

**Λειτουργικά Συστήματα**

**Εργασία 2018-2019**

**Δάμη Υπατία**

**8606**

**LINUX SHELL**

## Περιγραφή προβλήματος

Στην εργασία αυτή καλούμαστε να σχεδιάσουμε σε γλώσσα προγραμματισμού C ,έναν διερμηνέα γραμμής εντολών(κέλυφος).Το κέλυφος που σχεδίασα έχει τη δυνατότητα λειτουργίας σε interactive mode,όπου ο χρήστης πληκτρολογεί τις εντολές στο command line, καθώς και σε batch mode, όπου οι εντολές που θα εκτελεστούν χορηγούνται στο πρόγραμμα μέσω ενός αρχείου batchfile. Και στις δύο λειτουργίες το shell έχει δυνατότητα εκτέλεσης των εντολών γραμμής ,καθώς και ενότητες εντολών διαχωρισμένες με τον χαρακτήρα ";" ,ή τον χαρακτήρα "&&".Στην πρώτη περίπτωση το shell εκτελεί διαδοχικά τις εντολές που του δίνονται .Σε περίπτωση που κάποια εντολή δεν υπάρχει ή δεν μπορεί να εκτελεσθεί , το πρόγραμμα εμφανίζει μήνυμα λάθους στο χρήστη. Στην δεύτερη περίπτωση το shell εκτελεί διαδοχικά τις εντολές ,με την προϋπόθεση ότι η προηγούμενη εντολή εκτελέστηκε επιτυχώς. Διαφορετικά αν κάποια εντολή δεν μπορεί να εκτελεστεί εμφανίζεται μήνυμα λάθους και δεν εκτελούνται οι υπόλοιπες εντολές. Σε περίπτωση που ο χρήστης πληκτρολογήσει μία εντολή, αυτή εκτελείται και αν όχι επιτυχώς εμφανίζεται μήνυμα λάθους. Το πρόγραμμα έχει επίσης τη δυνατότητα ανακατεύθυνσης εντολών σε αρχεία ,καθώς και χειρισμού κενών εντολών , κενών ανάμεσα σε εντολές, κενής γραμμής στο αρχείο batchfile και έλειψης της εντολής quit από το batchfile. Το πρόγραμμα είναι φτιαγμένο έτσι ώστε να δέχεται ανα γραμμή εντολές διαχωρισμένες είτε με "&&" είτε με ";" ,σύμφωνα με τη λογική της εκκώνισης να ακολουθείται άλλη διαδικασία εκτέλεσης σε κάθε περίπτωση.

## Περιγραφή λειτουργίας προγράμματος

### Συνάρτηση int main(char\*argc,char\*argv[])

Στη συνάρτηση main , εξετάζεται το όρισμα argc για να επιλεγεί η λειτουργία του shell , interactive ή batch.Σε περίπτωση interactive mode, το πρόγραμμα μπαίνει σε ένα αέναο for loop, όπου αρχικά τυπώνεται στην κονσόλα το username (dami\_8606) και στη συνέχεια διαβάσει σε έναν buffer την είσοδο του χρήστη. Έπειτα ψάχνει στην είσοδο για να αναγνωρίσει τον χαρακτήρα διαχωρισμού των εντολών και τον αποθηκεύει. Με κατάλληλες συναρτήσεις αρχικά διαχωρίζονται οι εντολές που χωρίζονται με χαρακτήρες διαχωρισμού ,έπειτα η κάθε εντολή διαχωρίζεται από τα ορίσματά της και τέλος οι εντολές εκτελούνται .Το πρόγραμμα τερματίζει αν δοθεί η κατάλληλη εντολή "quit" . Στην λειτουργία batch , ακολουθείται παρόμοια διαδικασία μέσα σε αέναο for loop, με τη διαφορά ότι η είσοδος διαβάζεται από το αρχείο batchfile και οι εντολές εκτελούνται διαδοχικά μέχρι να αναγνωριστεί το τέλος του αρχείου ή η εντολή "quit" μέσα στο αρχείο .

