

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
Τμήμα Πληροφορικής και Τηλεπικοινωνιών
4η Εργασία - Τμήμα: Περιττών Αριθμών Μητρώου
K22: Λειτουργικά Συστήματα – Χειμερινό Εξάμηνο '16
Ημερομηνία Ανακοίνωσης: 8 Ιανουαρίου
Ημερομηνία Υποβολής: 19 Φεβρουαρίου και Ώρα 23:59

Εισαγωγή στην Εργασία:

Ο στόχος αυτής της εργασίας είναι να δημιουργήσετε ένα πρόγραμμα συστήματος, με την ονομασία `mydiz`, που επιπεδοποιεί λογικές ιεραρχίες καταλόγων και αρχείων στο LINUX. Το αρχείο που δημιουργείται έχει postfix `.di` και μπορεί να διαθέτει τα περιεχόμενα των αρχείων στην ιεραρχία σε συμπυκνωμένη μορφή με την βοήθεια του προγράμματος `gzip` όπως επίσης και όλες τις απαραίτητες πληροφορίες σχετικά με την δομή της λογικής ιεραρχίας στο LINUX. Το πρόγραμμα σας θα πρέπει να δέχεται σαν παραμέτρους εισόδου σημαίες και ονόματα/λίστες αρχείων/καταλόγων που θα πρέπει να αποθηκευτούν στο υπό διαμόρφωση `.di` αρχείο.

Το πρόγραμμα σας θα πρέπει με επιτυχία να «ανακτά» είτε ολόκληρη ή μέρος της ιεραρχίας καταλόγων/αρχείων από το επιπεδοποιημένο `.di` αρχείο. Θα πρέπει να τονιστεί ότι το `mydiz` θα πρέπει να διαχειρίζεται όχι μόνο καταλόγους και αρχεία αλλά και συνδέσμους αρχείων. Το ζητούμενο πρόγραμμα θα πρέπει να δουλεύει με παρεμφερή τρόπο όπως εκείνος των υπηρεσιών `zip`, `tar`, `rar` κλπ.

Διεπαφή του `mydiz` και Βασικές Λειτουργίες:

Οι σημαίες της γραμμής κλήσης καθορίζουν την λειτουργικότητα του `mydiz`. Σε περίπτωση που δοθούν λάθος παράμετροι, το πρόγραμμα σας θα πρέπει να δίνει κάποια ένδειξη λάθους και να τερματίζει. Παρακάτω, περιγράφουμε την διεπαφή του `mydiz` και την αναμενόμενη λειτουργία των επιμέρους σημαιών:

```
./mydiz {-c|-a|-x|-m|-d|-p|-j} <archive-file> <list-of-files/dirs>
```

Οι σημαίες της γραμμής εντολής δουλεύουν ως εξής:

- c Αρχειοθέτηση των οντοτήτων συστήματος αρχείου που παρέχονται από την `<list-of-files/dirs>` στο αρχείο `<archive-file>`. Κατάλογοι και αρχεία που είναι φωλιασμένα στην λίστα `<list-of-files/dirs>` αποθηκεύονται με αναδρομικό τρόπο στο `<archive-file>`.
- a Προσθήκη των εγγράφων και καταλόγων στην `<list-of-files/dirs>` στον πιθανά ήδη υπάρχον αρχείο `<archive-file>`. Αν υπάρχουν φωλιασμένοι κατάλογοι στο `<list-of-files/dirs>` περιλαμβάνονται αναδρομικά μαζί με όλο το περιεχόμενο τους στο `<archive-file>`.
- x Εξαγωγή όλων των εγγράφων/καταλόγων που βρίσκονται αποθηκευμένοι στο `<archive-file>`.
- j Τα έγγραφα πριν την αρχειοθέτηση στο `<archive-file>` θα πρέπει να συμπιεστούν με `gzip`.
- d Διαγραφή των οντοτήτων που αναφέρονται στη λίστα `<list-of-files/dirs>` από το υπάρχον αρχείο επιπεδοποίησης `<archive-file>`. Η υλοποίηση της σημαίας αυτής είναι προαιρετική.
- m Εκτύπωση μεταπληροφοριών (owner, group, access rights) για όλες τις οντότητες που είναι αποθηκευμένες στο `<archive-file>`.
- q Επερώτηση για το αν η όχι τα στοιχεία της λίστας `<list-of-files/dirs>` βρίσκονται στο αρχείο `<archive-file>`. Για κάθε στοιχείο της `<list-of-files/dirs>` θα παρέχεται μια θετική ή μια αρνητική απάντηση.
- p Εκτύπωση της ιεραρχίας των καταλόγων και των αρχείων που βρίσκονται στο `<archive-file>` με τρόπο που να είναι εύκολα αντιληπτός/αναγνώσιμος.

