Προγραμματισμός Συστήματος – **Εργασία 1η** – εαρινό εξάμηνο 2023

Μέγιστη βαθμολογία: 1.5 / 10

Προθεσμία υποβολής : Δευτέρα 24/4/2023 23:59

Προγραμματιστική εργασία bash shell scripting

Έκδοση 1.0

Εισαγωγή

Στην μη απωλεστική συμπίεση (lossless compression) διατηρείται η ακεραιότητα των δεδομένων. Τα αρχικά δεδομένα και τα δεδομένα μετά τη συμπίεση και την αποσυμπίεση είναι ακριβώς τα ίδια, και κατά τη διαδικασία δε χάνεται κανένα μέρος των δεδομένων. Τα «πλεονάζοντα» δεδομένα κωδικοποιούνται κατά τη συμπίεση και αποκωδικοποιούνται κατά την αποσυμπίεση.

Δύο βασικές τεχνικές στη κατηγορία της μη απωλεστικής συμπίεσης που χρησιμοποιούνται από γνωστά προγράμματα συμπίεσης αρχείων είναι:

• Η Κωδικοποίηση χαρακτήρων σταθερού μήκους με κώδικα μεταβλητού μήκους (Huffman).

• Η Κωδικοποίηση χαρακτήρων μεταβλητού μήκους με κώδικα σταθερού μήκους (LZW).

Κάθε χαρακτήρας ενός αρχείου κειμένου κωδικοποιημένου με ASCII (ASCII αρχείο) χρησιμοποιεί 7bits για την κωδικοποίηση του. Το πρόβλημα της συμπίεσης ενός αρχείου στο οποίο δίνουν λύση οι παραπάνω γνωστές τεχνικές μη απωλεστικής συμπίεσης εκφράζεται ως εξής:

Μας δίνεται αρχείο με n διαφορετικούς χαρακτήρες. Θέλουμε να αντιστοιχίσουμε τον κάθε χαρακτήρα ή συμβολοσειρά σε ένα δυαδικό κωδικό, έτσι ώστε να ελαχιστοποιηθεί ο συνολικός αριθμός των bits όλου του αρχείου.

Η βασική ιδέα των τεχνικών είναι η εύρεση ενός νέου «αλφαβήτου» όπου το μέγεθος του κάθε κωδικού (αριθμός bits) που θα επιλεχθεί να απεικονίσει κάποιο χαρακτήρα ή συμβολοσειρά θα είναι αντιστρόφως ανάλογο με την συχνότητα εμφάνισης του χαρακτήρα ή της συμβολοσειράς εντός τους αρχείου. Με αυτόν τον τρόπο και εκμεταλλευόμενοι τον πλεονασμό που παρουσιάζουν τα αρχεία κειμένου επιτυγχάνεται μια πιο αποτελεσματική κωδικοποίηση που οδηγεί σε μείωση του μεγέθους. Η αποσυμπίεση του αρχείου που προκύπτει απαιτεί την γνώση και χρήση του νέου «αλφαβήτου» προκειμένου να λάβουμε πάλι το αρχικό αρχείο.

Αναζητείστε περισσότερη πληροφορία σχετικά με τη μη απωλεστική συμπίεση (lossless compression) καθώς και τη κωδικοποίηση βασισμένη σε εξεύρεση ενός άλλου «αλφαβήτου» στο διαδίκτυο.

Ενδεικτικά:

https://en.wikipedia.org/wiki/Data_compression

https://en.wikipedia.org/wiki/Lossless_compression

https://en.wikipedia.org/wiki/Morse_code

Αντικείμενο εργασίας

Δανειζόμενοι τη προσέγγιση εύρεσης ενός νέου «αλφαβήτου» στοχεύοντας στον «πλεονασμό» που παρουσιάζουν τα αρχεία κειμένου σε κάποια φυσική γλώσσα, στοχεύουμε στην εκ νέου κωδικοποίηση ενός αρχείου κειμένου στην Αγγλική έτσι ώστε να μειωθεί το μέγεθος του στον δίσκο και να ασφαλίσουμε το περιεχόμενο του από όποιον δεν έχει τους νέους κώδικες - «αλφάβητο».

Δεν στοχεύουμε στη βέλτιστη συμπίεση αλλά σε κάποια συμπίεση αποφεύγοντας την αλγοριθμική πολυπλοκότητα. Θα επιχειρήσουμε μια **νέα κωδικοποίηση σε επίπεδο λέξης**.

