

Εργασία 4 – Αυτόματος έλεγχος και βαθμολόγηση προγραμμάτων C

Αναπτύξτε μια εφαρμογή για τον αυτόματο έλεγχο και βαθμολόγηση προγραμμάτων C (κοντά στο πνεύμα του autolab). Η επιθυμητή λειτουργικότητα περιγράφεται παρακάτω. Αποθηκεύστε το κυρίως πρόγραμμα της εφαρμογής σε αρχείο με όνομα `project4.c`.

Ορίσματα

Το κυρίως πρόγραμμα της εφαρμογής δέχεται ως ορίσματα:

(α) Το όνομα ενός αρχείου, στη μορφή `<progname>.c`, που περιέχει τον κώδικα του προγράμματος προς βαθμολόγηση.

(β) Το όνομα ενός αρχείου κειμένου `<progname>.args` που περιέχει τα ορίσματα που απαιτούνται για την εκτέλεση του προγράμματος `<progname>.c`, σε μία γραμμή, με χαρακτήρα ' ' (space) ανάμεσα σε διαδοχικά ορίσματα. Δε μπορείτε να κάνετε υποθέσεις για το μήκος της γραμμής ούτε για τον αριθμό των ορισμάτων.

(γ) Το όνομα ενός αρχείου κειμένου `<progname>.in` που περιέχει τα δεδομένα που θα δεχτεί το πρόγραμμα `<progname>.c` από τη συμβατική είσοδό του.

(δ) Το όνομα ενός αρχείου κειμένου `<progname>.out` που περιέχει την αναμενόμενη έξοδο που πρέπει να παράξει το πρόγραμμα `<progname>.c` όταν εκτελεστεί με τα ορίσματα που προσδιορίστηκαν στο αρχείο `<progname>.args` και την είσοδο που προσδιορίστηκε στο αρχείο `<progname>.in`.

(ε) Μια θετική ακέραια τιμή `<timeout>` που αναπαριστά το μέγιστο πραγματικό χρόνο (real time) εκτέλεσης σε δευτερόλεπτα του προγράμματος `<progname>.c`.

Υπολογισμός τελικού βαθμού και έξοδος

Ο τελικός βαθμός που θα ανατεθεί από την εφαρμογή σας στο πρόγραμμα `<progname>.c` προκύπτει από το άθροισμα των επιμέρους βαθμών στις παρακάτω κατηγορίες:

(α) Μεταγλώττιση: Εάν απέτυχε, -100. Εάν πέτυχε αλλά παρήγαγε warnings, -5 ανά warning. Διαφορετικά 0.

(β) Αν το πρόγραμμα προς βαθμολόγηση δεν έχει τερματίσει μετά από `<timeout>` δευτερόλεπτα, -100, διαφορετικά 0.

(γ) Ομοιότητα της εξόδου του προγράμματος με την αναμενόμενη: το ακέραιο $\frac{\text{πλήθος } \langle \text{πλήθος bytes} \rangle * 100}{\max(\langle \text{πλήθος bytes αναμενόμενης εξόδου} \rangle, \langle \text{πλήθος bytes εξόδου} \rangle)}$.

(δ) Ποινή λόγω λανθασμένης προσπέλασης μνήμης : -15 αν το πρόγραμμα τερμάτισε λόγω σήματος SIGSEGV ή SIGABRT ή SIGBUS.

Η εφαρμογή σας στο τέλος εκτυπώνει στη συμβατική έξοδο τα εξής μηνύματα:

Compilation: C

Timeout: T

Output: O

Memory access: M

Score: S

με χαρακτήρα '\n' πριν και μετά από κάθε μήνυμα. Οι τιμές C, T, O, M είναι οι βαθμοί που υπολογίστηκαν στις αντίστοιχες επιμέρους κατηγορίες (α), (β), (γ) και (δ) όπως περιγράφονται στην προηγούμενη παράγραφο, ενώ το S είναι ίσο με $\max(0, C+T+O+M)$. Αυτά τα μηνύματα εκτυπώνονται ακόμη κι αν οι αντίστοιχοι βαθμοί είναι μηδέν.

Λειτουργία

Αν δεν έχει δοθεί ο σωστός αριθμός ορισμάτων με βάση την παραπάνω περιγραφή, η εφαρμογή εκτυπώνει κατάλληλο μήνυμα λάθους στην έξοδο λαθών και τερματίζει επιστρέφοντας την τιμή 1

Αν έχει δοθεί ο σωστός αριθμός ορισμάτων, τότε:

(α) Το κυρίως πρόγραμμα της εφαρμογής δημιουργεί μία νέα διεργασία P1 και ανακατευθύνει την έξοδο λαθών της σε ένα αρχείο με όνομα `<progrname>.err`. Η P1 παράγει το εκτελέσιμο για το πρόγραμμα `<progrname>.c` εκτελώντας την εντολή:

```
gcc -Wall <progrname>.c -o <progrname>
```

Το κυρίως πρόγραμμα περιμένει να τερματίσει η P1 και μετά ελέγχει αν το αρχείο `<progrname>.err` περιλαμβάνει (τουλάχιστον μια φορά) ως μεμονωμένη λέξη τη συμβολοσειρά "error:" πράγμα που σημαίνει ότι η μεταγλώττιση απέτυχε ή, αν όχι, τη συμβολοσειρά "warning:" το οποίο σημαίνει ότι το πρόγραμμα `<progrname>.c` μεταγλωττίζεται επιτυχώς αλλά με warnings. Σε κάθε περίπτωση υπολογίζεται η αντίστοιχη ποινή (αρνητικός βαθμός).

