

ΕΡΓΑΣΙΑ ΣΤΑ ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

ΟΝΟΜΑ: ΚΩΝΣΤΑΝΤΙΝΟΣ

ΕΠΙΘΕΤΟ: ΛΕΤΡΟΣ

ΣΧΟΛΗ: ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ: ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ. ΚΑΙ ΜΗΧ.
ΥΠΟΛΟΓΙΣΤΩΝ

ΑΕΜ: 8851

ΕΞΑΜΗΝΟ: 7^ο

ΕΤΟΣ: 2018

Ζητείται να υλοποιήσουμε έναν διερμηνέα γραμμής εντολών - κέλυφος (command line interpreter - shell) το οποίο περιλαμβάνει δύο λειτουργίες, την interactive λειτουργία και την batch. Στη main γίνεται έλεγχος για το ποια λειτουργία πρέπει να ενεργοποιηθεί ανάλογα με το πλήθος των ορισμάτων κατά την εκτέλεση του shell. Συγκεκριμένα, αν δεν δοθεί κανένα όρισμα, το shell εισέρχεται στην interactive λειτουργία. Αν δοθεί ένα όρισμα και είναι έγκυρο(batch-text αρχείο) τότε ενεργοποιείται η batch λειτουργία. Σε διαφορετική περίπτωση το shell επιστρέφει μήνυμα λάθους. Σε περίπτωση που δεν συμβαίνει το τελευταίο, καλείται η συνάρτηση loop με το αντίστοιχο όρισμα και συνεχίζει η αντίστοιχη λειτουργία του προγράμματος.

Η λειτουργία του προγράμματος βασίζεται σε έναν ατέρμων βρόχο (συνάρτηση loop) στον οποίο τυπώνεται το prompt letros_8851> και στη συνέχεια ζητείται από τον χρήστη να δώσει τις επιθυμητές εντολές.

Η διακοπή του ατέρμονος βρόχου πραγματοποιείται με την εντολή quit, η οποία υλοποιήθηκε ειδικά. Σημειώνεται ότι οι εντολές που υλοποιήθηκαν ειδικά είναι η cd για την περιήγηση στους φακέλους του συστήματος (συνάρτηση my_cd) και η quit για τον τερματισμό του προγράμματος.

Η κάθε γραμμή που δίνεται από το χρήστη μπορεί να αποτελείται από μία ή περισσότερες εντολές και η εκτέλεσή της γίνεται ως εξής. Αρχικά διαχωρίζεται με βάση τους χαρακτήρες '&', ';' σε δύο πίνακες. Ο πρώτος περιέχει σε κάθε θέση ένα string με την κάθε εντολή και ο δεύτερος περιέχει τους χαρακτήρες με βάση τους οποίους έγινε ο διαχωρισμός (συνάρτηση split). Στη συνέχεια γίνεται έλεγχος της γραμμής για τυχόν σφάλματα της μορφής '&' ή '&&&' αντί για '&&' (αντίστοιχα για όλους τους ειδικούς χαρακτήρες που αναφέρονται – συνάρτηση multichk) και στην περίπτωση που όλα είναι σωστά, γίνεται εκ νέου έλεγχος για τυχόν ύπαρξη των χαρακτήρων '|', '>', '<' για την κάθε εντολή. Αν βρίσκονται εντός κάποιας εντολής τότε αυτή εκτελείται μέσω συνάρτησης η

οποία υλοποιήθηκε ειδικά για αυτόν το σκοπό, όπως αναφέρεται στη συνέχεια. Οι υπόλοιπες εντολές, που δεν περιέχουν κάποιον από τους παραπάνω χαρακτήρες, εκτελούνται με ξεχωριστή συνάρτηση.

