό που διαχειρίζεται τους πόρους και συσκευές του υπολογιστή (π.χ. μνήμη, επεξεργαστής, δίσκος κτλ.), υποστηρίζει την εκτέλεση εφαρμογών (π.χ. browser, παιχνίδια) και αποτελεί σύνδεσμο ανάμεσα στο χρήστη και στο σύστημα. Οι περισσότεροι πιθανώς έχετε εξοικείωση με τα λειτουργικά συστήματα Windows, iOS, Android. Σε αυτό το μάθημα θα χρησιμοποιήσετε Linux.

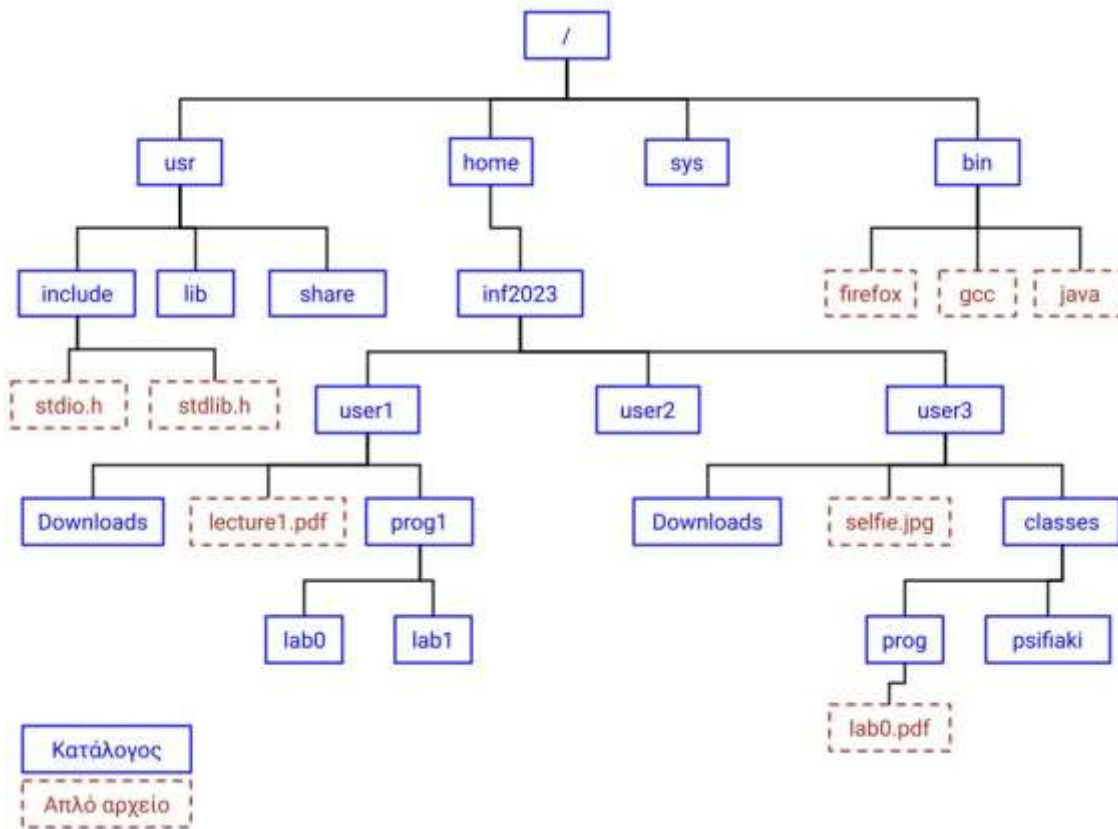
Προς το παρόν μας ενδιαφέρουν δύο πράγματα: (α) ο τρόπος με τον οποίο είναι οργανωμένα τα αρχεία και (β) ο τρόπος με τον οποίο ένας χρήστης επικοινωνεί με το λειτουργικό σύστημα.

Ιεραρχική δομή αρχείων

Το σύστημα αρχείων είναι οργανωμένο ιεραρχικά. Όλα ξεκινούν από ένα **κατάλογο (directory)** που λέγεται root και συμβολίζεται / . Μέσα σε αυτόν τον κατάλογο υπάρχουν διάφορα αρχεία του συστήματος καθώς και υποκατάλογοι. Σε κάθε υποκατάλογο υπάρχουν άλλα αρχεία και υποκατάλογοι, κ.ο.κ.

Δείτε στην [Εικόνα 1](#) ένα μικρό δείγμα της δομής των αρχείων σε ένα σύστημα όπως των εργαστηρίων. Παρατηρήστε τους καταλόγους με τα ενδεικτικά ονόματα user1, user2, user3. Αυτοί είναι οι λεγόμενοι **προσωπικοί κατάλογοι (home directories¹)** των χρηστών που έχουν ως usernames τα user1, user2 και user3 αντίστοιχα. Στον προσωπικό του κατάλογο μπορεί ο κάθε χρήστης να αποθηκεύσει ό,τι αρχεία θέλει και να κατασκευάσει δικούς του υποκαταλόγους. Ένας χρήστης δεν έχει πρόσβαση στους προσωπικούς καταλόγους των άλλων χρηστών.

¹ Μη μπερδεύετε τον όρο "home directory" με το όνομα του καταλόγου "home" που βλέπετε στην [Εικόνα 1](#).



Εικόνα 1: Ιεραρχική δομή αρχείων

Η διαδρομή από έναν αρχικό κατάλογο προς έναν άλλο λέγεται **μονοπάτι (path)** και αποτελείται από τους καταλόγους που πρέπει να διανύσει κανείς για να μεταβεί από τον αρχικό στον τελικό κατάλογο, με το σύμβολο / ανάμεσά τους. Για παράδειγμα, στο σύστημα της εικόνας 1 το μονοπάτι από το root προς τον κατάλογο prog1 του χρήστη user1 είναι /home/inf2023/user1/prog1, ενώ το μονοπάτι από τον κατάλογο inf2023 προς τον κατάλογο psifiaki του user3 είναι inf2023/user3/classes/psifiaki. Θα δούμε αργότερα πιο σύνθετους τρόπους να περιγράψουμε ένα μονοπάτι.