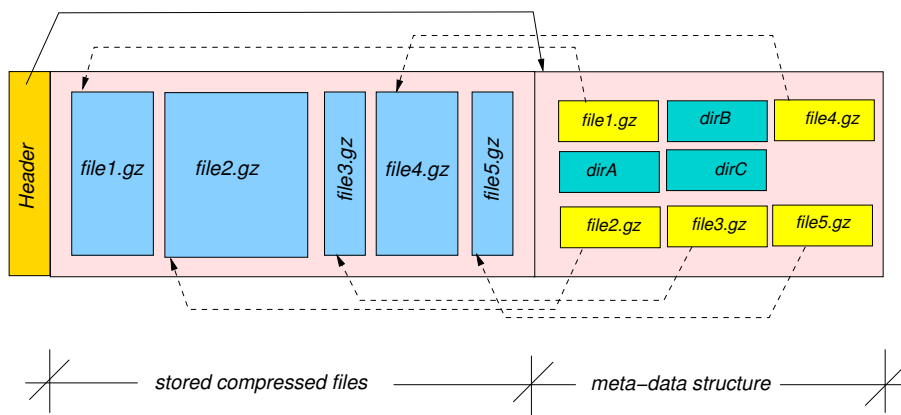
Μερικές παρατηρήσεις:

- Το πρόγραμμα σας δεν είναι απαραίτητο να χρησιμοποιεί την στάνταρντ είσοδο/έξοδο με οποιοδήποτε άλλο τρόπο πλην αυτού που περιγράφεται παραπάνω.
- Η διαδικασία ανάκτησης περιλαμβάνει μόνο αρχεία ή/και καταλόγους που ο χρήστης παραθέτει στην γραμμή εκτέλεσης του προγράμματος με την χρήση της παραμέτρου/λίστας <list-of-files/dirs>. Αν δεν δίνεται καμία τέτοια λίστα, με την επίκληση της σημαίας -x, όλες οι αποθηκευμένες οντότητες «ανα-δημιουργούνται» στο κατάλογο που διενεργείται η ανάκτηση.
- Η σημαία -j μπορεί να χρησιμοποιηθεί μόνο μαζί με την σημαία -c ή την -a.
- Όταν γίνεται ανάκληση αρχείων, η τελική μορφή των αρχείων στην (λογική) ιεραρχία θα είναι χωρίς συμπίεση (ανεξάρτητα αν είχε υπάρξει συμπίεση περιεχομένου στα σχετικά αρχεία).

Ανάπτυξη του Προγράμματος σας:

Η αποθήκευσή της λίστας των αρχείων/καταλόγων στο .di αρχείο, θα πρέπει να γίνει με ένα τρόπο που βοηθά την ανάκτηση της πληροφορίας και να είναι συμπαγής. Εκτός από τα απλά αρχεία, το .di αρχείο θα πρέπει να αποθηκεύει με ορθό τρόπο την λογική ιεράρχησή καταλόγων αλλά και τις μεταπληροφορίες για όλες τις αποθηκευμένες οντότητες συστήματος αρχείου όπως κάτοχο (owner), ομάδα (group), δικαιώματα προσπέλασης (access rights), timestamps δημιουργίας/ αλλαγής/ προσπέλασης, κλπ.

Το Σχήμα 1 προτείνει μια γενική διάταξη για τα επιπεδοποιημένα αρχεία που θα παράγει το mydiz. Μπορείτε να επεκτείνετε η και να αλλάξετε την παραπάνω δομή όπως εσείς θεωρείτε καλύτερα ώστε να επιτύχετε συμπαγή οργάνωση και γρήγορη λειτουργία (ακόμα και όταν μεγάλες σε όγκο και μορφή ιεραρχίες καταλόγων πρέπει να επιπεδοποιηθούν). Θα πρέπει ωστόσο να εξηγήσετε και να δικαιολογήσετε τις σχεδιαστικές επιλογές σας.



