

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 `hw4.c`.

Ορίσματα

Το κυρίως πρόγραμμα της εφαρμογής που πρέπει να αναπτύξετε δέχεται ως ορίσματα:

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

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

(γ) Το όνομα ενός αρχείου κειμένου `<progname>.in` που περιέχει τα δεδομένα που θα δεχτεί το πρόγραμμα `<progname>.c` από τη συμβατική είσοδό του όταν εκτελεστεί με τα ορίσματα που προσδιορίστηκαν στο αρχείο `<progname>.args`.

(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 καθορίζει το μέγιστο αποδεκτό 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/ποινών στις παρακάτω κατηγορίες:

(α) *Μεταγλώττιση*: Εάν η μεταγλώττιση του προγράμματος απέτυχε, ποινή -100 ανεξαρτήτως του πλήθους των errors και warnings. Εάν πέτυχε αλλά παρήγαγε warnings, ποινή -5 ανά warning (προσμετρώνται όλα τα warnings, ακόμη κι αν αυτό σημαίνει πως η απόλυτη τιμή του αθροίσματος υπερβαίνει το 100). Διαφορετικά 0 (δεν υπάρχει ποινή).

(β) *Τερματισμός*: Αν το πρόγραμμα προς βαθμολόγηση δεν τερματίσει μέσα σε `<timeout>` δευτερόλεπτα, ποινή -100, διαφορετικά 0.

(γ) *Ομοιότητα εξόδου*: Το ακέραιο πηλίκο $M * 100 / \max(N, A)$, όπου M είναι το πλήθος των όμοιων bytes μεταξύ της αναμενόμενης (σωστής) εξόδου και αυτής που παρήγαγε το πρόγραμμα, N είναι το πλήθος bytes της αναμενόμενης εξόδου, και A είναι το πλήθος bytes της εξόδου που παρήγαγε το πρόγραμμα. Εάν το N το και το A είναι μηδέν, ο βαθμός είναι 100.

