ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν Πανεπιστήμιον Αθηνών
ΙΔΡΥΘΕΝ ΤΟ 1837

Εισαγωγή στον Προγραμματισμό

Εργασία #0

Οκτώβριος 2024

Στόχος της εργασίας είναι να αποκτήσουμε εμπειρία με τα βασικά εργαλεία του προγραμματιστή και να γράψουμε ένα από τα πρώτα προγράμματα σε C. Συγκεκριμένα, στοχεύουμε γνωριμία με τα ακόλουθα:

1. git, version control, repositories, GitHub.
2. Γραμμή εντολών Linux και bash.
3. Χρήση ακεραίων στην C.
4. Χρήση βασικών δομών (loops, if, functions).
5. Δεδομένα εισόδου από πρότυπη είσοδο.

Υποβολή Εργασίας. Όλες οι υποβολές των εργασιών θα γίνουν μέσω GitHub και συγκεκριμένα στο github.com/progintro [7]. Προκειμένου να ξεκινήσεις, μπορείς να δεχτείς την άσκηση με αυτήν την: πρόσκληση [8].

1. Νέο URL στο GitHub (pages - 20 Μονάδες)

Έχεις λογαριασμό στο GitHub; Έχεις ξαναφτιάξει repository; Έχεις φτιάξει URL στο διαδίκτυο; Μετά από αυτήν την άσκηση θα μπορείς να απαντήσεις "ναι" σε όλα. Αν ήδη μπορείς να απαντήσεις ναι, η άσκηση αυτή (ελπίζουμε) να μην σου πάρει πολύ χρόνο.

Η ιστοσελίδα του μαθήματος βασίζεται σε ένα public GitHub repository [3]. Σκοπός αυτής της άσκησης είναι να δημιουργήσεις ένα καινούριο repository στον προσωπικό σου λογαριασμό GitHub και να το συνδέσεις με GitHub Pages προκειμένου να είναι διαθέσιμο στο διαδίκτυο. Αυτό το tutorial [1] μπορεί να σε βοηθήσει.

Τεχνικές Προδιαγραφές

Η υποβολή σου, πρέπει να έχει τα ακόλουθα χαρακτηριστικά:

- Repository Name: progintro/hw0-<YourUsername>
- Αρχείο Λύσης Filepath: pages/solution.txt
- README Filepath (optional): pages/README.md

Έστω ότι το GitHub username σου είναι YourUsername. Το αρχείο solution.txt πρέπει να περιέχει το URL με το domain YourUsername.github.io που δημιουργήσατε. Αν χρησιμοποιήσετε κάποιο *.github.io URL που δεν δημιουργήσατε η άσκηση μηδενίζεται. Ενδεικτική συμπεριφορά για μια επιτυχημένη υποβολή (**προσοχή: η γραμμή με την εντολή curl εκτείνεται σε δύο γραμμές, πρέπει να αντιγράψετε και τις δύο προκειμένου να τρέξετε αυτήν την εντολή**):

```
# Check out solution URL
$ cat solution.txt
YourUsername.github.io
# Ensure the URL exists
$ curl --output /dev/null --silent --head --fail YourUsername.github.io && \
  echo "URL exists" || echo "URL does not exist"
URL exists
```

Το αρχείο README.md είναι προαιρετικό αν θέλετε να προσθέσετε κάτι στην υποβολή σας.

2. Παιχνίδια με Κονσόλα (cmdline - 30 Μονάδες)

Έχεις χρησιμοποιήσει την γραμμή εντολών; Linux; Γνωρίζεις πως να κάνεις SSH σε απομακρυμένους servers και να λύνεις προβλήματα με εντολές κονσόλας; Αν όχι, καιρός να μάθουμε! Αν ναι, τότε αυτή η άσκηση θα σου είναι παιχνιδάκι.