Α. **80% της βαθμολογίας**

Σας δίνεται ένα αρχείο κειμένου με το οποίο θα πειραματιστείτε (- διαθέσιμο στον φάκελο του eclass ΑΝΑΘΕΣΕΙΣ ΕΡΓΑΣΙΩΝ) . **Τρία διαφορετικά bash shell scripts πρέπει να γραφτούν**:

1. myencode <text_file> <coded_file> <alphabet_file>

Βρείτε το σύνολο των διαφορετικών λέξεων μέσα στο text_file, την συχνότητα τους και το αθροιστικό κόστος σε χαρακτήρες μέσα στο αρχείο. Για παράδειγμα μια λέξη με 3 χαρακτήρες μπορεί να παρουσιάζεται 100 φορές μέσα στο κείμενο, ενώ μια λέξη με 8 χαρακτήρες να παρουσιάζεται 50 φορές μέσα στο κείμενο. Είναι σαφές ότι το αθροιστικό κόστος της πρώτης είναι 300 χαρακτήρες, ενώ της δεύτερης 400 χαρακτήρες. Οι λέξεις σε μια πρόταση χωρίζονται μεταξύ τους με τον κενό χαρακτήρα και οι γραμμές με τον χαρακτήρα νέας γραμμής.

Χτίστε ένα νέο «αλφάβητο» και συμβολίστε μοναδικά με λιγότερους χαρακτήρες εκείνες τις λέξεις που έχουν το μεγαλύτερο αθροιστικό κόστος. Παράγετε και σώστε το νέο κωδικοποιημένο αρχείο στο coded_file. Σώστε το νέο αλφάβητο σε ξεχωριστό αρχείο alphabet_file.

(*τρέξτε ascii –d να δείτε τους ascii χαρακτήρες, την αρίθμηση τους στο δεκαδικό και ποιοι είναι εκτυπώσιμοι – printable και μπορείτε να χρησιμοποιήσετε κατά το χτίσιμο του νέου «αλφαβήτου», τρέξτε echo 65 | awk '{printf("%c",$1)}' για να τυπώσετε τον χαρακτήρα που αντιστοιχεί στον δεκαδικό αριθμό 65 και είναι το A*)

Σκοπός μας είναι και η ασφάλεια. Ο χρήστης μπορεί να σώζει σε διαφορετικό αποθηκευτικό μέσο το αλφάβητο καθιστώντας την αποκωδικοποίηση χωρίς αυτό, το λιγότερο, δύσκολη από κάποιον τρίτο.

Συγκρίνετε το μέγεθος του παραγόμενου με αυτό του αρχικού. **Τι ποσοστό συμπίεσης πετύχατε;** Πως συγκρίνεται με την συμπίεση που θα πετυχαίνατε χρησιμοποιώντας το gzip;

2. mydecode <coded_file> <alphabet_file> <text_file2>

Αποκωδικοποιήστε το coded_file χρησιμοποιώντας το αλφάβητο που βρίσκεται στο alphabet_file και σώστε το στο text_file2. Το text_file2 πρέπει να είναι ακριβώς ίδιο με το αρχικό text_file (χρησιμοποιήστε την εντολή cmp για να το επιβεβαιώσετε).

3. myencat <coded_file> <alphabet_file>

Αποκωδικοποιήστε το coded_file χρησιμοποιώντας το αλφάβητο που βρίσκεται στο alphabet_file και παρουσιάστε το στη οθόνη.

