

Διαθέσιμα αρχεία ελέγχου

Για κάθε έλεγχο X σας δίνουμε τα παρακάτω αρχεία:

argsX: Τα ορίσματα που πρέπει να χρησιμοποιήσετε κατά την εκτέλεση του προγράμματός σας.

inX: Η συμβατική είσοδος που λαμβάνει το πρόγραμμα.

outX: Η αναμενόμενη έξοδος.

Παράδειγμα ελέγχου συμβατικής εξόδου

Για το τεστ 3b σας δίνουμε το αρχείο args3b, το οποίο περιλαμβάνει τους ακραίους 10 10, και τα αρχεία in3b, out3b.

Αρχικά δοκιμάστε να εκτελέσετε τον κώδικά σας χωρίς ανακατεύθυνση εξόδου. Ο λόγος είναι ότι πριν ανακατευθύνετε την έξοδο σε αρχείο, πρέπει να βεβαιωθείτε ότι δεν γίνεται ατέρμονη επανάληψη (ώστε να μη δημιουργηθεί κάποιο τεράστιο αρχείο το οποίο πιθανώς να σας γεμίσει τον δίσκο):

```
./project1 10 10 < in3b
```

Εάν το πρόγραμμα τερματίσει κανονικά, εκτελέστε το ξανά κι αυτή τη φορά ανακατευθύνετε την έξοδο σε αρχείο:

```
./project1 10 10 < in3b > my3b
```

Και μετά συγκρίνετε τη δική σας έξοδο με τη δική μας. Μην κάνετε σύγκριση με το μάτι γιατί είναι εύκολο να σας ξεφύγει κάτι. Συνίσταται να χρησιμοποιήσετε την εντολή `diff`:

```
diff out3b my3b
```

Αν δεν υπάρχουν διαφορές, δε θα δείτε τίποτα.

Αν υπάρχουν διαφορές, θα δείτε κάτι σαν το παρακάτω :

```
13a14,15
> A-NOK 122, 5 10
>
22,23d23
< A-OK 4, 8 10
<
43c43
< 122 MARY-ANNE LOCKHART 19
---
> 122 MARY ANNE LOCKHART 19
```

Αυτό σημαίνει πως βρέθηκαν τρία σημεία με διαφορές. Οι πληροφορίες για κάθε διαφορά ξεκινούν στη γραμμή που εμφανίζεται αριθμός (13, 22, 43). Ακολουθεί επεξήγηση της εξόδου:

```
13a14,15
> A-NOK 122, 5 10
>
```

Αυτό σημαίνει ότι μετά τη γραμμή 13 του πρώτου αρχείου που δόθηκε στην εντολή diff (δηλαδή του out3b) πρέπει να προστεθούν (a == add) τα περιεχόμενα των γραμμών 14-15 του δεύτερου αρχείου έτσι ώστε τα αρχεία να ταιριάζουν. Οι εν λόγω γραμμές παρατίθενται αμέσως μετά.

Ουσιαστικά, το δεύτερο αρχείο (δηλαδή η δική σας έξοδο) περιέχει δύο επιπλέον γραμμές, την A-NOK 122, 5 10 και μια κενή γραμμή.

```
22,23d23
< A-OK 4, 8 10
<
```

Αυτό σημαίνει ότι πρέπει να σβηστούν (d == delete) οι γραμμές 22-23 του πρώτου αρχείου (οι οποίες παρατίθενται αμέσως μετά) ώστε να ταιριάζουν τα δύο αρχεία. Ουσιαστικά, το πρώτο αρχείο περιέχει δύο επιπλέον γραμμές (την A-OK 4,8 10 και μια κενή), το οποίο σημαίνει ότι το πρόγραμμά σας δεν εκτύπωσε κάποιο μήνυμα που έπρεπε να εκτυπώσει.

```
43c43
< 122 MARY-ANNE LOCKHART 19
---
> 122 MARY ANNE LOCKHART 19
```

Αυτό σημαίνει ότι η γραμμή 43 του πρώτου αρχείου που δόθηκε στην εντολή diff (του out3b) πρέπει να αλλάξει (c == change) για να μοιάζει με τη γραμμή 43 του δεύτερου αρχείου. Ακολουθούν οι εν λόγω γραμμές, πρώτα η γραμμή 43 του πρώτου αρχείου και μετά η αντίστοιχη γραμμή του δεύτερου. Παρατηρήστε πως η δεύτερη γραμμή δεν περιέχει παύλα ανάμεσα στα ονόματα.

Παράδειγμα ελέγχου εξόδου σφαλμάτων

Εάν θέλετε να κατευθύνετε και την έξοδο stderr σε αρχείο, κι εφόσον το τερματικό σας τρέχει bash shell (είναι το default, αλλά μπορείτε και να το επιβεβαιώσετε γράφοντας `echo $0` στο τερματικό), γράφετε:

```
./project1 0 2 < in5b > my5b 2> err5b
```

Στο αρχείο `err5b` θα έχουν αποθηκευτεί τα πλήθη συγκρίσεων κατά τις λειτουργίες `find` του τεστ 5b. Αυτό θα σας φανεί ιδιαίτερα χρήσιμο για τη μελέτη απόδοσης.