Ερώτηση κατανόησης 1: Γιατί το μονοπάτι του πρώτου παραδείγματος ξεκινά από / ενώ το δεύτερο όχι; (Δείτε την απάντηση στο τέλος)

Επικοινωνία με το λειτουργικό σύστημα

Αν έχετε χρησιμοποιήσει Windows, τότε ο τρόπος με τον οποίο επικοινωνείτε με το λειτουργικό σύστημα είναι μέσα από ένα γραφικό περιβάλλον και με τη βοήθεια συσκευών όπως το ποντίκι που σας επιτρέπουν να αλληλεπιδράσετε με ό,τι βλέπετε στην οθόνη. Αυτό είναι ένα παράδειγμα **γραφικής διεπαφής με το χρήστη (graphical user interface, GUI)**.

Μια κατηγορία γραφικής διεπαφής είναι το περιβάλλον οθόνης αφής που έχετε χρησιμοποιήσει αν έχετε smartphone. Υπάρχουν και άλλου είδους διεπαφές, όπως για παράδειγμα μέσω φωνής (voice user interface).

Το Linux παρέχει γραφική διεπαφή, αλλά και μια **διεπαφή γραμμής εντολών (command-line interface, CLI)**. Η ιδιαιτερότητα της διεπαφής γραμμής εντολών είναι ότι ο χρήστης αντί να χρησιμοποιεί το ποντίκι για να κάνει click σε διάφορα εικονίδια, εφαρμογές κτλ., χρησιμοποιεί το πληκτρολόγιο: ο χρήστης γράφει εντολές σε ένα ειδικό παράθυρο που λέγεται **τερματικό (terminal window)** ή κονσόλα (console) οι οποίες ερμηνεύονται και εκτελούνται από ένα ειδικό πρόγραμμα που λέγεται shell.

Δείτε [εδώ](#) ένα παράδειγμα ανοίγματος της εφαρμογής Visual Studio Code χρησιμοποιώντας αποκλειστικά το γραφικό περιβάλλον (κάνουμε click στο εικονίδιο της εφαρμογής), και [εδώ](#) ένα παράδειγμα ανοίγματος της ίδιας εφαρμογής χρησιμοποιώντας τη γραμμή εντολών (ανοίγουμε πρώτα ένα τερματικό παράθυρο και μετά γράφουμε το όνομα της εφαρμογής).

Όπως ίσως μαντέψατε, για την επικοινωνία μέσω του CLI ο χρήστης πρέπει να γνωρίζει ακριβώς ποιες εντολές είναι διαθέσιμες και πώς χρησιμοποιούνται. Στα εργαστήρια του προγραμματισμού θα χρησιμοποιήσετε ένα πολύ μικρό υποσύνολο αυτών των εντολών για να μεταβαίνετε σε καταλόγους, να δημιουργείτε νέους, να μεταγλωττίζετε κώδικα και να εκτελείτε τα προγράμματά σας καθώς και κάποια βοηθητικά εργαλεία.

Η επόμενη ενότητα θα γίνει στο εργαστήριο, αλλά θα θέλαμε να την έχετε διαβάσει τουλάχιστον μια φορά πριν έρθετε.

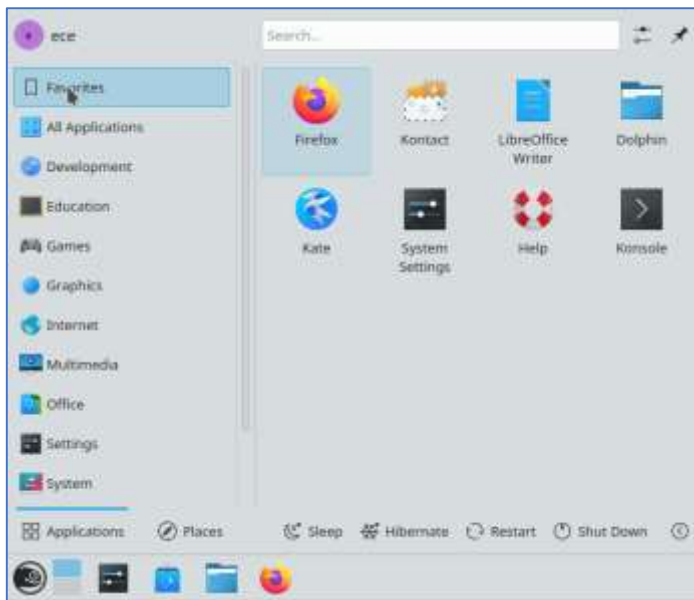
Εξάσκηση στο εργαστήριο

Αρχική σύνδεση

Όταν ανοίξετε ένα PC του εργαστηρίου σε Linux και εισάγετε επιτυχώς το username και τον κωδικό σας, θα δείτε στο κάτω αριστερά μέρος της οθόνης ένα εικονίδιο χαμαιλέοντα:



Κάνοντας click σε αυτό εμφανίζεται το βασικό μενού από το οποίο μπορείτε να προσπελάσετε τις διάφορες εγκατεστημένες εφαρμογές. Κάποιες εφαρμογές που χρησιμοποιούνται συχνά εμφανίζονται στην κατηγορία Favorites για εύκολη πρόσβαση:



Σε αυτή την εικόνα βλέπετε το αρχικό μενού με επιλεγμένη την κατηγορία Favorites. Προς το παρόν σας ενδιαφέρουν οι εξής εφαρμογές:

- Firefox: Ο γνωστός browser.
- Dolphin: Εφαρμογή για πλοήγηση στους καταλόγους και στα αρχεία σας.
- Konsole: Εφαρμογή για τερματικό παράθυρο.

