

Ομάδα: Ιωάννης Αμελαδιώτης(sdi2300005) και Ευστάθιος Κωνσταντίνου(sdi2300099)

Στην τελική υποβολή του προγράμματος , τα .c αρχεία βρίσκονται εντός του φακέλου src ενώ τα .h στον φάκελο include. Το αρχείο myz.c το οποίο περιέχει την main συνάρτηση του προγράμματος βρίσκεται εντός του φακέλου programs.

Πως να εκτελέσετε το πρόγραμμα

Το πρόγραμμα χρησιμοποιεί Makefile για να επιτυχει Seperate Compilation. Για να γίνει build το πρόγραμμα αρκεί να εκτελεστεί η εντολή make στο root του project. Τα object files γίνονται build σε ξεχωριστό directory (build), το οποίο διατηρεί την ιεραρχία των αρχείων, για να μην κανουν clutter τα υπόλοιπα αρχεία. Έπειτα απο την εν λόγω εντολή το πρόγραμμα είναι έτοιμο να εκτελεστεί απο τη γραμμή εντολών με τον τρόπο που περιγράφεται στην εκφώνηση: ./myz {-c|-a|-x|-d|-m|-q|-p} [-j] <archive-file> [list-of-files/dirs...]

Υπάρχουν επίσης διάφορα targets για τη διευκόλυνση εκτέλεσης των διαφόρων λειτουργιών.

Αρκεί ο χρήστης να βάλει στις πρώτες 2 γραμμές του Makefile το archive file και τη λιστα με τα αρχεία και directories και να τρεξει ένα από τα:

```
make run_create
make run_append
make run_extract
make run_delete
make run_metadata
make run_query
make run_print
make run_create_compressed
make run_append_compressed
```

μπορεί επίσης στο τέλος να τρέξει make clean για να διαγράψει τα object files και το εκτελεσιμο.

Δομή .myz αρχείου:

Το .myz αρχείο αποθηκεύεται με παρόμοιο τρόπο με αυτόν που παρουσιαζόταν στην εκφώνηση της εργασίας. Το πρόγραμμα έχει χωριστεί σε 3 segments. Το πρώτο περιέχει το header (MYZ magic number, File Size, Metadata_offset) , το δεύτερο περιέχει τα δεδομένα των αρχείων και το τρίτο όλα τα μεταδεδομένα σχετικά με τις ιδιότητες και την ιεραρχία των αρχείων.

Πολιτικές προγράμματος:

- Αν βάλω ένα αρχείο με filepath dir1/dir2/file , το myz θα αποθηκεύσει στην ιεραρχία όλα τα αρχεία που υπάρχουν στο filepath που δίδεται (δηλαδή τα dir1,dir2,dir3).
- Αν βάλω ένα filepath που καταλήγει σε directory , μέσω της create ή της append , θα προστεθούν όλα τα entries του directory αυτού.
- Αν βάλω ένα directory που υπάρχει ήδη μέσα στο myz (στην ίδια ιεραρχία αρχείων με το filepath) τότε το directory δεν θα προστεθεί στο .myz αρχείο και ας έχει διαφορετικά entries απο αυτό που είναι ήδη μέσα σε αυτό.
- Αν τυπώσω το .myz αρχείο με το -m flag , τότε θα τυπωθούν όλα τα directories/αρχεία μαζί με τα permissions και τα uid, gid που τους αντιστοιχούν.
- Αν βάλω ένα filepath στο .myz μαζί με το -j flag τότε το πρόγραμμα θα συμπίεσει τα αρχεία , θα τα εισαγάγει στο .myz αρχείο και θα τα αποσυμπίεσει.
- Τα hardlinks αντιμετωπίζονται ως κανονικά αρχεία, δηλαδή όταν αποσυμπίεστεί το αρχείο θα έχει το δικό του inode.

ADTVector:

Για τις ανάγκες του προγράμματος έχει υλοποιηθεί ένα ADT Vector το οποίο υλοποιεί τις βασικές λειτουργίες ενός Vector.

Entry:

Το Entry αποτελεί ένα struct το οποίο αποθηκεύει κάποιο entry του directory στο οποίο το MyzNode περιέχεται. Συγκεκριμένα αποθηκεύει το όνομα του αρχείου που περιέχει το directory καθώς και το index του στο MyzNode list.

Info:

Το Info αποτελεί ένα struct στο οποίο αποθηκεύονται κάποια properties (mode, user_id, group_id) του αρχείου στο οποίο αναφέρεται.