Σε αυτήν την άσκηση θα μπούμε με ssh σε έναν απομακρυσμένο server που βρίσκεται στην IP διεύθυνση `52.86.144.139` μέσω τερματικού και θα ανακτήσουμε συγκεκριμένες πληροφορίες. Υπάρχουν 12 διαφορετικοί χρήστες (byte0-byte12) που αντιστοιχούν σε 12 διαφορετικά προβλήματα/υποερωτήματα. Για όλους τους χρήστες ο κωδικός για να μπείτε είναι `bits&bytes`. Σε κάθε ένα υποερώτημα θα αναζητούμε ένα string που μπορεί να είναι μια μόνο λέξη ή πολλές λέξεις διαχωρισμένες με τον χαρακτήρα underscore _ - για παράδειγμα: `something_like_that`.

byte0: Αλλάζοντας Φάκελο (cd)

Ο χρήστης `byte0` είναι ελαφρώς μανιακός με την οργανωτικότητα. Του αρέσει να οργανώνει όλα του τα αρχεία σε φακέλους, υποφακέλους κ.ο.κ.. Με αυτά και με αυτά όμως, δεν μπορεί να βρει το αρχείο του, μπορείτε να τον βοηθήσετε; Για να μπείτε στον server τρέξτε:

```
ssh byte0@52.86.144.139
```

και βάλτε τον κωδικό που γράψαμε παραπάνω. Η εντολή `cd` μπορεί να σας φανεί χρήσιμη όπως και η εντολή `cat` (ίσως και το tab / autocomplete :)). Όταν διαβάσετε τα περιεχόμενα του αρχείου, σημειώστε το string που βρήκατε μέσα στο αρχείο ώστε να το προσθέσετε στο solution.txt (δες παρακάτω στις τεχνικές προδιαγραφές).

byte1: Μια Άγνωστη Εντολή

Ο χρήστης `byte1` έφτιαξε μια εντολή `supercalifragilisticexpialidocious` αλλά δεν γνωρίζουμε τι κάνει. Τρέξτε την. Πως θα βρείτε τι κάνει;

byte2: Τα Άπαντα του Shakespeare (grep)

Ο χρήστης `byte2` έχει δύο βασικά χαρακτηριστικά: (1) είναι μεγάλος θαυμαστής του Shakespeare και (2) είναι κάπως μυστικοπαθής. Αποφάσισε να συνδυάσει τα δύο και καταχώνιασε ένα μυστικό string μέσα στην συλλογή του με τα άπαντα του Shakespeare. Μπορείτε να τον βοηθήσετε να το βρεί;

Το μόνο που γνωρίζουμε είναι πως το κρυφό κείμενο περιέχει το string "will find". Η εντολή `grep` μπορεί να σας φανεί χρήσιμη.

byte3: Τα Άπαντα του Shakespeare Άλλαξαν (diff)

Ο χρήστης `byte3` αποφάσισε να αλλάξει κάτι στα άπαντα του Shakespeare ελπίζοντας πως η αλλαγή θα περάσει απαρατήρητη. Ελέγξαμε τον αριθμό των λέξεων και χαρακτήρων του κάθε αρχείου:

```
byte3@ip-172-31-37-131:~$ wc -w *.txt
  901325 shakespeare.modified.txt
  901325 shakespeare.txt
 1802650 total
byte3@ip-172-31-37-131:~$ wc -c *.txt
 5458203 shakespeare.modified.txt
 5458199 shakespeare.txt
```

και παρατηρήσαμε πως ενώ ο αριθμός λέξεων παρέμεινε ίδιος (901.325), ο αριθμός των χαρακτήρων άλλαξε. Επομένως υποψιαζόμαστε πως άλλαξε μια λέξη του αυθεντικού με μια καινούρια. Βρείτε την καινούρια λέξη - η εντολή `diff` ίσως σας φανεί χρήσιμη.

byte4: Λαβύρινθος (find)