Εικόνα 2: Το αρχικό μενού, με επιλεγμένα τα Favorites

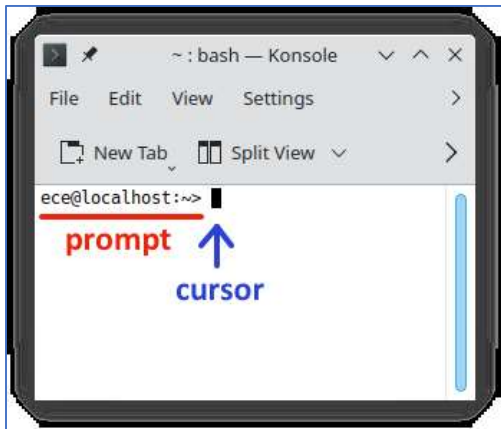
Άσκηση 1: Εντοπίστε την κατηγορία Utilities (είναι πιο κάτω από το System) και μέσα σε αυτή την εφαρμογή Visual Studio Code. Κάντε δεξί click στο Visual Studio Code κι επιλέξτε "Add to Favorites". Αυτό θα έχει ως αποτέλεσμα να συμπεριληφθεί το Visual Studio Code στα Favorites. Μην ανοίξετε ακόμη το Visual Studio Code.

Άσκηση 2: Ανοίξτε το Dolphin για να δείτε τα περιεχόμενα του προσωπικού σας καταλόγου. Παρατηρήστε πώς στο αριστερό panel εμφανίζονται σύνδεσμοι προς τους πιο συνηθισμένους καταλόγους. Για παράδειγμα, ο σύνδεσμος Home σας παραπέμπει πάντα στον προσωπικό σας κατάλογο, ενώ ο σύνδεσμος Desktop στην επιφάνεια εργασίας. Πλοηγηθείτε σε διάφορους καταλόγους, κι επιβεβαιώστε αυτά που διαβάσατε στην προηγούμενη ενότητα αναφορικά με τη δομή των αρχείων και καταλόγων στο Linux.

Βασικές εντολές τερματικού

Πριν ξεκινήσετε να δοκιμάζετε τις βασικές εντολές τερματικού σημειώστε πως το Linux είναι **case sensitive**, δηλαδή ευαίσθητο στα κεφαλαία και μικρά γράμματα. Οι εντολές τερματικού γράφονται πάντα με μικρά. Αν κατά λάθος γράψετε κάποια εντολή με κεφαλαία, δε θα αναγνωριστεί και θα εμφανιστεί το μήνυμα λάθους "Command not found".

Άσκηση 3: Ανοίξτε ένα τερματικό. Θα πρέπει να δείτε κάτι σαν το παρακάτω. Το τερματικό παράθυρο ανοίγει κατά κανόνα στον προσωπικό κατάλογο του χρήστη (όπως και το Dolphin που είδατε νωρίτερα).



Εικόνα 3: Τερματικό παράθυρο.

Ο **cursor** σας δείχνει το σημείο που μπορείτε να γράψετε μια εντολή.

Το **prompt** (προτροπή για εντολή) εμφανίζεται πάντα στην αρχή της γραμμής και αρκεί να είναι ένα απλό σύμβολο όπως \$ ή >. Συχνά η γραμμή περιέχει επιπλέον πληροφορίες όπως το όνομα του χρήστη (ece σε αυτό το παράδειγμα) και το μονοπάτι προς τον κατάλογο που βρίσκεται αυτή τη στιγμή ο χρήστης.

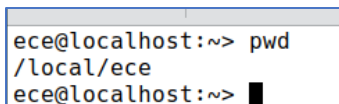
Παρατηρήστε το **ειδικό σύμβολο ~**: Αυτό είναι συντομογραφία για το μονοπάτι από το root προς τον προσωπικό κατάλογο του χρήστη που έχει συνδεθεί στο σύστημα.

Ερώτηση κατανόησης 2: Σε ένα σύστημα όπως αυτό της εικόνας 1, πώς θα χρησιμοποιούσε ο χρήστης user1 το σύμβολο ~ για να αναφερθεί στο μονοπάτι από το root προς τον κατάλογο lab0 του; (Η απάντηση στο τέλος)

Η εντολή pwd

Το όνομα της εντολής pwd προκύπτει από τη φράση **p**rint **w**orking **d**irectory και χρησιμοποιείται για να εμφανίσει το μονοπάτι από το root μέχρι τον κατάλογο στον οποίο βρίσκεται εκείνη τη στιγμή ο χρήστης.

Άσκηση 4: Πληκτρολογήστε pwd στο τερματικό που ανοίξατε προηγουμένως. Θα πρέπει να δείτε κάτι αντίστοιχο του παρακάτω (αν και σε εσάς το μονοπάτι θα είναι μάλλον /home/inf2023/username, όπου username το όνομα χρήστη σας)



Εικόνα 4: Η εντολή pwd

Η εντολή ls

Το όνομα της εντολής ls προκύπτει από τη λέξη **l**ist και χρησιμοποιείται για να εμφανίσει τα περιεχόμενα του καταλόγου στον οποίο βρίσκεται εκείνη τη στιγμή ο χρήστης.

Άσκηση 5: Πληκτρολογήστε ls στο τερματικό που ανοίξατε προηγουμένως. Εφόσον είστε ακόμη μέσα στον προσωπικό σας κατάλογο, θα πρέπει να δείτε κάτι σαν το παρακάτω. Θυμηθείτε πως τα ίδια περιεχόμενα είχατε δει νωρίτερα μέσω Dolphin όταν κάνατε click στο Home.

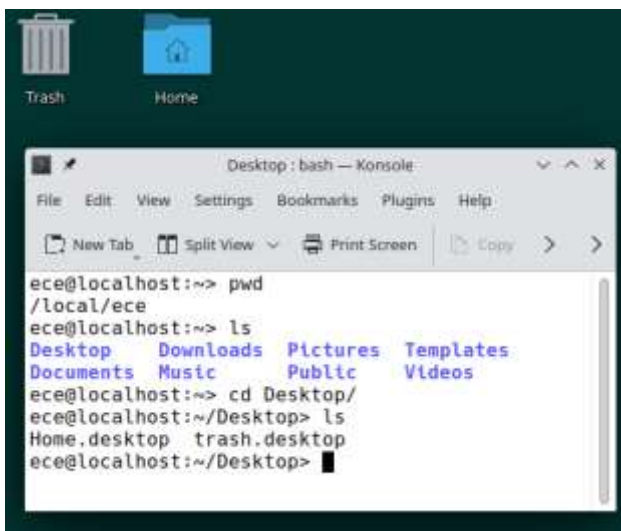
```
ece@localhost:~> pwd
/local/ece
ece@localhost:~> ls
Desktop  Downloads  Pictures  Templates
Documents Music      Public    Videos
ece@localhost:~> █
```

Εικόνα 5: Η εντολή ls

Η εντολή cd

Το όνομα της εντολής cd προκύπτει από τη φράση change directory και χρησιμοποιείται όταν ο χρήστης επιθυμεί να μεταβεί σε διαφορετικό κατάλογο από τον οποίο βρίσκεται. Ο χρήστης πρέπει να προσδιορίσει το νέο μονοπάτι μετά το όνομα της εντολής έχοντας αφήσει ένα ή περισσότερα κενά. Μπορεί να προσδιοριστεί είτε ένα μονοπάτι από το root προς τον προορισμό, είτε ένα μονοπάτι από τον τρέχων κατάλογο έως τον προορισμό.