MyzNode:

Το MyzNode αποτελεί ένα struct για την αποθήκευση του εκάστοτε αρχείου/directory που αποθηκεύεται στο .myz αρχείο,

περιέχει το όνομα του αρχείου, μια μεταβλητή που υποδεικνύει αν το αρχείο είναι compressed, ένα struct με πληροφορίες για το αρχείο (Info struct), το offset απο το οποίο ξεκινάνε τα data του αρχείου στο .myz, το μέγεθος του αρχείου , τα δεδομένα του αρχείου και ένα vector με τα entries (Vector απο Entry structs) του στην περίπτωση που είναι directory.

Metadata:

Το Metadata αποτελεί ένα struct το οποίο περιέχει ένα Vector απο MyzNodes , δηλαδή περιέχει όλα τα δεδομένα των αρχείων και της ιεραρχίας τους.

Header:

Το header αποτελεί ένα struct που περιέχει πληροφορίες για το magic number του myz αρχείου, το μέγεθος του αρχείου και το offset των metadata. (οι συναρτήσεις του header βρίσκονται στο src/header.c και είναι αρκετά απλές)

Myz:

Το Myz αποτελεί ένα struct που περιέχει τα structs του header και των metadata του αρχείου.

Metadata.c

read_Data:

Η συνάρτηση read_Data υλοποιεί το διάβασμα ενός directory ή αρχείου μέσω του filepath που δίδεται απο τον χρήστη. Η συνάρτηση σπάει το path με βάση τον χαρακτήρα '/' και δημιουργεί τους αντίστοιχους καταλόγους (με την σωστή ιεραρχία). Αν συναντήσει αρχείο αντί για directory (που θα είναι τελευταίο στο filepath αλλιώς δεν θα ήταν valid), η συνάρτηση θα εισαγάγει το αρχείο και τα δεδομένα του στα metadata και θα επιστρέψει. Αλλιώς αν το τελευταίο είναι directory καλεί την read_data η οποία αρχικοποιεί αναδρομικά το αρχείο.

read_data:

Η read_data εισάγει ένα directory αναδρομικά στα metadata , διατηρώντας την ιεραρχία του. Η read_data διαβάζει ένα directory μέχρι να εξαντλήσει τα entries του, για κάθε entry εξετάζει τον τύπο του ,το διαβάζει τα ανάλογα δεδομένα και το εισάγει στα metadata, επίσης εισάγει στο MyzNode του directory που περιέχεται το αρχείο (του οποίου το index έχει δοθεί σαν όρισμα) ένα entry που αντιστοιχεί στο αρχείο. Αν κάποιο entry του είναι directory , τότε η συνάρτηση αφού κάνει τις κατάλληλες εισαγωγές, καλεί ξανά τον εαυτό της , αλλάζοντας το file_path και δίνοντας το index του directory το οποίο θα εξεταστεί στην νέα κλήση.

write_data:

Η συνάρτηση write_data χρησιμοποιείται για την εξαγωγή των αρχείων απο myz. Αρχικά προσπελαίνει σειριακά το vector των metadata και αν το αρχείο δεν έχει εξαχθεί ήδη τότε, αν είναι directory το δημιουργεί και στη συνέχεια καλεί την αναδρομική write_rec. Αλλιώς αν είναι κανονικό αρχείο το δημιουργεί, το αποσυμπίεζει αν είναι συμπιεσμένο, και τυπώνει τα δεδομένα του. Τέλος καλείται η write_links , η οποία γράφει όλα τα symbolic links που υπάρχουν στο αρχείο.

write_rec:

Η συνάρτηση write_rec εξάγει αναδρομικά ένα directory που της δίνεται. Αρχικά εξετάζει τα entries του directory που της δόθηκε, αν δεν έχει εξαχθεί το εκάστοτε entry τότε αν είναι directory το φτιάχνει και στην συνέχεια καλεί τον εαυτό της με το filepath που αντιστοιχεί στο directory για το οποίο καλέστηκε καθώς και με το νέο entry του directory. Αν είναι κανονικό αρχείο τότε το εξάγει και , αν είναι συμπιεσμένο, το αποσυμπίεζει.

write_links και write_l:

Οι write_links και write_l είναι παρόμοιες με την write_data και την write_rec αλλά εξάγουν μόνο τα soft links. Ο λόγος που καλούνται μετά την write_data, είναι ώστε να υπάρχουν όλα τα πιθανά references του κάθε soft links σε κάποιο αρχείο που υπάρχει στο .myz αρχείο.

metadata_find_node:

Η συνάρτηση metadata_find_node δέχεται ένα filepath και στην συνέχεια προσπελαίνει τα metadata ώστε να δει αν το filepath (δηλαδή όλα τα αρχεία με την ίδια ιεραρχία στο filepath) υπάρχει στο .myz αρχείο. Αν υπάρχει ακριβώς το ίδιο αρχείο τότε η συνάρτηση επιστρέφει το MyzNode που του αντιστοιχεί και η μεταβλητή exists παίρνει την τιμή true. Αλλιώς αν δεν υπάρχει γυρνάει το κοντινότερο node στην ιεραρχία που του δόθηκε , το οποίο υπάρχει στο .myz αρχείο.

metadata_find_parent:

Ίδια λειτουργία με την metadata_find_node αλλά βρίσκει αν υπάρχει το προ-τελευταίο αρχείο του path.

print και print_rec:

Ίδιος αλγόριθμος με την write και την write_rec αντίστοιχα αλλά τυπώνει τα ονόματα των αρχείων αντί να τα εξάγει.

Myz.c

create_myz_file:

Η συνάρτηση create_myz_file δημιουργεί το .myz αρχείο που έφτιαξε ο χρήστης μέσω του προγράμματος myz. Αρχικά , αφήνει χώρο για το header του αρχείου και στην συνέχεια γράφει όλα τα δεδομένα των αρχείων ενώ παράλληλα κρατάει το offset των

δεδομένων του εκάστοτε αρχείου. Στην συνέχεια, αφού κρατήσει το offset των μεταδεδομένων στο header, αποθηκεύει όλα τα μεταδεδομένα των αρχείων. Τέλος γράφει το ολοκληρωμένο header στην αρχή του αρχείου.

read_myz_file:

Η συνάρτηση read_myz_file διαβάζει ομοίως το υπάρχων .myz αρχείο που έχει δώσει ο χρήστης και στην συνέχεια διαβάζει με αντίστοιχο τρόπο , δηλαδή με παρόμοια σειρά που έγραψε τα δεδομένα, το αρχείο και αποθηκεύει το header, τα δεδομένα και τα μεταδεδομένα των αρχείων.

write_after_append:

Η write_after_append γράφει ΜΟΝΟ τα δεδομένα των νέων αρχείων που προστεθήκανε μετά την λειτουργία -a (δηλαδή τα γράφει στο αρχείο αρχίζοντας απτο , πλέον παλιό, metadata_offset) και όλα τα μεταδεδομένα των αρχείων που υπάρχουν στο .myz .

append:

Η συνάρτηση append είναι υπεύθυνη για την προσθήκη νέων αρχείων στο .myz αρχείο που δέχεται σαν όρισμα. Αρχικά ελέγχει αν υπάρχει κάποιο μέρος του path που τις δίδεται εντός του .myz. Αν υπάρχει όλο το path τότε επιστρέφει χωρίς να κάνει κάτι, αλλιώς αν δεν υπάρχει κανένα μέρος του path τότε απλώς προσθέτει όλη την ιεραρχία αρχείων που της δόθηκε στα μεταδεδομένα και διαβάζει τα δεδομένα των αρχείων , αν είναι directory, μέσω της read_data. Αν υπάρχει μέρος του path τότε φτιάχνει το μέρος που λείπει με την κατάλληλη ιεραρχία και αν είναι directory το εισάγει με την read_data, αλλιώς απλά το κάνει insert στα metadata.

delete_non_dir:

Η συνάρτηση delete_non_dir καλείται για να διαγράψει ένα κενό directory, ένα symbolic link ή ένα κανονικό αρχείο. Η delete_non_dir διαγράφει το αρχείο απτα metadata και απτα entries του directory που το περιέχει (δηλαδή του parent που δέχεται σαν όρισμα) , και φτιάχνει τα indexes των entries (μιας και η vector_remove_at κάνει swap το στοιχείο στο index που του δίνουμε με το τελευταίο και αφαιρεί το τελευταίο).

delete_dir:

Η delete_dir καλείται αναδρομικά και διαγράφει όλα τα subdirectories, και τα αρχεία που περιέχει το directory που δέχεται σαν όρισμα.

write_after_delete:

Η write_after_delete γράφει ξανά το myz αρχείο αλλάζοντας το header και γράφοντας , για εξοικονόμηση χώρου, μόνο τα δεδομένα των αρχείων των οποίων το data_offset είναι μεγαλύτερο απτο μικρότερο data_offset κάποιου αρχείου που διαγράφηκε και στη συνέχεια ξαναγράφοντας όλα τα μεταδεδομένα.

Το αρχείο args.c περιέχει μια συνάρτηση που κάνει parse τα arguments και το αρχείο sys_utils.c περιέχει macros και συναρτήσεις για ασφαλή εκτέλεση των διάφορων syscall και δεσμεύσεων μνήμης