### Συνάρτηση parse\_commands(char\*div, char\* in, char\* com[50])

Η συνάρτηση parse\_commands , διαχωρίζει τις εντολές που πληκτρολογήθηκαν από το χρήστη σε περίπτωση που αυτές διαχωρίζονται από χαρακτήρα διαχωρισμού. Δέχεται ως είσοδο το χαρακτήρα διαχωρισμού που ανιχνεύτηκε στην είσοδο, την είσοδο και έναν πίνακα όπου θα εκχωρηθούν οι εντολές μαζί με τα ορίσματά τους. Αν ο χαρακτήρας διαχωρισμού είναι " " , τότε η συνάρτηση εκχωρεί στην πρώτη θέση του πίνακα την μοναδική εντολή που εισήχθη και επιστρέφει .Διαφορετικά μέσω της συνάρτησης strtok()

διαχωρίζει τις εντολές , τις εκχωρεί στον πίνακα com και επιστρέφει τον αριθμό των εντολών που ανιχνεύθηκαν.

#### **Συνάρτηση parse\_args( int count, char\* com[50], char\* arg[20][20])**

Η συνάρτηση parse\_args, διαχωρίζει τα επιμέρους μέλη των εντολών , δηλαδή την κύρια εντολή και τα ορίσματά της και τα αποθηκεύει στον δισδιάστατο πίνακα arg. Σε κάθε γραμμή του πίνακα υπάρχει και η αντίστοιχη εντολή με τα ορίσματά της. Η συνάρτηση δέχεται ως είσοδο τον αριθμό των εντολών, τον πίνακα com και τον δισδιάστατο πίνακα συμβολοσειρών arg. Στον πίνακα com η εντολή με τα ορίσματά της διαχωρίζονται με χαρακτήρα κενού. Αρχικά η συνάρτηση ελέγχει την περίπτωση όπου έχουμε μόνο μία εντολή , χωρίς ορίσματα , και θέτει τις κατάλληλες τιμές στον πίνακα arg. Διαφορετικά με την συνάρτηση strtok(), διαχωρίζει την κάθε τιμή του πίνακα com, όπου υπάρχει κενό και αποθηκεύει τις τιμές που προκύπτουν στον πίνακα arg. Ο πίνακας arg πλέον περιέχει όλες τις εντολές προς εκτέλεση , διαχωρισμένες κατάλληλα για να εκτελεσθούν με τη χρήση της execvp().

#### **Συνάρτηση execute\_comm(char\* div, int count, char\* arg[20][20])**

Η συνάρτηση execute\_comm εκτελεί τις εντολές που περιέχονται στον πίνακα arg , με τη χρήση της συνάρτησης execvp(). Δέχεται ως είσοδο τον χαρακτήρα διαχωρισμού , τον αριθμό των εντολών και τον πίνακα arg. Με if έλεγχο εξακριβώνεται ο χαρακτήρας διαχωρισμού των εντολών και αναλόγως επιλέγεται η διαδικασία εκτέλεσης των εντολών. Αν ο χαρακτήρας είναι “;”, τότε σε ένα for loop δημιουργούνται διαδικασίες παιδιά τις αρχικής , τόσες όσες και ο αριθμός των εντολών με την χρήση της συνάρτησης fork(). Οι διαδικασίες παιδιά εκτελούν η κάθε μία και μια εντολή , αφού έχει τελειώσει η εκτέλεση της προηγούμενης . Αυτό επιτυγχάνεται με τη χρήση της συνάρτησης wait() από την διαδικασία –πατέρα, που την καλεί σε κάθε επανάληψη του for loop περιμένοντας την κάθε διεργασία παιδί να ολοκληρώσει την εκτέλεση. Εάν ο χαρακτήρας διαχωρισμού είναι “&&” , ακολουθείται παρόμοια διαδικασία με προηγούμενως με τη διαφορά ότι σε περίπτωση που δεν εκτελεστεί μια εντολή, μετά την ενημέρωση της μεταβλητής επιστροφής της συνάρτησης , η συνάρτηση επιστρέφει επι τόπου και εμποδίζεται η εκτέλεση των υπόλοιπων εντολών. Αν ο χαρακτήρας διαχωρισμού είναι το κενό , δηλαδή έχουμε μόνο μία εντολή προς εκτέλεση, τότε δημιουργείται μια διαδικασία –παιδί, η οποία εκτελείται και η συνάρτηση έπειτα επιστρέφει το αποτέλεσμα της εκτέλεσης. Επίσης, στην αρχή της συνάρτησης γίνεται ξεχωριστός έλεγχος για τυχόν εντοπισμό των χαρακτήρων “<” και “>” , έτσι ώστε να κληθεί η συνάρτηση redirection, που είναι υπεύθυνη για τις ανακατευθύνσεις.