Στην άσκηση 4 διαπιστώσατε με χρήση ls ότι τα περιεχόμενα του καταλόγου στον οποίο βρίσκεστε περιλαμβάνουν έναν κατάλογο με όνομα Desktop. Δείτε στην εικόνα 6 πώς μπορείτε να χρησιμοποιήσετε την εντολή cd για να μεταβείτε στο Desktop.



Εικόνα 6: Παράδειγμα χρήσης cd

Παρατηρήστε ότι:

(α) Αμέσως μετά τη χρήση της εντολής cd, ανανεώθηκε το τρέχον μονοπάτι που εμφανίζεται στο prompt.

(β) Αν μετά την εντολή cd γράψετε ls, θα δείτε τα περιεχόμενα του καταλόγου Desktop στον οποίο έχετε μεταβεί. Επιβεβαιώστε ότι το αποτέλεσμα της εντολής ls είναι συμβατό με τους συνδέσμους Home και Trash που βρίσκονται στην επιφάνεια εργασίας.

Επίσης, σημειώστε πως αν η εντολή cd χρησιμοποιηθεί χωρίς να προσδιοριστεί κάποιος προορισμός (δηλαδή γράφοντας σκέτο cd και πατώντας enter), τότε γίνεται αυτόματα μετάβαση στον προσωπικό κατάλογο του χρήστη.

Τέλος, το τελικό / στο τέλος του μονοπατιού είναι προαιρετικό. Η εντολή cd Desktop/ είναι ίδια με την εντολή cd Desktop

Υπάρχει ένα ειδικό όνομα καταλόγου το οποίο λέγεται .. (δύο τελείες χωρίς κενό ανάμεσά τους) και αναφέρεται πάντα στον "γονικό" κατάλογο του τρέχοντος. Για παράδειγμα, αν είστε μέσα στο Desktop, τότε το .. αναφέρεται στον προσωπικό σας κατάλογο. Δείτε πώς χρησιμοποιείται το .. στην εικόνα 7.

```
ece@localhost:~> pwd
/local/ece
ece@localhost:~> ls
Desktop  Downloads  Pictures  Templates
Documents Music      Public    Videos
ece@localhost:~> cd Desktop/
ece@localhost:~/Desktop> ls
Home.desktop trash.desktop
ece@localhost:~/Desktop> pwd
/local/ece/Desktop
ece@localhost:~/Desktop> cd ..
ece@localhost:~> pwd
/local/ece
ece@localhost:~> █
```

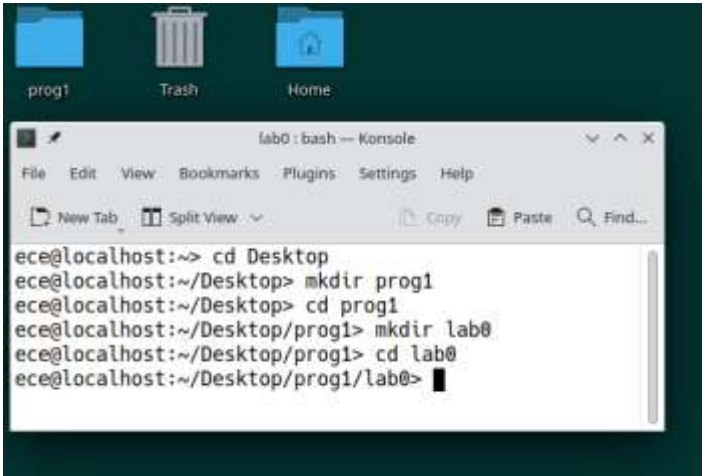

Για ένα πιο σύνθετο παράδειγμα, δείτε και πάλι την [Εικόνα 1](#). Ας υποθέσουμε ότι ο χρήστης user1 βρίσκεται μέσα στον κατάλογο lab1 και θέλει να μεταβεί στον κατάλογο lab0. Θα μπορούσε να το κάνει σε δύο βήματα, πρώτα με cd .. ώστε να μεταβεί στο prog1 και από εκεί με cd lab0 ώστε να μεταβεί στο lab0. Αλλά θα μπορούσε να το κάνει και σε ένα βήμα, γράφοντας cd ../lab0

Ερώτηση κατανόησης 3: Ο χρήστης user1 βρίσκεται στον κατάλογο lab1 και θέλει να μεταβεί στον κατάλογο Downloads. Προτείνετε τουλάχιστον τρεις τρόπους να το κάνει με χρήση της εντολής cd. (Η απάντηση στο τέλος).

Η εντολή mkdir

Το όνομα της εντολής mkdir προκύπτει από τη φράση **make directory** και χρησιμοποιείται όταν ο χρήστης επιθυμεί να κατασκευάσει ένα νέο κατάλογο μέσα σε αυτό στον οποίο βρίσκεται². Ο χρήστης πρέπει να προσδιορίσει το όνομα του νέου καταλόγου μετά το όνομα της εντολής έχοντας αφήσει ένα ή περισσότερα κενά. Συνιστάται να μην υπάρχουν κενά στο όνομα του νέου καταλόγου. Αντί για κενά, συνήθως χρησιμοποιείται κάτω παύλα (underscore).

