Αναφορά 1ης Εργασίας- Λειτουργικά Συστήματα

Μελάκη Δέσποινα 3140116
Στριμμένος Ιωάννης 3140197
Τρικάλης Χρήστος 3140205

Όλα τα κομμάτια του φλοιού μας υποστηρίζονται από κάποιες βασικές μεθόδους, όπου οι δηλώσεις των μεθόδων βρίσκονται στο αρχείο **p3140116-p3140197-p3140205-mysh.h** και οι υλοποιήσεις στο αρχείο **p3140116-p3140197-p3140205-mysh-common.c** . Οι μέθοδοι είναι οι εξής:

- Readline(): Διαβάζει μια γραμμή από το πληκτρολόγιο του χρήστη η οποία έχει μέγιστο μέγεθος τους 255 χαρακτήρες.
- Concat(str1, str2): Παίρνει 2 strings και τα ενώνει φτιάχνοντας ένα καινούργιο.

p3140116-p3140197-p3140205-mysh1

Το mysh1 χρησιμεύει ώστε να διαβάζει εντολές του χρήστη χωρίς παραμέτρους. Για να το κάνουμε αυτό χρησιμοποιήσαμε την εντολή execve(source, argv, env) η οποία παίρνει σαν παραμέτρους την εντολή στην οποία έχουμε προσθέσει το “/bin/”, τον πίνακα δύο θέσεων που περιέχει την εντολή στην πρώτη θέση και NULL στην δεύτερη, καθώς και το current directory.

p3140116-p3140197-p3140205-mysh2

Το mysh2 χρησιμεύει ώστε να διαβάζει εντολές του χρήστη μες παραμέτρους. Για να το πετύχουμε αυτό αφού κάνουμε tokenize την string στα κενά τοποθετούμε όλα τα token σε πίνακα και τα δίνουμε ως ορίσματα μαζί με την πρώτη θέση του πίνακα στην execvp(args[0], args)

p3140116-p3140197-p3140205-mysh3

Το mysh3 χρησιμεύει να διαβάζει εντολές του χρήστη που μπορεί να περιλαμβάνουν και ανακατευθύνσεις εισόδου και εξόδου. Για να το κάνει αυτό διαβάζει την εντολή του χρήστη και μετράει πόσους χαρακτήρες τύπου «<» υπάρχουν και πόσες «>». Αν υπάρχει κάποιος από τους δύο χαρακτήρες, κάνει κατάλληλα tokenize στην εντολή, χρησιμοποιεί την fd για να ανοίξει ή να δημιουργήσει το αρχείο και ύστερα χρησιμοποιώντας την dup2 κάνει override τις default θέσεις τους συστήματος για διάβασμα ή και γράψιμο και εκτελεί την εντολή. Τέλος, πριν τελειώσει επαναφέρει τα default slots μνήμης για input/output στο σύστημα.

p3140116-p3140197-p3140205-mysh4

Το mysh4 χρησιμεύει ώστε να μπορεί το σύστημα να δέχεται εντολές με μια ή καμιά pipe «|». Διαβάζει την εντολή και κοιτάει αν έχει «|». Αν υπάρχει δημιουργούμε έναν πίνακα int

2 θέσεων και καλούμε την συνάρτηση pipe με όρισμα αυτόν τον πίνακα ώστε να δημιουργηθεί και να αποθηκευτεί στις δύο θέσεις του πίνακα το input και output για το συγκεκριμένο pipe. Κάνουμε override την θέση του γραψίματος και με fork() αναθέτουμε στο παιδί διεργασία να τρέξει την πρώτη εντολή του pipe. Γυρίζοντας στον πατέρα κάνουμε close το write και κάνουμε override την αρχική θέση διαβάσματος του συστήματος με αυτή του pipe. Έτσι ο πατέρας καλώντας την executeUserCommand() διαβάζει το αποτέλεσμα της προηγούμενης εντολής και κάνοντας fork() εκτελεί την νέα. Μόλις τελειώσει κάνει close() και στην θέση διαβάσματος του pipe και επαναφέρει τα default slots μνήμης για input/output στο σύστημα.

p3140116-p3140197-p3140205-mysh5

Το mysh5 χρησιμεύει ώστε να μπορεί να δέχεται εντολές με περισσότερες από 1 pipes. Διαβάζει την εντολή και κοιτάει πόσα pipes έχει. Κάνοντας ένα for loop κάνει tokenize την εντολή στους χαρακτήρες «|» και τοποθετεί όλες τις εντολές σε πίνακα. Αρχικοποιεί επίσης τις θέσεις για input και output σε αυτές του συστήματος. Ξεκινάει λοιπόν ένα for loop για όλα τα pipes και δημιουργεί έναν πίνακα int 2 θέσεων τον οποίο γεμίζει με την pipe() δίνοντας του έτσι τις θέσεις για input και output του pipe. Κάνοντας override στην αρχή μόνο την θέση γραψίματος με αυτή του νέου pipe και εκτελεί την εντολή. Μόλις τελειώσει κάνει close() την θέση γραψίματος και την προηγούμενη θέση διαβάσματος και κάνει override αυτή του διαβάσματος με αυτή του pipe. Η διαδικασία επαναλαμβάνεται για όλα τα pipes μόνο που ελέγχουμε αν βρίσκεται στην αρχή να μην κάνει close() την default θέση του συστήματος για διάβασμα και επίσης στην τελευταία εντολή δεν δημιουργούμε νέο pipe αλλά αποκαθιστούμε την default θέση του συστήματος για γράψιμο.