#### **Συνάρτηση redirection(char\* args, int j, int pid, int dir)**

Η συνάρτηση redirection καλείται από τη συνάρτηση execute\_comm και διαχειρίζεται τις εντολές με ανακατεύθυνση εισόδου εξόδου σε αρχεία. Δέχεται ως είσοδο τον πίνακα args , τον δείκτη της εντολής που εκτελείται στην execute\_comm στην οποία εντοπίστηκε χαρακτήρας ανακατεύθυνσης , το αντίστοιχο pid και τον χαρακτήρα ανακατεύθυνσης. Έπειτα με τη χρήση της dup2, γίνεται η εναλλαγή του stdin ή του stdout , αναλόγως την περίπτωση , από την γραμμή εντολών στο αρχείο που πληκτρολόγησε ο χρήστης. Έπειτα με fork() και

execvp() εκτελείται η εντολή και η συνάρτηση επιστρέφει την κατάσταση τερματισμού του προγράμματος.

### **Συνάρτηση read\_file(char \* buffer, char\* argv[], int line)**

Η συνάρτηση read\_file διαβάζει από το αρχείο batchfile τις εντολές που θα εκτελεστούν. Δέχεται ως είσοδο έναν buffer όπου θα αποθηκεύονται τα δεδομένα του αρχείου, τον πίνακα argv[] της main, για να αναγνωριστεί το αρχείο προς ανάγνωση που πληκτρολογήσε ο χρήστης και η γραμμή του αρχείου από όπου σταμάτησε την ανάγνωση προηγουμένως. (Η συνάρτηση διαβάζει μια-μια τις γραμμές του αρχείου για να μην δημιουργείται σύγχυση με τους χαρακτήρες διαχωρισμού.) Αρχικά ανοίγει το αρχείο με τη χρήση της συνάρτησης fopen(). Αν η fopen() επιστρέψει 0 σημαίνει πως το αρχείο δεν υπάρχει και και το πρόγραμμα τερματίζεται. Διαφορετικά σε ένα βρόχο while, διαβάζονται ένα –ένα τα δεδομένα του αρχείου και αποθηκεύονται στον buffer. Αν έρθει το τέλος της γραμμής με εντολή break το πρόγραμμα βγαίνει από το βρόχο, το αρχείο κλείνει και η συνάρτηση επιστρέφει και εκτελούνται οι εντολές της πρώτης γραμμής του αρχείου από την main. Η συνάρτηση καλείται διαρκώς από την main μέχρι να διαβαστούν όλες οι γραμμές του αρχείου και να διαβαστεί ο χαρακτήρας τέλους του αρχείου “EOF”. Όταν γίνει αυτό ενημερώνεται η μεταβλητή που επιστρέφει η συνάρτηση καταλλήλως έτσι ώστε να τερματιστεί το πρόγραμμα από τη main.