Β. **20% της βαθμολογίας**

Προτείνετε και υλοποιήστε ένα αποτελεσματικό απλό τρόπο για να προστατέψετε ακόμα περισσότερο το κωδικοποιημένο αρχείο αλλά και το αλφάβητο, δυσκολεύοντας την αποκωδικοποίηση του από τρίτους. Για παράδειγμα θα μπορούσατε να «ανακατέψετε» την θέση των χαρακτήρων του κωδικοποιημένου αρχείου σύμφωνα με κάποιο αλγόριθμο που να παίρνει σαν όρισμα ένα αριθμό που θα γνωρίζετε μόνο εσείς και θα προσδιορίζει τον τρόπο του ανακατέματος (διαφορετικός αριθμός πρέπει να οδηγεί σε διαφορετικό τρόπο ανακατέματος ενώ πρέπει να είναι δυνατή η αντιστροφή της διαδικασίας). Το ίδιο θα μπορούσατε να κάνετε και για το αλφάβητο. Σε αυτή τη περίπτωση θα πρέπει να φτιάξετε ένα script με όνομα shuffle <#code> <input_file> <shuffled_file> και ένα script unshuffle <#code> <shuffled_file> <output_file>. Έτσι μετά την κωδικοποίηση πρέπει να «ανακατέψετε» το κωδικοποιημένο αρχείο ή/και το αρχείο με το αλφάβητο, ενώ πριν την αποκωδικοποίηση να επαναφέρετε το κωδικοποιημένο αρχείο ή/και το αρχείο με το αλφάβητο. Η πιο απλή περίπτωση θα ήταν να κάνετε το ανακάτεμα σε επίπεδο κωδικοποιημένης λέξης. Όπως είναι προφανές το script θα μπορεί να χρησιμοποιηθεί ως εργαλείο για να προστατέψετε οποιοδήποτε αρχείο κειμένου άσχετα αν πιο πριν του έχετε κάνει και κωδικοποίηση όπως ζητείται στο πρώτο σκέλος της εργασίας.

Ο τρόπος που προτείνεται είναι ενδεικτικός. Δεκτοί όλοι οι τρόποι που θα αποδείξετε ότι είναι αποτελεσματικοί αρκεί να τους αναπτύξετε εσείς.

*Σε όλα τα scripts που θα παράγετε ακολουθήστε όλες τις ενδεδειγμένες τακτικές για να παράγετε κώδικα που θα είναι κατάλληλα δομημένος και **με σχόλια** που θα τον συνοδεύουν, ενώ το παραγόμενο πρέπει να είναι φιλικό στον χρήστη παρέχοντας τις **κατάλληλες οδηγίες στο command line**.*

Παραδοτέα

- Μια σύντομη και περιεκτική εξήγηση για τις επιλογές που έχετε κάνει στο σχεδιασμό του προγράμματός σας καθώς και αναφορά στις συγκρίσεις που σας ζητούνται.

- Ο κώδικας που θα υποβάλετε θα πρέπει να είναι δικός σας. Δεν επιτρέπεται η χρήση κώδικα που δεν έχει γραφεί από εσάς **(**αυτό συμπεριλαμβάνει και κώδικα από το Διαδίκτυο).

- Όλη η δουλειά σας (πηγαίος κώδικας και README) σε ένα tar.gz file με ονομασία OnomaEponymoProject1.tar.gz.

- Κρατήστε ένα backup .tar της άσκησής σας όπως ακριβώς αυτή υποβλήθηκε.

- Η σωστή υποβολή ενός σωστού tar.gz που περιέχει τον κώδικα της άσκησής σας και ότι αρχεία χρειάζονται είναι αποκλειστικά ευθύνη σας.

Διαδικαστικά

- Βεβαιωθείτε πως ακολουθείτε καλές πρακτικές software engineering κατά την υλοποίηση της άσκησης. Η οργάνωση, η αναγνωσιμότητα και η ύπαρξη σχολίων στον κώδικα αποτελούν κομμάτι της βαθμολογίας σας.

- Η υποβολή θα γίνει μέσω του eclass. Θα ειδοποιηθείτε σχετικά όταν ξεκινήσει η υποβολή των εργασιών. Η υποβολή θα είναι δυνατή μέχρι την ημέρα και ώρα που αναφέρεται στη προθεσμία υποβολής.

Άλλες σημαντικές παρατηρήσεις

- Οι εργασίες είναι ατομικές.

- Αν και αναμένεται να συζητήσετε με φίλους και συνεργάτες το πώς θα επιχειρήσετε να δώσετε λύση στο πρόβλημα, αντιγραφή κώδικα (οποιασδήποτε μορφής) είναι κάτι που δεν επιτρέπεται και ελέγχεται με αυτοματοποιημένο και μη τρόπο.

- Θα πρέπει να λάβετε τα κατάλληλα μέτρα ώστε να είναι προστατευμένος ο κώδικάς σας και να μην αποθηκεύεται κάπου που να έχει πρόσβαση άλλος χρήστης.

- Ενδέχεται να κληθείτε να παρουσιάσετε και να εξεταστείτε στις εργασίες σας αναλύοντας τη προσέγγιση σας και τις προγραμματιστικές σας επιλογές.

- Οι ασκήσεις προγραμματισμού πρέπει να δοθούν μέχρι την καταληκτική ημερομηνία.