Ο χρήστης byte4 είναι συγγενής του byte2 παραπάνω και ελαφρώς πιο μυστικοπαθής. Προκειμένου να "ανεβάσει" το παιχνίδι του αποφάσισε να κρύψει τα δεδομένα του σε ένα δαιδαλώδες κατασκεύασμα από φακέλους. Μπορείτε και πάλι να τον βοηθήσετε να βρει το μυστικό που έχει κρύψει μέσα σε ένα από όλα τα αρχεία των υποφακέλων; Το μόνο που γνωρίζουμε είναι πως έχει κρύψει το μυστικό του σε ένα αρχείο που λέγεται cup.txt. Η εντολή find μπορεί να σας φανεί χρήσιμη.

byte5: Compile and Run (gcc)

Ο byte5 έγραψε ένα προγραμματάκι σε C ονόματι byte5.c. Μεταγλωττίστε το και τρέξτε το προκειμένου να πάρετε τον κωδικό σας! Προσοχή: ο default φάκελος του byte5 δεν σας επιτρέπει να χρησιμοποιήσετε αρχεία, ίσως χρειαστεί να φτιάξετε έναν καινούριο υποφάκελο μέσα στον φάκελο /tmp προκειμένου να τοποθετήσετε το εκτελέσιμό σας. Οι εντολές mkdir, cp, gcc μπορεί να σας φανούν χρήσιμες.

byte6: Αποσυμπίεση αρχείου 1 (unzip)

Κάποιες φορές προκειμένου να μικρύνουμε τον χώρο που πιάνει ένα αρχείο το συμπιέζουμε. Ο χρήστης byte6 έχει ένα αρχείο byte6.zip το οποίο χρειάζεται αποσυμπίεση. Ανοίξτε το για να αποκτήσετε πρόσβαση στα περιεχόμενά του. Η εντολή unzip μπορεί να σας φανεί χρήσιμη.

byte7: Αποσυμπίεση αρχείου 2 (tar)

Υπάρχουν διάφοροι τρόποι και εργαλεία για να συμπιέσεις αρχεία και τα περιεχόμενα τους. Ο χρήστης byte7 χρησιμοποίησε ένα πρόγραμμα που λέγεται tar - μπορείτε να χρησιμοποιήσετε το ίδιο προκειμένου να βρείτε τα περιεχόμενα του byte7.tar.gz;

byte8: Carriage Return (xxd, pico, vim)

Ο χρήστης byte8 μας είπε πως ο κωδικός που ψάχνουμε είναι μέσα στο αρχείο carriage_return.txt. Όμως δοκιμάσαμε να το τυπώσουμε:

```
byte8@ip-172-31-37-131:~$ ls
carriage_return.txt
byte8@ip-172-31-37-131:~$ cat carriage_return.txt
There is absolutely nothing to see here. Move along.
```

και δεν είδαμε κάτι. Μπορείς να μας βοηθήσεις; Πιστεύουμε πως οι εντολές xxd, pico ή vim μπορεί να φανούν χρήσιμες.

byte9: Ένα Περίεργο Όνομα (cat)

Ο χρήστης `byte9` ανέφερε πως ο κωδικός του είναι μέσα στον φάκελο `/home/byte9` - μπορείς να μας βοηθήσεις να τον ανακτήσουμε; Ίσως σε βοηθήσει να θυμηθείς τι είναι το filepath ενός αρχείου.

byte10: Το 10ο Όνομα (sort, head)

Ο χρήστης `byte10` έχει ένα αρχείο γεμάτο με ονόματα και θέλει να τα ταξινομήσει αλφαβητικά και να βρει το 10ο όνομα στην σειρά. Γνωρίζει ότι το 1ο όνομα είναι το `Aarika` - μπορείτε να τον βοηθήσετε να βρει το 10ο; Οι εντολές `sort` και `head` ίσως σας φανούν χρήσιμες.

byte11: Επιλογές Ονόματος (sort, uniq)

Ο χρήστης `byte11` μόλις απέκτησε ένα παιδάκι και προκειμένου να μην το κοροϊδεύουν στο σχολείο αποφάσισε να του δώσει το πιο συχνό όνομα εκείνης της χρονιάς. Στο αρχείο `births.txt` περιέχονται τα ονόματα όλων των φετινών γεννήσεων. Μπορείτε να τον βοηθήσετε να βρει το πιο συχνό όνομα; Οι εντολές `sort` και `uniq` ίσως σας φανούν χρήσιμες.