Έλεγχοι autolab

Θα σας δώσουμε ένα αντιπροσωπευτικό υποσύνολο των τεστ που κάνει το autolab. Όσα τεστ σας δίνουμε αντιστοιχούν στα ομώνυμα τεστ του autolab. Περιμένουμε πως θα κάνετε κι εσείς δικά σας μικρά τεστ για να ελέγξετε κάθε μία από τις λειτουργίες καθώς και συνδυασμούς.

Κάθε τεστ του autolab αποτελείται από 1-4 υποτέστ που ελέγχουν πιο εξειδικευμένες περιπτώσεις.

Το υποτέστ με όνομα prerequisite ελέγχει την στοιχειώδη ορθότητα της λειτουργίας αναζήτησης. Στις περιπτώσεις που χρησιμοποιείται, αν δε λειτουργεί σωστά αυτό, τότε δε γίνονται οι υπόλοιποι έλεγχοι του συγκεκριμένου τεστ.

Σχεδόν σε όλα τα τεστ η μεταγλώττιση γίνεται με -fsanitize=address. Επιπλέον, στα τεστ 4, 6, 7, 9 γίνεται έλεγχος για memory leaks με χρήση valgrind.

[*] Test0, Λάθη στα ορίσματα προγράμματος

- subtests 0a, 0b: περισσότερα ορίσματα από όσα πρέπει.
- subtests 0c, 0d: λιγότερα ορίσματα από όσα πρέπει.

[*] Test1, Οριακές περιπτώσεις διαχείρισης μνήμης

- subtest 1a: Βασικός έλεγχος quit.
- subtest 1b: Βασικός έλεγχος clear.
- subtest 1c: Αποτυχία malloc.

[*] Test2, Λανθασμένη είσοδος κατά την add.

- subtest 1a: Μήκος ονομάτων μεγαλύτερο του NAMESIZE.
- subtest 1b: Έλεγχος scanf για NAMESIZE ίσο 50.
- subtest 1c: Έλεγχος scanf για NAMESIZE ίσο 2.
- subtest 1d: Μηδενικά σε AEM ή/και πλήθος μαθημάτων.

[*] Test3, add χωρίς μεγέθυνση πίνακα

- subtest 3a: μοναδικά AEM
- subtest 3b: γίνεται προσπάθεια προσθήκης ήδη υπάρχοντος AEM
- subtest 3c: Ομοίως με b και υπάρχει επιπλέον whitespace ανάμεσα στα τμήματα εντολών.

[*] Test4, add με μεγέθυνση πίνακα

- prerequisite
- subtest 4a: μοναδικά AEM
- subtest 4b: γίνεται προσπάθεια προσθήκης ήδη υπάρχοντος AEM
- subtest 4c: παρόμοιο με b, διαφορετικά ορίσματα

[*] Test5, find, modify

- subtest 5a: έλεγχος αριθμού συγκρίσεων σε μη ταξινομημένο πίνακα.
- subtest 5b: modify με μεγέθυνση πίνακα.
- subtest 5c: modify με λανθασμένη είσοδο (μηδενικά)

[*] Test6, remove

- prerequisite
- subtest 6a: χωρίς μεγέθυνση, γίνονται μόνο αφαιρέσεις εγγραφών.
- subtest 6b: χωρίς μεγέθυνση, γίνονται και προσθήκες μετά τις αφαιρέσεις.
- subtest 6c: με μεγέθυνση, γίνονται μόνο αφαιρέσεις εγγραφών.
- subtest 6d: με μεγέθυνση, γίνονται και προσθήκες μετά τις αφαιρέσεις.

[*] Test7 συνδυασμός εντολών

- prerequisite
- subtest 7a: προσθαφαιρέσεις με ενδιάμεσες clear
- subtest 7b: προσθαφαιρέσεις και mod με ενδιάμεσες clear

[*] Test8, ταξινόμηση (δε γίνεται έλεγχος αριθμού συγκρίσεων)

- subtest 8a: sort και find
- subtest 8b: sort και ακολούθως επιπλέον add.
- subtest 8c: sort, remove
- subtest 8d: sort και ακολούθως add, remove

[*] Test9, συνδυασμός εντολών

- Τυχαίες εντολές για πολλά ονόματα.

[*] Test10, #comparisons

- subtest 10a: Έλεγχος αριθμού συγκρίσεων κατά την ταξινόμηση ήδη ταξινομημένου πίνακα και κατά την αναζήτηση σε ταξινομημένη λίστα. Θα γίνει δεκτή απόκλιση κατά 1
- subtest 10b: Έλεγχος αριθμού συγκρίσεων κατά την ταξινόμηση ανάποδα ταξινομημένου πίνακα.
- subtest 10c: Έλεγχος αριθμού συγκρίσεων κατά την ταξινόμηση τυχαίου πίνακα.

ΠΡΟΣΟΧΗ:

Στο autolab υπάρχει όριο 35 υποβολών χωρίς ποινή. Κάθε υποβολή πλέον των 35 έχει βαθμολογικό πέναλτυ 1%.

Μελέτη απόδοσης

Για τη μελέτη της απόδοσης, σας δίνουμε κάποια αρχεία με μεγάλο αριθμό εντολών προσθήκης εγγραφών (1000, 10000, 100000) με σειρά ταξινομημένη αύξουσα, ταξινομημένη φθίνουσα και τυχαία. Προσθέστε σε αυτά εντολές find ή/και sort ανάλογα με τις μετρήσεις που θέλετε να κάνετε.