

ΑΝΑΦΟΡΑ ΕΡΓΑΣΙΑΣ

ΕΞΑΜΗΝΟΥ

Υλοποίηση ενός διερμηνέα γραμμής εντολών – *shell*

Interactive λειτουργία

- Αρχικά ο χρήστης δίνει το `input`, όποιο και αν είναι αυτό. Το πρώτο πράγμα που κάνει ο αλγόριθμος είναι να επεξεργαστεί τα κενά και να τα αφαιρέσει. Φτιάχνω έναν πίνακα `counttabs[]`, ο οποίος για κάθε χαρακτήρα που έχει ο `input[]`, αποθηκεύει το 0, ενώ για κάθε κενό ή tab το 1, στην αντίστοιχη θέση. Μετρώ πρώτα τα κενά μετά τον τελευταίο χαρακτήρα του `input[]` και μετά τα αρχικά κενά, πριν απο τον πρώτο και κάθε φορά που βρίσκω 0 κάνω ολίσθηση και τους δύο πίνακες προς τα αριστερά, έτσι ώστε στο τέλος της λούπας να έχω τον πρώτο χαρακτήρα της εντολής στην πρώτη θέση. Έπειτα μετρώ τα ενδιάμεσα κενά και ξεκινάω να αντιγράψω τον `input[]` στον `formed[]`, ο οποίος είναι ο τελικός πίνακας, που θα έχει μέσα την σωστή εντολή με τον σωστό αριθμό κενών, π.χ `[ls -l; cat file] -> [ls -l;cat file]`. Αφού γίνουν αυτά, θέτω τα υπόλοιπα-περίσσεια στοιχεία του `formed[]` ίσα με 0 και ξεκινάω να μετρώ τα ερωτηματικά.
- Φτιάχνω και πάλι έναν νέο πίνακα `formed1[]`, στον οποίο αποθηκεύω κάθε φορά την προς εκτέλεση εντολή **μέχρι το τάδε**

ερωτηματικό και ολισθαίνω τον *formed[]* αριστερά, ώστε μετά την εκτέλεση της πρώτης εντολής, να μου δώσει την δεύτερη κ.ο.κ. Αν οι πρώτοι τέσσερις χαρακτήρες της επόμενης εντολής είναι 'q','u','l','t', τότε το πρόγραμμα τερματίζει.

- Ο αλγόριθμος συνεχίζει βρίσκοντας τον αριθμό των pipes και παίρνει τις εξής περιπτώσεις :

A) Αν δεν υπάρχουν pipes, τότε παίρνει υπο-περιπτώσεις -> **i)** αν οι πρώτοι δύο χαρακτήρες του *formed1[]* είναι 'c', 'd' τότε καλώ την ***create_cd()***, η οποία αφού χωρίσει σε δύο μέρη το string με την *strtok()*, χρησιμοποιεί την συνάρτηση βιβλιοθήκης *chdir()*, για την αλλαγή φακέλου. **ii)** αν δεν είναι 'c', 'd' τότε μετρώ τις παραμέτρους που έχω (ξεκινώντας από *parameters+1*), δεσμεύω *parameters+1* μνήμη για έναν πίνακα *given()* από strings και καλώ την ***parse_and_ex(formed1, givenpar)***; Αυτή χωρίζει και πάλι με την *strtok()* τον *formed1[]* σε strings, τα οποία αποθηκεύονται στον *givenpar[]*, χωρίς κενά. Τέλος καλείται η *ex(givenpar)*, που δημιουργεί μία θυγατρική διεργασία η οποία εκτελεί την εντολή μέσα στον πίνακα *givenpar[]* με την συνάρτηση βιβλιοθήκης *execvp()*.

B) Αν υπάρχουν pipes, τότε καλείται η *ex_ripe()*. Φτιάχνω λοιπόν ένα πίνακα μεγέθους *numpipes = 2*ripenums*, αφού κάθε pipe έχει ένα read και ένα write κελί. Αφού περάσω τον πίνακα *formed1[]* σαν όρισμα, ακολουθώ πάλι την ίδια τακτική με την δεύτερη βούλα. Αυτή την φορά αποθηκεύω την προς εκτέλεση εντολή **μέχρι την τάδε σωλήνωση** σε έναν καινούριο πίνακα *c_formed[]* και ολισθαίνω τον *formed[]* προς τα αριστερά, ώστε μετά την εκτέλεση της πρώτης εντολής πριν την τάδε σωλήνωση να διαβάσει από αυτήν η δεύτερη εντολή κ.ο.κ. Ακολουθώ τα ίδια βήματα με την δεύτερη υποπερίπτωση του A), όπου αποθηκεύω στον *give[]* τις παραμέτρους και αρχίζω το forking. Κάθε παιδί, με

την χρήση της *dup2()*, ανακατευθύνει το *stdout* και το *stdin* στο πρώτο και στο δεύτερο κελί του κάθε *pipe* αντίστοιχα. Αν η εντολή προς εκτέλεση είναι η πρώτη, το παιδί θα κάνει ανακατεύθυνση μόνο του *stdout*, γιατί προφανώς δεν θέλουμε να πάρει κάποιο *input* και έτσι όταν αυτή εκτελεστεί, αντί να εμφανιστεί στο κέλυφος, το αποτέλεσμα γράφεται στο δεύτερο κελί του πρώτου *pipe*. Αφού κλείσουν όλες οι σωληνώσεις για αυτό το παιδί, η μητρική διεργασία αναλαμβάνει την εκ νέου ολίσθηση του *formed[]*, ώστε πλέον να αποθηκευθεί η εντολή μετά το *pipe*. Έπειτα δημιουργείται νέο παιδί, το οποίο θα ακολουθήσει την ίδια διαδικασία, ώστε τώρα το αποτέλεσμα της νέας εντολής να ανακατευθυνθεί στο δεύτερο κελί του δεύτερου *pipe* και ταυτόχρονα παίρνει *input* από την πρώτη εντολή μέσω του πρώτου κελιού του πρώτου *pipe* για να εκτελέσει την δεύτερη εντολή. Η διαδικασία αυτή γίνεται μέχρι να φτάσει και η τελευταία εντολή, όπου το τελευταίο παιδί δεν θα κάνει ανακατεύθυνση του *stdout* και το τελικό αποτέλεσμα εμφανίζεται στο *shell*.

Batch λειτουργία

Η *batch* λειτουργία βασίζεται στην *interactive*, το μόνο που αλλάζει είναι ότι ζητείται από τον χρήστη να δώσει το όνομα του *batchfile*, το περιεχόμενο του οποίου αποθηκεύεται στον *input[]* και αυτός με την σειρά του υπόκειται στην ίδια επεξεργασία που αναφέρθηκε παραπάνω. Αν δεν βρεθεί η εντολή *quit* στο τέλος του *batchfile*, το κέλυφος τερματίζει από μόνο του.

Κώδικας