Τεχνικές Προδιαγραφές

Η υποβολή σου, πρέπει να έχει τα ακόλουθα χαρακτηριστικά:

- Repository Name: progintro/hw0-<YourUsername>
- README Filepath: cmdline/README.md
- Αρχείο Λύσης Filepath: cmdline/solution.txt
- **Κάθε γραμμή που θα προστεθεί στο αρχείο solution.txt πρέπει να γίνει με διαφορετικό git commit. Κοινώς, θέλουμε να δούμε *12* διαφορετικά git commits. Η κάθε γραμμή πρέπει να προστεθεί όταν λύσετε το αντίστοιχο πρόβλημα, όχι όλες μαζί στο τέλος.**

Το περιεχόμενο του solution.txt πρέπει να έχει την ακόλουθη μορφή:

```
$ cat solution.txt
byte0: ...
byte1: ...
byte2: ...
...
byte11: ...
```

Όπου οι τελείες (. . .) περιγράφουν το μέρος της λύσης που έχει παραληφθεί.

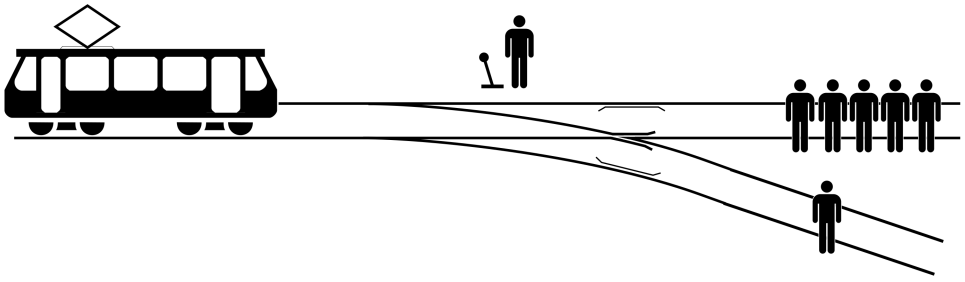
Στο αρχείο README.md πρέπει να γράψετε μια σύντομη περιγραφή για τον τρόπο που λύσατε το κάθε πρόβλημα.

Αν σου άρεσε αυτή η άσκηση ή έχεις κολλήσει με ένα από τα παραπάνω και δεν ξέρεις πως να συνεχίσεις, υπάρχουν και άλλα site online στα οποία μπορείς να εξασκηθείς όπως για παράδειγμα το overthewire [6] τα οποία μπορούν να σου δώσουν ιδέες. Σημείωση: μέχρι το επίπεδο 11 τα προβλήματα σε αυτό το site είναι εντός θέματος για το μάθημα, από εκεί και πέρα γίνονται πιο δύσκολα / απαιτούν γνώσεις που θα αποκτήσουμε σε μεγαλύτερα έτη.

3. Το Πρόβλημα του Τρόλεϊ (trolley - 50 Μονάδες)

Μετά τα ηλεκτρικά αυτοκίνητα, η νέα hip τεχνολογία που φαίνεται να μας πλησιάζει είναι τα αυτοκινούμενα οχήματα ή αλλιώς self-driving cars [4]. Τα αυτοκινούμενα οχήματα φέρνουν πολλές υποσχέσεις μαζί τους: οι μετακινήσεις θα γίνουν πιο ασφαλείς, οικονομικές, γρήγορες και άνετες. Θες να πας οπουδήποτε; Απλά λες τον προορισμό σου στο όχημα και σε πηγαίνει μόνο του ενώ εσύ παίρνεις έναν υπνάκο!