Σχήμα 1: Παράδειγμα οργάνωσης της δομής του αρχείου .di

Η προτεινόμενη οργάνωση του .di αρχείου αποτελείται από τρία τμήματα: μια επικεφαλίδα, μια σειρά από τα (συμπεσμένα) αρχεία και μια δομή που να αναπαριστά με ακρίβεια την λογική ιεραρχία και τις μεταπληροφορίες των οντοτήτων που αποθηκεύονται. Όλη η παραπάνω οργάνωση αποθηκεύεται στο δίσκο. Πιο συγκεκριμένα: 1) Η επικεφαλίδα μπορεί να βαστάει οποιαδήποτε στοιχεία που περιγράφουν την φύση του .di αρχείου όπως π.χ. μέγεθος σε bytes αλλά και ένα δείκτη (δηλ. αριθμό byte) πέρα από τον οποίο αποθηκεύεται η αναπαράσταση της δομής της ιεραρχίας (δηλ. μεταδεδομένα). Το μέγεθος της επικεφαλίδας παραμένει σταθερό και είναι γνωστό εξ' αρχής. 2) Τα (συμπεσμένα) αρχεία αποθηκεύονται το ένα μετά το

άλλο στο δεύτερο τμήμα του .di αρχείου. 3) Το τρίτο και τελευταίο κομμάτι του .di αρχείου είναι ουσιαστικά το «λεξικό» το οποίο διαθέτει όλες της πληροφορίες ιεράρχησης της <list-of-files/dirs> και τις μεταπληροφορίες των επιμέρους οντοτήτων (αρχεία, κατάλογοι, δεσμοί).

Θα πρέπει να τονιστεί ότι η <list-of-files/dirs> μπορεί να περιέχει υποκαταλόγους πολλαπλών επιπέδων των οποίων το βάθος δεν είναι γνωστό εκ των προτέρων. Το mydiz θα πρέπει αναδρομικά να καλύπτει όλη την ιεραρχία καταλόγων όπως επίσης και φωλιασμένων αρχείων και δεσμών. Τέλος, όταν το πρόγραμμα σας επικαλείται με σημαία -j θα πρέπει για κάθε αρχείο με περιεχόμενο (ASCII ή binary) να γίνει κλήση της gzip. Αυτό μπορεί να γίνεται με ταυτόχρονη επίκληση fork()/exec*() κατά τα γνωστά. Στην διάρκεια της ανάκτησης η αντίστροφη διαδικασία θα πρέπει να ακολουθηθεί.

Το πρόγραμμα σας θα μπορεί να διαβάσει πληροφορίες για i-nodes χρησιμοποιώντας την κλήση stat (κάνετε man -S 2 stat). Επίσης θα φανούν χρήσιμες οι παρακάτω κλήσεις lstat, readdir, opendir, readlink, fchdir, chdir, getcwd, open, close, read, write, κλπ.

Διαδικαστικά:

Το πρόγραμμά σας θα πρέπει να γραφτεί σε C (ή C++ αν θέλετε αλλά χωρίς την χρήση STL/Templates) και να τρέχει στις μηχανές Linux του τμήματος. Παρακολουθείτε την ιστοσελίδα του μαθήματος για επιπρόσθετες ανακοινώσεις στο URL <http://www.di.uoa.gr/~ad/k22>.

- Υπεύθυνοι για την άσκηση αυτή (ερωτήσεις, αξιολόγηση, βαθμολόγηση κλπ.) είναι οι Θάνος Γιαννόπουλος (thanosgn+AT-di) και Ιωσήφ Αγγελίδης (iosang+AT-di).
- Μέσα από την σελίδα <https://piazza.com/uoa.gr/fall2016/k22/home>, θα μπορείτε να κάνετε ερωτήσεις και να δείτε απαντήσεις ή/και διευκρινήσεις που δίνονται σχετικά με την άσκηση (όπως έγινε με τις τρεις προηγούμενες ασκήσεις).