Στην εικόνα 8 μπορείτε να δείτε πώς χρησιμοποιούμε την εντολή mkdir για να κατασκευάσουμε ένα νέο κατάλογο με όνομα prog1 μέσα στο Desktop και μετά ένα νέο κατάλογο με όνομα lab0 μέσα στο prog1.



```
ece@localhost:~> cd Desktop
ece@localhost:~/Desktop> mkdir prog1
ece@localhost:~/Desktop> cd prog1
ece@localhost:~/Desktop/prog1> mkdir lab0
ece@localhost:~/Desktop/prog1> cd lab0
ece@localhost:~/Desktop/prog1/lab0>
```

Εικόνα 8: Παράδειγμα χρήσης mkdir

Παρατηρήστε πως όταν κατασκευάστηκε ο κατάλογος prog1 μέσα στο Desktop με χρήση της εντολής mkdir, εμφανίστηκε άμεσα στην επιφάνεια εργασίας που βλέπετε στο γραφικό περιβάλλον.

Το lab0 δε φαίνεται, εφόσον είναι μέσα στο prog1, αλλά αν κάνετε click στο εικονίδιο του prog1 τότε θα ανοίξει μέσω Dolphin και θα δείτε μέσα σε αυτό το lab0.

Άσκηση 6: Χρησιμοποιήστε τις εντολές cd και mkdir για να κατασκευάσετε (α) ένα κατάλογο στον οποίο θα αποθηκεύετε το υλικό του Προγραμματισμού 1 και (β) μέσα σε αυτόν έναν ακόμη κατάλογο στον οποίο θα αποθηκεύετε το πρόγραμμά σας για το lab0. Μπορείτε να χρησιμοποιήσετε τα ίδια ονόματα που έχει το παράδειγμα της εικόνας 8 ή δικά σας.

Χρήση VS Code για συγγραφή προγραμμάτων C

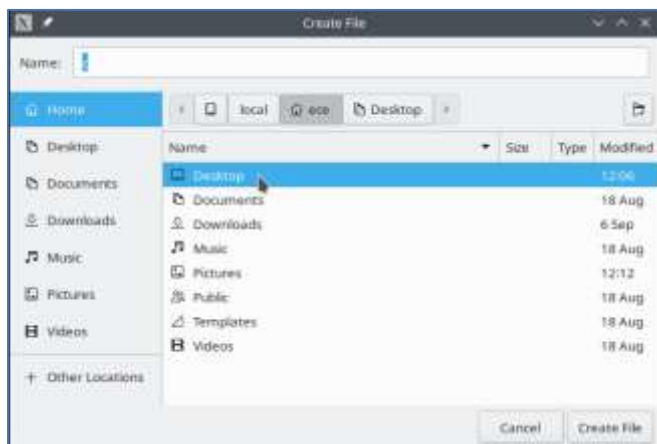
Τώρα είστε σχεδόν έτοιμοι να γράψετε το πρώτο σας πρόγραμμα. Κάθε φορά που έχετε να γράψετε ένα πρόγραμμα C, θα πρέπει να δημιουργείτε πρώτα ένα αρχείο με το κατάλληλο όνομα και να το αποθηκεύετε στον κατάλογο που θέλετε. Είναι σημαντικό να δώσετε στο αρχείο τη σωστή κατάληξη (.c) γιατί αυτό δίνει τη δυνατότητα στο VS Code να χρωματίζει με κατάλληλο τρόπο τις εντολές του προγράμματος και να αυξάνει την αναγνωσιμότητά του.

² Μπορεί να κατασκευαστεί και κατάλογος σε διαφορετικό σημείο. Δε θα καλύψουμε κάτι τέτοιο, αλλά μπορείτε να σκεφτείτε πώς θα γινόταν - έχετε τις απαραίτητες γνώσεις.

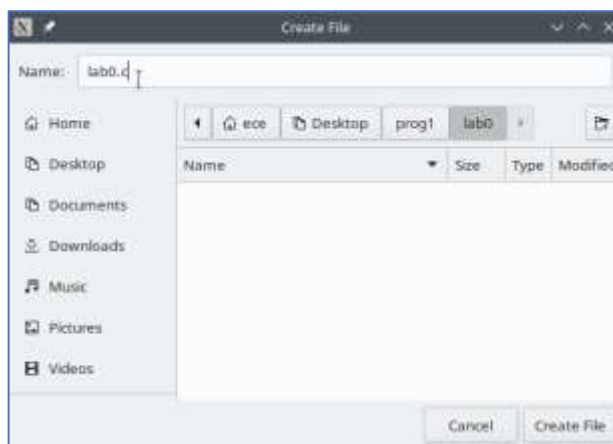
Επίσης, είναι σημαντικό να κάνετε συχνά αποθήκευση (save) του κώδικά σας.

Άσκηση 7:

1. Ανοίξτε το VS Code είτε μέσα από το γραφικό περιβάλλον, είτε γράφοντας την εντολή code στο τερματικό σας.
2. Στο VS Code επιλέξτε από το μενού File και μετά New File. Θα εμφανιστεί ένα πεδίο με την υπόδειξη "Select file type". Γράψτε c και κάντε click στο "New File (c) Built-in" που θα εμφανιστεί.
3. Θα ανοίξει ένα παράθυρο με τίτλο "Create File" και ένα πεδίο "Name:". Εδώ πρέπει να κάνετε τα εξής: Να επιλέξετε πού θέλετε να αποθηκευτεί το αρχείο C που θα φτιάξετε και να προσδιορίσετε το όνομά του:
 - a. Επιλέξτε τον κατάλογο που φτιάξατε νωρίτερα για να αποθηκεύσετε το υλικό του lab0. Αν ακολουθήσατε το παράδειγμα της εικόνας 8, τότε θα κάνετε διπλό click στο Desktop, μετά στο prog1 και μετά στο lab0. Δείτε την Εικόνα 9.
 - b. Στο πεδίο Name: γράψτε το όνομα του αρχείου, lab0.c χωρίς κενά. Δείτε την Εικόνα 10.
4. Κάντε click στο "Create File". Είστε πια στο περιβάλλον του επεξεργαστή κειμένου και μπορείτε να ξεκινήσετε να γράφετε το πρόγραμμά σας.