Παρόλο που η τεχνολογία έχει τρέξει πιλοτικά σε κάποιες περιορισμένες περιοχές, πολλές χώρες διστάζουν να περάσουν νομοθετικά πλαίσια που να επιτρέπουν ευρέως τα αυτόνομα οχήματα. Ένας από τους λόγους που συνήθως αναφέρεται όταν γίνονται αυτές οι συζητήσεις είναι το *Πρόβλημα του Τρόλεϊ*, ένα ηθικό πρόβλημα που προτάθηκε από την Philippa Foot το 1967 [5].



Σχήμα 1: Ένα από τα διλήμματα στο πρόβλημα του τρόλεϊ - στρίβει αριστερά ή δεξιά;

Το Σχήμα 1 εικονίζει μια απλοποιημένη εκδοχή του διλήμματος που πρέπει να λύσει οποιοδήποτε αυτοκινούμενο όχημα: φτάνει σε μια διασταύρωση και διαπιστώνει πως τα φρένα δεν λειτουργούν. Το όχημα πρέπει να στρίψει αριστερά ή δεξιά και κάθε επιλογή έχει κόστος. Πως θα επιλέξει και ποιες θα είναι οι επιπτώσεις αυτής της επιλογής; Το ηθικό αυτό πρόβλημα έχει τυραννήσει γενιές και γενιές φιλοσόφων [2].

Για τις ανάγκες αυτής της άσκησης, θα γράψετε ένα πρόγραμμα το οποίο λύνει επαναλαμβανόμενες εκδοχές του προβλήματος του τρόλεϊ όπως θα χρειαζόταν να τις

λύσει και ένας αυτόματος πιλότος σε ένα αυτοκινούμενο όχημα. Επειδή η άσκηση μας είναι κυρίως προγραμματιστική, δεν θα αποπειραθούμε να λύσουμε τα ηθικά διλήμματα και θα ζητάμε από τον χρήστη να μας ενημερώσει για το κόστος της κάθε επιλογής μας αν πάμε αριστερά ή δεξιά. Προχωρήστε στις τεχνικές προδιαγραφές παρακάτω για λεπτομέρειες της υλοποίησης.

Τεχνικές Προδιαγραφές

- Repository Name: progintro/hw0-<YourUsername>
- Αρχείο C (Filepath): trolley/src/trolley.c
- Το αρχείο C που θα υποβληθεί πρέπει να μεταγλωττίζεται χωρίς ειδοποιήσεις για λάθη και με κωδικό επιστροφής (exit code) που να είναι 0. Συγκεκριμένα, το αρχείο σας **πρέπει** να μπορεί να μεταγλωττιστεί επιτυχώς με την ακόλουθη εντολή σε ένα από τα μηχανήματα του εργαστηρίου (linuxXY.di.uoa.gr):

  ```
  gcc -O0 -m32 -Wall -Wextra -Werror -pedantic -o trolley trolley.c
  ```