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

Η εφαρμογή σας στο τέλος εκτυπώνει στη συμβατική έξοδο τα εξής 5 μηνύματα:

```
Compilation: C
Termination: 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/ποινές που υπολογίστηκαν στις αντίστοιχες επιμέρους κατηγορίες (α), (β), (γ) και (δ) όπως περιγράφονται στην προηγούμενη παράγραφο, ενώ ο τελικός βαθμός S είναι ίσος 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 P1 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 ελέγχει την τιμή επιστροφής της. Αν αυτή είναι 1, τότε η μεταγλώττιση απέτυχε, υπολογίζεται η σχετική ποινή και το πρόγραμμα εκτυπώνει τα μηνύματα βαθμολογίας όπως περιγράφονται στην προηγούμενη ενότητα κι επιστρέφει 0. Διαφορετικά, εντοπίζει πόσα warnings έχουν καταγραφεί στο αρχείο `<progrname>.err`, ώστε να υπολογίσει τη σχετική ποινή. Ο υπολογισμός λαμβάνει υπόψη ότι κάθε warning εμφανίζεται σε μια γραμμή που ξεκινά με `<progrname>.c:<line>:<column>: warning:`

(β) Εφόσον η μεταγλώττιση είναι επιτυχής, το κυρίως πρόγραμμα της εφαρμογής 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 εκτελέσιμο πρόγραμμα `<progrname>` με τα ορίσματα που προσδιορίζονται στο αρχείο `<progrname>.args`.

Η διεργασία P3 εκτελεί το πρόγραμμα `p4diff` (που πρέπει να υλοποιήσετε εσείς) με όρισμα το όνομα του αρχείου `<progrname>.out`. Το πρόγραμμα `p4diff` διαβάζει σε blocks των 64 bytes δεδομένα από τη συμβατική του είσοδο και σε blocks των 64 bytes τα περιεχόμενα του αρχείου που παίρνει ως όρισμα, και συγκρίνει ένα προς ένα τα bytes των αντίστοιχων blocks ώστε να υπολογίσει το ποσοστό ομοιότητας όπως περιγράφεται σε προηγούμενη ενότητα. Το ποσοστό ομοιότητας 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 μέσω κατάλληλης λειτουργίας αναμονής τερματισμού παιδιού. Εάν το ξυπνητήρι χτυπήσει πριν τερματίσει η P2 (δηλ. διακόψει τη λειτουργία αναμονής), τότε το κυρίως πρόγραμμα στέλνει στην P2 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 προσπέλαση μνήμης όπως περιγράφεται στην προηγούμενη ενότητα, διαφορετικά (αν η P2 τερμάτισε κανονικά ή λόγω άλλου σήματος) θεωρείται ότι το πρόγραμμα `<progname>` εκτελέστηκε χωρίς πρόβλημα οπότε δεν υπάρχει κάποια ποινή.

Σε κάθε περίπτωση, 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.



Makefile

Παρέχετε κατάλληλο Makefile για την εφαρμογή σας. Εκτέλεση της εντολής make πρέπει να παράγει τα εκτελέσιμα hw4 και p4diff από τα αντίστοιχα .c αρχεία έχοντας κάνει ξεχωριστή μεταγλώττιση του καθενός, εφόσον χρειάζεται.

Απαιτήσεις/υποθέσεις

Γενικά θέματα:

- Απαγορεύεται η χρήση goto, gets, wordexp, sleep και system καθώς και η εκτέλεση προγραμμάτων/scripts εκτός από αυτά που περιγράφει η εκφώνηση.
- Απαγορεύεται η χρήση καθολικών (global) ή static μεταβλητών με μοναδική εξαίρεση μια global volatile μεταβλητή για χρήση σε χειριστή σήματος.
- 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.
- Πρέπει να ελέγχονται οι τιμές επιστροφής όλων των κλήσεων συστήματος. Εάν δεν ισχύει αυτό θα υπάρξει μείωση βαθμού (έως 15 πόντοι) μετά την οριστική υποβολή της εργασίας.

Σε περίπτωση αποτυχίας κάποιας κλήσης συστήματος, εκτυπώνεται κατάλληλο μήνυμα στην έξοδο σφαλμάτων κι επιστρέφεται η τιμή 255. Εάν κάποια από τις διεργασίες P1, P2, P3 επιστρέψει 255, η διεργασία γονιός τερματίζει τυχόν άλλες διεργασίες-παιδιά που εκτελούνται, κλείνει όλους τους ανοιχτούς περιγραφείς αρχείων κι επιστρέφει επίσης 255.

Διεργασίες

- Η διεργασία P1 δημιουργείται, τρέχει και τερματίζει προτού δημιουργηθούν οι διεργασίες P2 και P3.
- Οι διεργασίες P2 και P3 πρέπει να τρέχουν ταυτόχρονα. Εάν δεν ισχύει αυτό, θα υπάρξει μείωση βαθμού (**έως 10 πόντοι**) μετά την οριστική υποβολή της εργασίας.
- Η P3 πρέπει σε κάθε περίπτωση να ολοκληρώνει την επεξεργασία και να τερματίζει μόνη της (ακόμη κι αν το πρόγραμμα τερματίσει την P2 μέσω του σήματος SIGKILL).

Ανάγνωση/Εγγραφή

- Το πρόγραμμα `p4diff` διαβάζει δεδομένα χρησιμοποιώντας αποκλειστικά λειτουργίες του συστήματος (όχι της βιβλιοθήκης `stdio`). Τα δεδομένα, τόσο από το αρχείο όσο και από τη συμβατική είσοδο, πρέπει πρώτα να διαβάζονται σε buffers των 64 bytes έως ότου αυτοί γεμίσουν (εφόσον συνεχίζουν να υπάρχουν δεδομένα) και στη συνέχεια να ξεκινά η σύγκριση των buffers byte προς byte.
- Στο `hw4.c`, για την ανάγνωση των περιεχομένων των αρχείων `<program>.err` και `<program>.args` και για την εκτύπωση των τελικών βαθμών στη συμβατική έξοδο μπορούν να χρησιμοποιηθούν είτε λειτουργίες της βιβλιοθήκης `stdio` είτε κλήσεις συστήματος. Σε όλες τις άλλες περιπτώσεις ανάγνωσης/εγγραφής από/προς αρχεία ή συμβατική είσοδο/έξοδο πρέπει να χρησιμοποιηθούν αποκλειστικά κλήσεις συστήματος.

Δημιουργία ξυπνητηριού

- Το ξυπνητήρι πρέπει να δημιουργείται από την κυρίως διεργασία. Εάν δεν ισχύει αυτό, θα υπάρξει μείωση βαθμού (**έως 10 πόντοι**) μετά την οριστική υποβολή της εργασίας.
- Δεν επιτρέπεται η χρήση της κλήσης συστήματος `signal`.

Πακετάρισμα και αποστολή εργασίας

Προθεσμία τελικής υποβολής: **Κυριακή 31/5/2026, 21:00**

Κατασκευάστε ένα αρχείο zip με όνομα `hw4.zip` που περιέχει τα αρχεία `hw4.c`, `p4diff.c`, `Makefile` και υποβάλετε το στο `autolab`.

Ο ελάχιστος βαθμός που πρέπει να πετύχετε είναι 70. **Λάβετε υπόψη πως πιθανώς να γίνει μείωση του βαθμού σας μετά την οριστική υποβολή (δείτε απαιτήσεις).**

Το μέγιστο πλήθος υποβολών στο `autolab` χωρίς βαθμολογική ποινή είναι 25. Κάθε υποβολή πλέον των 25 έχει βαθμολογική ποινή 1 μονάδα.

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 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.