Εικόνα 9: Επιλογή καταλόγου



Εικόνα 10: Προσδιορισμός ονόματος αρχείου

Κάποιες συμβουλές πριν ξεκινήσετε να γράφετε κώδικα:

- Να σώζετε συχνά το πρόγραμμά σας.
- Να σώζετε πάντα πριν κάνετε compile.
- Να γράφετε σωστά από την αρχή (μη λέτε ποτέ "θα το διορθώσω αργότερα")
- Να είστε πάντα προσεκτικοί και ακριβείς όσον αφορά στα ζητούμενα της άσκησης.

Άσκηση 8: Γράψτε το παρακάτω πρόγραμμα στο VS Code:



```
lab0.c
local > ece > Desktop > prog1 > lab0 > C lab0.c > ...
1  /* Read a person's age and print a message about it */
2
3  #include<stdio.h>
4
5  int main (int argc, char *argv[]) {
6      int age;
7
8      printf("How old are you?\n");
9      scanf("%d", &age);
10
11     printf("You are %d years old!\n", age);
12
13     return 0;
14 }
15
Ln 17, Col 1 Spaces: 4 UTF-8 LF C Linux
```

Εικόνα 11: Το πρώτο σας πρόγραμμα

Προσοχή: Να αποθηκεύετε συχνά το αρχείο σας (μενού File->Save ή ctrl-s). Παρατηρήστε πως αν έχουν γίνει αλλαγές που δεν έχουν αποθηκευτεί, τότε εμφανίζεται μία κουκίδα στο tab με το όνομα του αρχείου (δείτε εικόνα 12). Αν οι αλλαγές έχουν αποθηκευτεί, τότε αντί για κουκίδα εμφανίζεται ένα X.

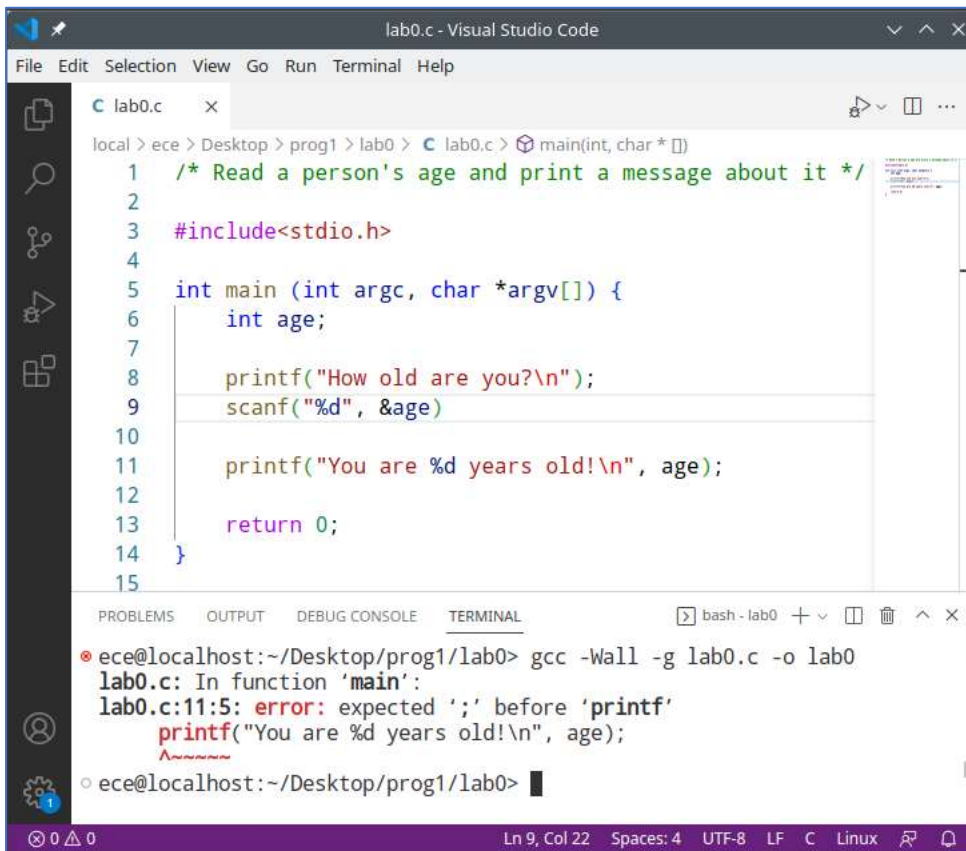
Μεταγλώττιση κι εκτέλεση προγράμματος

Για τη μεταγλώττιση κι εκτέλεση του προγράμματός σας, θα χρειαστεί να χρησιμοποιήσετε τερματικό. Μπορείτε να ανοίξετε ένα τερματικό ανεξάρτητα από το VS Code, ή να χρησιμοποιήσετε αυτό που είναι ενσωματωμένο στο VS Code κάνοντας click στο μενού Terminal κι επιλέγοντας New Terminal. Στα παραδείγματα χρησιμοποιούμε το ενσωματωμένο τερματικό.

Άσκηση 9:

1. Ανοίξετε ένα τερματικό και χρησιμοποιήστε την εντολή `cd` (και ίσως `ls` για επιβεβαίωση) για να μεταβείτε στον κατάλογο που αποθηκεύσατε το `lab0.c`. Δείτε την εικόνα 13 για ένα παράδειγμα.
2. **Αποθηκεύστε (save) το αρχείο!**
3. Χρησιμοποιήστε την εντολή `gcc` όπως έχετε δει στο μάθημα για να μεταγλωττίσετε το πρόγραμμά σας. Θυμίζουμε είναι `gcc -Wall -g lab0.c -o lab0`
4. Αν σας εμφανιστεί μήνυμα λάθους (error) ή προειδοποίηση (warning), δείτε κάποιες συμβουλές στην ενότητα "Εντοπισμός και διόρθωση λαθών" που ακολουθεί κι αν χρειαστεί ζητήστε βοήθεια από το προσωπικό του εργαστηρίου.
5. Αν δεν υπάρχει λάθος, θα δημιουργηθεί ένα εκτελέσιμο με όνομα `lab0`. Για σιγουριά, χρησιμοποιήστε την εντολή `ls` στον κατάλογο που βρίσκεται το `lab0.c` για να επιβεβαιώσετε ότι όντως υπάρχει το αρχείο `lab0`.
6. Στον ίδιο κατάλογο, εκτελέστε το πρόγραμμά σας μέσω τερματικού γράφοντας `./lab0`