- README Filepath: trolley/README.md
- Το πρόγραμμά σας πρέπει να διαβάζει όλα τα κόστη που έδωσε ο χρήστης μέχρι το τέλος αρχείου (End Of File - EOF).
- Όλα τα κόστη που θα δοθούν στο πρόγραμμά σας θα είναι δεκαδικοί ακέραιοι στο εύρος: -10^{18} έως 10^{18}.
- Σε κάθε επανάληψη του προβλήματος το πρόγραμμά σας θα πρέπει να ζητάει από τον χρήστη να δώσει τα κόστη για δύο επιλογές: αν πάει αριστερά (left) ή αν πάει δεξιά (right). Το πρόγραμμά σας πρέπει να ζητάει το κόστος για την αριστερή επιλογή πρώτα και στην συνέχεια για την δεξιά.
- Τα κόστη που θα δώσει ο χρήστης μπορούν να διαχωρίζονται με κενό (space), tab ή νέα γραμμή (new line).
- Για κάθε δύο κόστη που δίνονται, το πρόγραμμά σας πρέπει να τυπώνει στην πρότυπη έξοδο (stdout) είτε `Go left` είτε `Go right` - διαλέγοντας πάντα την επιλογή με το μικρότερο κόστος. Αν οι δύο επιλογές έχουν το ίδιο κόστος, το πρόγραμμά σας πρέπει να τυπώνει `Go left`.
- Αν ο χρήστης δώσει EOF αντί για το πρώτο κόστος το πρόγραμμά σας πρέπει να τερματίζει με κωδικό εξόδου (exit code) 0.
- Αν δοθεί οποιαδήποτε είσοδος εκτός προδιαγραφών (για παράδειγμα ο χρήστης δίνει μόνο το κόστος της αριστερής επιλογής) το πρόγραμμά σας πρέπει να τερματίζει με κωδικό εξόδου (exit code) 1.

- Για 10.000 επαναλήψεις του πειράματος, το πρόγραμμά σας πρέπει να ολοκληρώνει την εκτέλεση μέσα σε: 1 δευτερόλεπτο.

Παρακάτω παραθέτουμε αλληλεπιδράσεις με μια ενδεικτική λύση:

```
thanassis@linux14:~$ hostname
linux14
thanassis@linux14:~$ gcc -O0 -m32 -Wall -Wextra -Werror -pedantic -o trolley trolley.c
thanassis@linux14:~$ ./trolley
Please enter the cost of going left: 10
Please enter the cost of going right: 100
Go left.
Please enter the cost of going left: 41
Please enter the cost of going right: 42
Go left.
Please enter the cost of going left: 42
Please enter the cost of going right: 41
Go right.
Please enter the cost of going left: 1000000
Please enter the cost of going right: 1000000
Go left.
Please enter the cost of going left: Terminating.
thanassis@linux14:~$ echo $?
0
thanassis@linux14:~$ ./trolley
Please enter the cost of going left: 42
Please enter the cost of going right: No right cost provided.
thanassis@linux14:~$ echo $?
1
thanassis@linux14:~$ ./trolley
Please enter the cost of going left: 10000000000000
Please enter the cost of going right: 4
Go right.
Please enter the cost of going left: -4
Please enter the cost of going right: -4
Go left.
Please enter the cost of going left: 10000000000000
Please enter the cost of going right: 4000000000000
Go right.
Please enter the cost of going left: Terminating.
thanassis@linux14:~$ echo $?
0
```

Στο αρχείο README.md πρέπει να προσθέσετε οποιεσδήποτε παρατηρήσεις σας κατά την διεκπεραίωση της άσκησης. Ο κώδικας απαιτείται να είναι καλά τεκμηριωμένος με σχόλια καθώς αυτό θα είναι μέρος της βαθμολόγησης.

Αναφορές

[1] GitHub Pages tutorial . https://docs.github.com/en/pages/quickstart.

[2] Justice: What's the right thing to do? (WARNING: μην πατήσεις αυτό το link αν είσαι κοντά στην προθεσμία για την υποβολή της άσκησης). https://www.youtube.com/watch?v=kBdfcR-8hEY.

[3] Repository για το μάθημα . https://github.com/progintro/progintro.github.io.

[4] Self-Driving Cars . https://en.wikipedia.org/wiki/Self-driving_car.

[5] The Trolley Problem . https://en.wikipedia.org/wiki/Trolley_problem.

[6] Ασκήσεις σε Γραμμή Εντολών . https://overthewire.org/wargames/bandit/bandit0.html.

[7] Οργανισμός για το μάθημα (GitHub progintro) . https://github.com/progintro.

[8] Πρόσκληση για Εργασία 0 . https://classroom.github.com/a/sdllo8YY.