(β) Εφόσον η μεταγλώττιση είναι επιτυχής, το κυρίως πρόγραμμα δημιουργεί δύο νέες διεργασίες P2 και P3 και έναν ανώνυμο αγωγό A, και ανακατευθύνει τη συμβατική είσοδο της P2 στο αρχείο κειμένου `<progrname>.in`, τη συμβατική έξοδο της P2 στο άκρο εγγραφής του αγωγού A και τη συμβατική είσοδο της P3 στο άκρο ανάγνωσης του αγωγού A.

Η διεργασία P2 εκτελεί το πρόγραμμα `<progrname>` με τα ορίσματα που προσδιορίζονται στο αρχείο `<progrname>.args`.

Η διεργασία P3 εκτελεί το πρόγραμμα `p4diff` με όρισμα το όνομα του αρχείου `<progrname>.out`. Το πρόγραμμα `p4diff` πρέπει να το υλοποιήσετε εσείς έτσι ώστε να συγκρίνει ένα προς ένα byte τα δεδομένα που διαβάζει από τη συμβατική του είσοδο με τα περιεχόμενα του αρχείου `<progrname>.out`, να υπολογίζει το ποσοστό ομοιότητας όπως περιγράφεται στην προηγούμενη ενότητα και να το επιστρέφει ως αποτέλεσμα μέσω `return`. Αποθηκεύστε τον πηγαίο κώδικα του προγράμματος `p4diff` σε αρχείο με όνομα `p4diff.c`.

Το κυρίως πρόγραμμα δημιουργεί ένα ξυπνητήρι για `<timeout>` δευτερόλεπτα και περιμένει τον τερματισμό της P2. Εάν το ξυπνητήρι χτυπήσει πριν τερματίσει η P2 (δηλ. διακόψει τη λειτουργία αναμονής τερματισμού του παιδιού), τότε το κυρίως πρόγραμμα της στέλνει σήμα SIGKILL. Διαφορετικά, το κυρίως πρόγραμμα καταργεί το ξυπνητήρι και ελέγχει τον λόγο τερματισμού της P2. Αν η P2 τερμάτισε λόγω σήματος SIGSEGV ή SIGABRT ή SIGBUS, υπολογίζεται η ποινή για λανθασμένη προσπάθεια μνήμης όπως περιγράφεται στην προηγούμενη ενότητα, διαφορετικά θεωρείται ότι το πρόγραμμα `<progrname>` εκτελέστηκε χωρίς πρόβλημα οπότε δεν υπάρχει κάποια ποινή.

Ακολούθως, το κυρίως πρόγραμμα περιμένει τον τερματισμό της P3, και παίρνει μέσω του `exit status` της το βαθμό που υπολόγισε η P3.

(γ) Τέλος, το κυρίως πρόγραμμα εκτυπώνει τα μηνύματα βαθμολογίας όπως περιγράφονται στην προηγούμενη ενότητα.



Απαιτήσεις/Υποθέσεις υλοποίησης

- Παρέχετε κατάλληλο `Makefile` για την εφαρμογή σας.
- Απαγορεύεται η χρήση καθολικών (global) ή static μεταβλητών, goto, gets, sleep και η χρήση της εντολής system.
- Οι διεργασίες P2 και P3 τρέχουν ταυτόχρονα.
- Το πρόγραμμα `p4diff` διαβάζει δεδομένα από το αρχείο και τον αγωγό σε blocks των 64 bytes.
- Για την ανάγνωση από/εγγραφή σε συμβατική είσοδο/έξοδο μπορούν να χρησιμοποιηθούν είτε λειτουργίες της βιβλιοθήκης stdio είτε κλήσεις συστήματος.
- Μπορείτε να υποθέσετε ότι το όνομα του αρχείου `<progname>.c` θα έχει πάντα τη σωστή κατάληξη (`.c`) καθώς και ότι τα ονόματα των αρχείων `<progname>.args`, `<progname>.in` και `<progname>.out` θα έχουν τη σωστή μορφή και κατάληξη.

Σημαντικές ημερομηνίες

Φροντιστήριο: Πέμπτη 27/5/2021 (στην ώρα του μαθήματος)

Προθεσμία υποβολής: Κυριακή 12/6/2021, 21:00

Πακετάρισμα και αποστολή εργασίας

1. Κατασκευάστε ένα κατάλογο με όνομα `project4`
2. Αντιγράψτε τα `project4.c`, `p4diff.c`, `Makefile` στον κατάλογο `project4`
3. Πακετάρετε και συμπιέστε τον κατάλογο `project4` είτε κάνοντας δεξί κλικ κι επιλέγοντας Compress here as `.tar.gz` είτε μέσω της εντολής `tar` στο τερματικό
4. Στο autolab υποβάλετε το `project4.tar.gz` που δημιουργήσατε στο προηγούμενο βήμα.

Για όσους έχουν κέφι

Αλλάξτε το κυρίως πρόγραμμα ώστε αντί για τα ορίσματα `<progname>.args`, `<progname>.in`, `<progname>.out` να λαμβάνει ως όρισμα το όνομα ενός καταλόγου μέσα στον οποίο βρίσκονται

υποκατάλογοι, ένας για κάθε έλεγχο. Σε κάθε υποκατάλογο περιλαμβάνονται τα παρακάτω αρχεία κειμένου:

- `test.args` με τα ορίσματα που απαιτούνται γι αυτόν τον έλεγχο.
- `test.in` με τα δεδομένα που αποτελούν τη συμβατική είσοδο γι αυτόν τον έλεγχο.
- `test.out` με την αναμενόμενη έξοδο του προγράμματος γι αυτόν τον έλεγχο.

Προσαρμόστε αναλόγως τον τρόπο υπολογισμού του τελικού βαθμού θεωρώντας ότι όλα τα τεστ είναι ισότιμα και το συνολικό άθροισμα πόντων (για την κατηγορία output) είναι 100.