Παράδειγμα 1



The screenshot shows the Visual Studio Code editor with a C program named lab0.c. The code is as follows:

```
1  /* Read a person's age and print a message about it */
2
3  #include<stdio.h>
4
5  int main (int argc, char *argv[]) {
6      int age;
7
8      printf("How old are you?\n");
9      scanf("%d", &age)
10
11     printf("You are %d years old!\n", age);
12
13     return 0;
14 }
15
```

The terminal output shows the compilation command and an error message:

```
ece@localhost:~/Desktop/prog1/lab0> gcc -Wall -g lab0.c -o lab0
lab0.c: In function 'main':
lab0.c:11:5: error: expected ';' before 'printf'
printf("You are %d years old!\n", age);
^~~~~~
ece@localhost:~/Desktop/prog1/lab0>
```

Εικόνα 13: Παράδειγμα λάθους μεταγλώττισης

Διαβάστε προσεκτικά το μήνυμα λάθους. Σας λέει: Στο αρχείο lab0.c, στη συνάρτηση main, στη γραμμή 11 και στήλη 5, εμφανίστηκε το εξής σφάλμα: "περίμενα να δω ένα ; πριν την printf". Ακολουθώντας, σας δείχνει με ένα ^ ακριβώς το σημείο στο οποίο αναφέρεται.

Αν κοιτάξετε τη γραμμή 11, θα δείτε πράγματι μια εντολή printf. Αλλά το μήνυμα λέει ότι το ; το περίμενε πριν την printf. Επομένως, πρέπει να κοιτάξετε τι γίνεται στην προηγούμενη εντολή, δηλαδή στο τέλος της γραμμής 9.

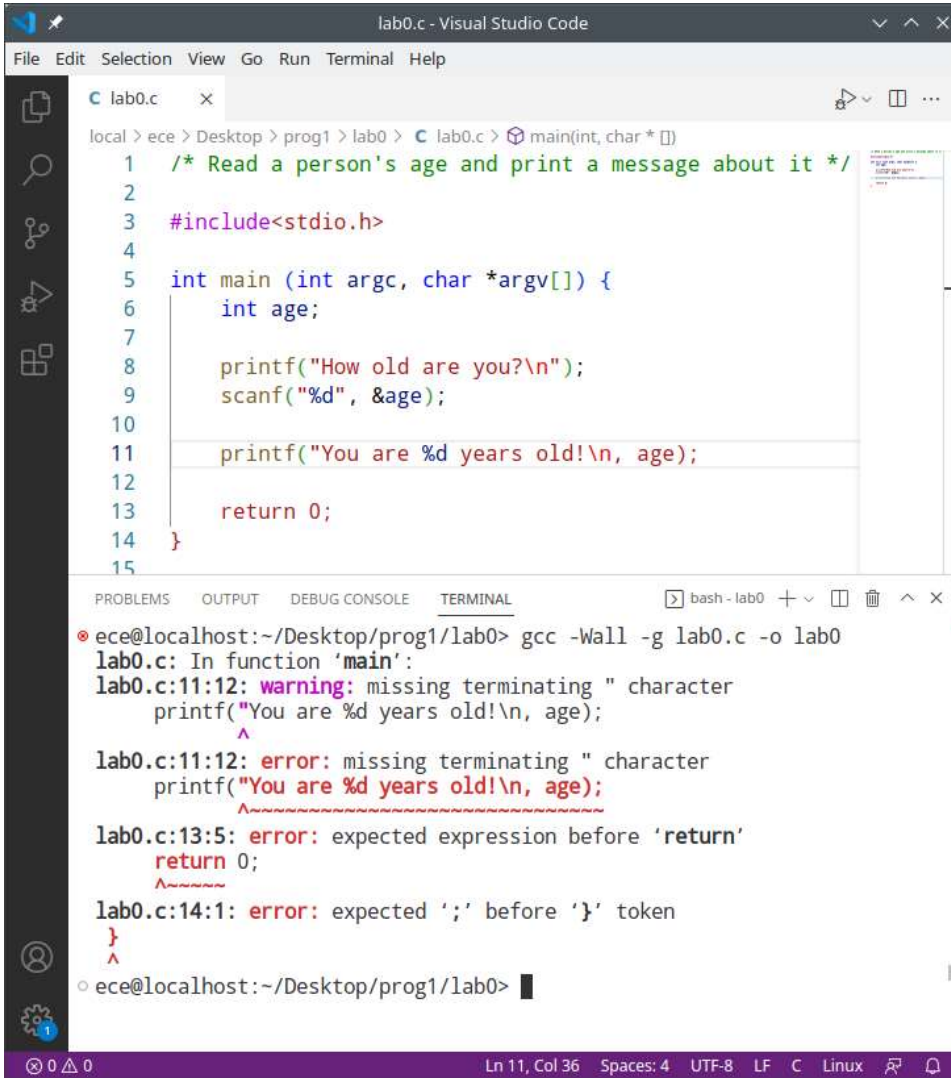
Πράγματι, εκεί λείπει ένα ; το οποίο θα έπρεπε να βρίσκεται μετά τη δεξιά παρένθεση.

Μετά από κάθε ΜΙΑ διόρθωση σε ένα πρόγραμμα πρέπει ΠΑΝΤΑ:

1. Να το αποθηκεύετε (save)
2. Να κάνετε ξανά μεταγλώττιση (gcc) και
3. Να εκτελείτε το πρόγραμμα (εφόσον δεν έχει άλλα λάθη) ώστε να βεβαιωθείτε ότι παράγει την αναμενόμενη έξοδο*.

***Προσοχή:** Δεν αρκεί να παραχθεί κάποια έξοδος. Πρέπει πάντα να ελέγχετε αν είναι σωστή ή τουλάχιστον αν είναι λογική. Για παράδειγμα, αν ένα πρόγραμμα πρέπει να υπολογίζει την απόσταση ανάμεσα σε δύο σημεία και παρατηρείτε ότι παράγει αρνητική τιμή, τότε προφανώς υπάρχει λάθος.