Συνεπώς, υπάρχουν τα εξής ενδεχόμενα. Όταν εισάγεται '&&' εκτελείται η προηγούμενη και κατόπιν η επομένη, από το σύμβολο, εντολή (συνάρτηση EXECfunc). Αν η πρώτη εντολή δεν είναι έγκυρη τότε η δεύτερη δεν εκτελείται. Όμοια, ανταποκρίνεται το shell, και στην περίπτωση του ';' με τη διαφορά ότι και οι δύο εντολές εκτελούνται ανεξαρτήτως εμφάνισης σφάλματος. Στην περίπτωση εισαγωγής '>' ή '>>' καλείται η συνάρτηση greaterfunc, στην οποία γίνεται ανακατεύθυνση της εξόδου στο αρχείο που προσδιορίζεται από το δεύτερο, στη σειρά, μέρος της εντολής (με χρήση της dup2). Η διαφορά τους είναι ότι ο χαρακτήρας '>' εγγράφει σε ένα νέο αρχείο (write) ενώ ο '>>' εγγράφει την είσοδο (πρώτο μέρος της εντολής) στο τέλος του αρχείου (append). Παρόμοια αποκρίνεται το shell στην περίπτωση εισαγωγής '<' (συνάρτηση lessfunc) με τη διαφορά ότι γίνεται ανακατεύθυνση της εισόδου στο αρχείο που προσδιορίζεται από το πρώτο μέρος της εντολής. Τέλος, στην περίπτωση εισαγωγής '|' καλείται η συνάρτηση διασωλήνωσης, (συνάρτηση PIPEfunc) η οποία δημιουργεί δύο child διεργασίες. Η πρώτη συνδέει την έξοδο της πρώτης εντολής με την πατρική ενώ η δεύτερη την είσοδο της δεύτερης εντολής με την πατρική, κι έτσι υλοποιείται η διασωλήνωση.

Αναφορικά με την εκτέλεση της κάθε εντολής, η διαδικασία που ακολουθείται είναι η εξής. Δημιουργείται (με τη χρήση της εντολής fork) μία διεργασία-παιδί (συνάρτηση child) η οποία είναι υπεύθυνη για την εκτέλεση της κάθε εντολής (με χρήση της execvp). Συγκεκριμένα, η κάθε εντολή διακρίνεται ξανά με βάση το χαρακτήρα 'κενό' (space) σε λέξεις, ώστε να διαχωριστεί η εντολή από τις παραμέτρους της (flags) και στη συνέχεια να εκτελεστεί με αυτές. Με το πέρας της εκτέλεσης ελέγχεται αν η διαδικασία-παιδί τερμάτισε ανεπιτυχώς ώστε να εκτυπωθεί μήνυμα σφάλματος. Σε αυτό το σημείο χρησιμοποιήθηκε η

συνάρτηση `wait` με τρόπο τέτοιο ώστε να εκτελούνται οι εντολές διαδοχικά, με τη σειρά που εκτελέστηκαν.

Όταν το shell είναι σε λειτουργία batch λαμβάνεται επιπλέον μέριμνα για την περίπτωση που δεν εμφανίζεται η εντολή `quit` σε κάποιο σημείο του εκτελούμενου αρχείου. Όταν εκτελεστούν όλες οι εντολές το πρόγραμμα τερματίζεται αυτόματα τη λειτουργία του.

Για την αποφυγή ανεπιθύμητων σφαλμάτων γίνονται διάφοροι έλεγχοι, όπως έλεγχος για την αποφυγή υπερχείλισης (buffer overflow) όταν το πλήθος των χαρακτήρων της γραμμής ξεπερνάει τους 512, έλεγχος για την ευσταθή λειτουργία σε περίπτωση κενής γραμμής ή γραμμής που περιέχει μόνο τον χαρακτήρα ‘κενό’ (space), έλεγχος σε περίπτωση μη υπαρκτού, άδειου ή μη έγκυρου αρχείου (όταν απαιτείται ανακατεύθυνση `>>><` ή όταν βρίσκεται σε batch λειτουργία), έλεγχος για την εισαγωγή ειδικού χαρακτήρα σε περίπτωση που δεν έχει νόημα (πχ. αν εισαχθεί μόνο ο χαρακτήρας ‘;’ ή μόνο ‘<<<<’ κτλ.).

Τέλος, σημειώνουμε ότι δεν έχουν υλοποιηθεί οι πολλαπλές ανακατευθύνσεις (πχ. `echo test < file1.txt < file2.txt`) και οι πολλαπλές διασωληνώσεις (πχ. `ls | wc | grep 5`).