Τι πρέπει να Παραδοθεί:

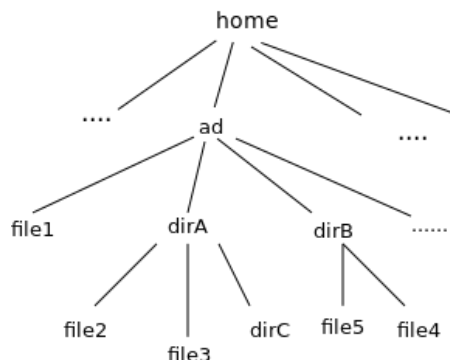
1. Μια σύντομη και περιεκτική εξήγηση για τις επιλογές που έχετε κάνει στο σχεδιασμό του προγράμματος σας (1-2 σελίδες σε ASCII κειμένου είναι αρκετές).
2. Ένα Makefile (που να μπορεί να χρησιμοποιηθεί για να γίνει αυτόματα το compile του προγράμματος σας).
3. Ένα tar-file με όλη σας την δουλειά σε έναν κατάλογο που πιθανώς να φέρει το όνομά σας και θα περιέχει όλη σας την δουλειά δηλ. source files, header files, output files (αν υπάρχουν) και οτιδήποτε άλλο χρειάζεται.

Άλλες Σημαντικές Παρατηρήσεις:

1. Η εργασία είναι είτε ατομική ή μπορεί να γίνει σε ομάδες των δύο ατόμων. Αν επιλέξετε έναν/μία συνάδελφο, καλό είναι να κάνετε μια καλή επιλογή ώστε να έχετε τον ίδιο βαθμό ενδιαφέροντος επιτυχούς ολοκλήρωσής της άσκησης.
2. Το πρόγραμμα σας θα πρέπει να τρέχει σε LINUX του Τμήματος αλλιώς δεν θα βαθμολογηθεί.
3. Αν και αναμένεται να συζητήσετε με φίλους και συνεργάτες το πως θα επιχειρήσετε να δώσετε λύση στο πρόβλημα, αντιγραφή κώδικα (οποιαδήποτε μορφής) είναι κάτι που δεν επιτρέπεται και δεν πρέπει να γίνει. Οποιοσδήποτε βρεθεί αναμειγμένος σε αντιγραφή κώδικά απλά παίρνει μηδέν στο μάθημα. Το ίδιο ισχύει και για τις ομάδες. Όσοι εμπλέκονται ανεξάρτητα από το ποιος έδωσε/πήρε κλπ. μηδενίζονται.

Appendix: Μια μέθοδος Οργάνωσης των Μεταδεδομένων σε .di Αρχεία

Το Σχήμα 2 παρουσιάζει μια λογική ιεραρχία κομμάτι της οποίας αποθηκεύεται σε .di αρχείο. Το εν λόγω τμήμα αποτελείται από το αρχείο file1 και τους καταλόγους dirA και dirB. Εμφανώς υπάρχουν και άλλα τμήματα της ιεραρχίας /home/ad/ καθώς επίσης και του /home/ που παραμένουν εκτός αποθήκευσης.



Σχήμα 2: Παράδειγμα τμήματος ιεραρχίας που αποθηκεύεται σε αρχείο .di

Το Σχήμα 3 δίνει μια ισοδύναμη εικόνα της παραπάνω ιεραρχίας που προσφέρεται για κατά-γραμμή εκτύπωση με τρόπο DFS. Στο σχήμα αυτό το βάθος κενών χαρακτήρων από την αρχή της γραμμής είναι μεταβλητό και μπορεί να ορίζεται από την/τον χρήστη.