Παράδειγμα 2



```
lab0.c - Visual Studio Code
File Edit Selection View Go Run Terminal Help

C lab0.c x
local > ece > Desktop > prog1 > lab0 > C lab0.c > main(int, char *[])
1  /* Read a person's age and print a message about it */
2
3  #include<stdio.h>
4
5  int main (int argc, char *argv[]) {
6      int age;
7
8      printf("How old are you?\n");
9      scanf("%d", &age);
10
11     printf("You are %d years old!\n, age);
12
13     return 0;
14 }
15

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
bash - lab0 + - - -
ece@localhost:~/Desktop/prog1/lab0> gcc -Wall -g lab0.c -o lab0
lab0.c: In function 'main':
lab0.c:11:12: warning: missing terminating " character
printf("You are %d years old!\n, age);
^
lab0.c:11:12: error: missing terminating " character
printf("You are %d years old!\n, age);
^~~~~~
lab0.c:13:5: error: expected expression before 'return'
return 0;
^~~~~~
lab0.c:14:1: error: expected ';' before '}' token
}
^
ece@localhost:~/Desktop/prog1/lab0>
```

Εικόνα 14: Παράδειγμα λάθους μεταγλώττισης που οδηγεί σε πολλαπλά μηνύματα

Εδώ εμφανίζονται αρκετά μηνύματα λάθους. Σε αυτές τις περιπτώσεις, **επικεντρωνόμαστε ΠΑΝΤΑ στο πρώτο σφάλμα**, το διορθώνουμε, αποθηκεύουμε και μεταγλωττίζουμε ξανά. Ο λόγος είναι ότι όταν ο μεταγλωττιστής εντοπίσει ένα σφάλμα, τότε κάνει κάποια υπόθεση για το πώς θα μπορούσε να διορθωθεί και συνεχίζει τη δουλειά του. Αν αυτή η υπόθεση είναι λάθος, τότε πιθανώς να θεωρήσει ότι και ο κώδικας που ακολουθεί είναι λανθασμένος και να παράγει μηνύματα που δεν αντιστοιχούν σε πραγματικά λάθη.

Κάτι τέτοιο συμβαίνει και σε αυτό το παράδειγμα. Εμφανίζεται ένα warning και τρία errors ενώ στην πραγματικότητα έχει γίνει μόνο ένα λάθος. Μελετήστε το πρώτο μήνυμα. Σας λέει ότι στη γραμμή 11 υπάρχει το αρχικό " στο μήνυμα που θα εκτυπώσουμε, αλλά δεν υπάρχει το τελικό ". Πράγματι, θα έπρεπε να βρίσκεται ένα ακόμη " πριν το κόμμα. Αν το τοποθετήσουμε, αποθηκεύσουμε και ξαναμεταγλωττίσουμε, όλα τα μηνύματα λάθους θα εξαφανιστούν.

Μια ενδιαφέρουσα σημείωση: Δείτε πώς τα χρώματα του κειμένου μπορούν να σας βοηθήσουν να εντοπίσετε το λάθος την ώρα που γράφετε κώδικα. Ότι περικλείεται σε " " σε ένα πρόγραμμα εμφανίζεται με κόκκινο χρώμα στο VS Code. Επειδή λείπει το δεύτερο " χρωματίστηκε κόκκινος όλος ο κώδικας από το σημείο που ξεκίνησε το μήνυμα μέχρι το τέλος. Αυτό σας δίνει μια ιδέα ότι κάτι δεν πάει καλά.

Αφού υποβάλετε την άσκηση σας στο autolab, κι εφόσον έχουν πάει όλα καλά, κάντε και δικές σας δοκιμές: εισάγετε λάθη στον κώδικά σας και δείτε τι μηνύματα βγάζει. Αυτό θα σας βοηθήσει να μάθετε τα μηνύματα λάθους και τι ακριβώς σημαίνουν (μερικές φορές δεν είναι πολύ προφανές).

Υποβολή προγράμματος για σχολιασμό και βαθμολόγηση

Όταν είστε σίγουροι ότι το πρόγραμμά σας λειτουργεί σωστά, το υποβάλετε μέσω της πλατφόρμας autolab. Θα πρέπει να έχετε ήδη ενεργοποιήσει το λογαριασμό σας στο autolab. Αν όχι, ανατρέξτε στις σχετικές οδηγίες στα έγγραφα του eclass.

Άσκηση 11:

- Κάντε login στο [autolab](#). Μπορείτε να βρείτε το link και στο μενού του eclass.
- Κάντε tick στο μήνυμα ακαδημαϊκής ακεραιότητας (I affirm that I have complied...) και μετά:
 - ή βρίσκετε μέσω Dolphin το lab0.c και το κάνετε drag & drop στο πλαίσιο με τη διακεκομμένη γραμμή,
 - ή κάνετε click στο πλαίσιο κι εντοπίζετε το lab0.c μέσα από το παράθυρο που θα εμφανιστεί.
- Κάνετε click στο **SUBMIT** για να υποβληθεί η εργασία σας.
- Μετά από λίγα δευτερόλεπτα, κάνετε refresh τη σελίδα (ctrl-r ή click στο σχετικό κουμπί) για να δείτε το αποτέλεσμα της αυτόματης βαθμολόγησης. Ο τελικός βαθμός θα προκύψει μετά κι από δικό μας έλεγχο.

Απαντήσεις ερωτήσεων κατανόησης

Απάντηση 1: Το πρώτο πράγμα σε ένα μονοπάτι είναι το όνομα του καταλόγου από το οποίο ξεκινά. Το πρώτο μονοπάτι που παρουσιάστηκε στην ερώτηση ξεκινά από τον κατάλογο root ο οποίος γράφεται /. Το δεύτερο μονοπάτι ξεκινά από τον κατάλογο inf2023.

Απάντηση 2: `~/prog1/lab0`

Απάντηση 3:

Όλοι οι παρακάτω τρόποι είναι ισοδύναμοι στη συγκεκριμένη περίπτωση.

- `cd ~/Downloads/`
- `cd ../../Downloads/`
- `cd /home/inf2023/user1/Downloads`
- `cd ..` μετά `cd ..` και μετά `cd Downloads`