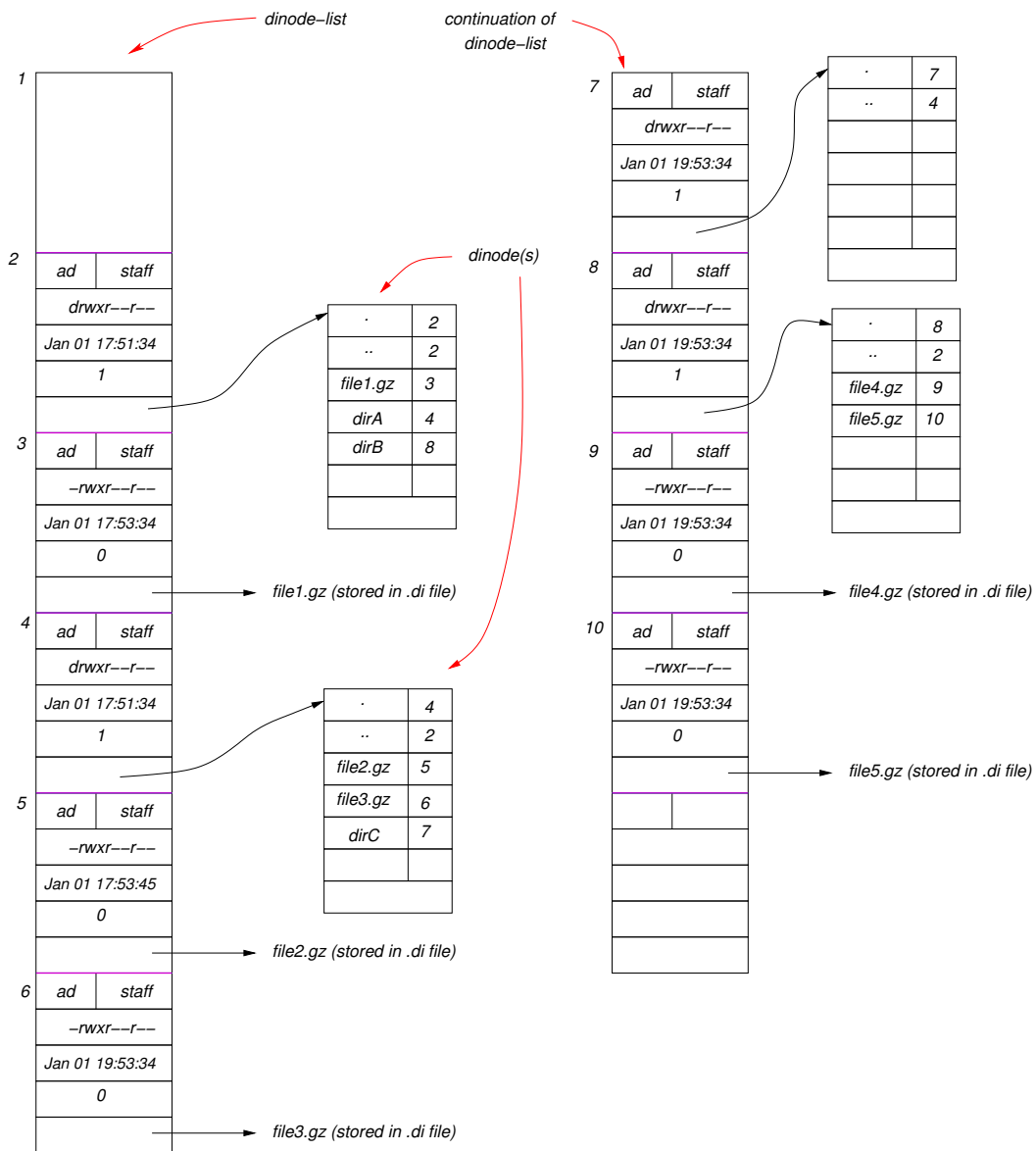
```
home
....
ad
file1
dirA
file2
file3
dirC
dirB
file4
file5
....
....
```

Σχήμα 3: Παράδειγμα εκτύπωσης κατά γραμμή της ιεραρχίας προς αποθήκευση σε αρχείο .di

Αν από το κατάλογο βάσης /home/ad/ εκτελέσουμε την εντολή:

```
./mydiz -c stored.di file1 dirA dirB
```

τότε το Σχήμα 4 παρουσιάζει μια προτεινόμενη οργάνωση των μεταδεδομένων για τα αρχεία και καταλόγους που αποθηκεύονται στο υπό σύνθεσή stored.di αρχείο. Η προτεινόμενη οργάνωση βασίζεται σε μια απλουστευμένη μορφή των i-node-list και i-nodes του LINUX, και με παρόμοιο τρόπο αποτελείται από μία dinodes-list και προσαρτώμενα dinodes. Και οι δύο αυτές οντότητες μπορούν να δημιουργούνται με βάση μπλοκς τα οποία έχουν σταθερό μέγεθος (ας πούμε 215 ή 1024 bytes).



Σχήμα 4: Παράδειγμα οργάνωσης της μεταδεδομένων του αρχείου .di με την βοήθεια μιας dinodes-list και σχετικών dinodes.