

# PROJECT ΣΤΑ ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

Ονοματεπώνυμο: Καγιάφας Νικόλαος

ΑΕΜ: 8817

Εξάμηνο φοίτησης: 7ο

Ακαδημαϊκό Έτος: 2018-2019

Η δοσμένη εργασία είχε ως σκοπό την δημιουργία του δικού μας διερμηνέα γραμμής εντολών ή αλλιώς shell. Το shell αυτό ανταποκρίνεται σε δύο είδη λειτουργίας, την interactive και την batch. Στο interactive mode το shell εκτελεί μία αέναη αναδρομή, στην οποία τυπώνεται αρχικά το prompt, στο οποίο ο χρήστης εισάγει την εντολή που θα ήθελε να εκτελεστεί, για παράδειγμα την εντολή `ls -l`. Έχει προβλεφθεί να μην υπάρχει πρόβλημα στην περίπτωση όπου μεταξύ της εντολής και των διαφόρων ορισμάτων της υπάρχουν κενά, γεγονός που διευρύνει και διευκολύνει την λειτουργία του shell. Επίσης, θα πρέπει να αναφερθεί ότι σε μία γραμμή δίνεται στον χρήστη η δυνατότητα να εισάγει περισσότερες εντολές από μία, οι οποίες χωρίζονται μεταξύ τους είτε με “;”, είτε με “&&”. Στην πρώτη περίπτωση, εάν μία εντολή στην αλυσίδα διαδοχικών εντολών είναι λανθασμένη, τότε η εκτέλεση συνεχίζεται απρόσκοπτα, ενώ στην δεύτερη μπλοκάρεται η εκτέλεση και τερματίζεται το πρόγραμμα του shell. Αυτή είναι και μία παραδοχή που έλαβα, σύμφωνα με την οποία αν υπάρχει λανθασμένη εντολή ακολοθούμενη από “&&” τότε το πρόγραμμα τερματίζεται, αντί να δίνεται στον χρήστη η δυνατότητα να εισάγει μία νέα εντολή προς εκτέλεση. Στην batch λειτουργία, τα πράγματα διαφέρουν μόνο ως προς το γεγονός ότι αντί να έχουμε μία

αέναη αναδρομή, έχουμε τόσες επαναλήψεις όσες είναι και οι γραμμές του αρχείου εντολών. Στο τέλος του αρχείου, έχω ρυθμίσει έτσι το πρόγραμμά μου, ώστε να εκτελείται σίγουρα η εντολή quit και να γίνεται τερματισμός λειτουργίας.

Παραπάνω, ανέφερα ορισμένα γενικά στοιχεία για τον τρόπο με τον οποίο το πρόγραμμά μου εκτελείται. Αξίζει να τονιστούν τώρα κάποια πιο ειδικά, τεχνικά χαρακτηριστικά του προγράμματός μου. Για παράδειγμα, σε εντολή μίας γραμμής του αρχείου ή εντολής που εισάγει ο χρήστης, η μέγιστη χωρητικότητα χαρακτήρων έχει οριστεί να είναι ίση με 512 χαρακτήρες. Επίσης, σε μία τέτοια εντολή η πρώτη ενέργεια είναι να χωριστεί σε υποεντολές που θα πρέπει να εκτελεστούν και οι οποίες διαχωρίζονται μεταξύ τους όπως ανέφερα προηγουμένως. Κάθε μία από αυτές τις εντολές λοιπόν, τμηματοποιείται σε ορίσματα ανάλογα με τους κενούς χαρακτήρες. Ακόμη, αποθηκεύω σε έναν προσωρινό πίνακα το 1 ή το 2 αν μία συγκεκριμένη εντολή ακολουθείται από “;” ή “&&” αντίστοιχα. Αυτήν την πληροφορία θα την χρησιμοποιήσω αργότερα, προκειμένου να ελέγξω αν θα τερματιστεί το πρόγραμμα σε περίπτωση που η εντολή είναι λανθασμένη. Αφού λάβω τμηματοποιημένη την εντολή σε ορίσματα, μεταβαίνω στην διαδικασία εκτέλεσης της κάθε μίας. Η πρώτη μου ενέργεια στην παρούσα φάση είναι να ελέγξω αν έχω ενσωματωμένη στην εντολή πρόθεση ανακατεύθυνσης της εισόδου και της εξόδου της. Προσπαθώ δηλαδή να ανιχνεύσω αν υπάρχουν τα σύμβολα “<” και “>”. Αν υπάρχουν, τότε δεξιά τους θα βρίσκεται το αρχείο στο οποίο θέλουμε να γίνει η ανακατεύθυνση. Αυτό στην συνέχεια το ανοίγω με τα κατάλληλα δικαιώματα, ή το δημιουργώ αν δεν έχει δημιουργηθεί. Ρυθμίζω τότε στον κώδικά μου τα αποτελέσματα ή η είσοδος να λαμβάνονται από το συγκεκριμένο αρχείο και αποκόπτω την εντολή έτσι, ώστε να εξαφανιστούν από αυτήν οι ανακατευθύνσεις και να μεταβιβαστεί χωρίς πρόβλημα στην execvp. Αξίζει να τονιστεί ότι στην συνέχεια θα πρέπει να δημιουργήσω δύο διεργασίες, τον πατέρα και το παιδί. Το παιδί εκτελεί την κάθε εντολή και ο πατέρας περιμένει το παιδί του να τελειώσει, για να συνεχίσει η λειτουργία του προγράμματος. Με αυτόν τον τρόπο εξασφαλίζεται ότι οι διαδοχικές εντολές θα εκτελεστούν με την ίδια

σειρά με την οποία έχουν εισαχθεί. Σε αυτό το σημείο αξίζει να αναφέρω ότι όταν έχουμε λανθασμένη εντολή με “&&” να την ακολουθεί, τότε το παιδί στέλνει σήμα στην διεργασία του πατέρα να τερματιστεί και αυτό στην συνέχεια τερματίζει την λειτουργία του προγράμματος, αφού τυπώσει κατάλληλο μήνυμα. Αν η εντολή είναι λανθασμένη, αλλά ακολουθείται από “;”, τότε συνεχίζεται κανονικά η λειτουργία του προγράμματος και το μήνυμα που τυπώθηκε πριν ξανατυπώνεται. Τα πράγματα στην batchfile λειτουργία είναι ακριβώς τα ίδια.

Στην τελευταία αυτήν παράγραφο αφού αναλύθηκε η λειτουργία του προγράμματός μου, θα αναφερθώ σε κάποια λεπτά σημεία και σε κάποιες πιο σπάνιες περιπτώσεις που τυχαίνει να συναντήσουμε κατά την εκτέλεση του προγράμματος. Για παράδειγμα, στην batchfile λειτουργία αν μία γραμμή του δοσμένου αρχείου είναι κενή, τότε είναι σαν να μην συμβαίνει τίποτα. Απλά, το πρόγραμμά μου συνεχίζει στην επόμενη γραμμή αγνοώντας την κενή. Παρόμοια, στην interactive λειτουργία μία κενή γραμμή δεν αλλάζει κάτι στην λειτουργία του προγράμματος και το prompt τυπώνεται ξανά. Επιπλέον, όπως τονίστηκε και πριν οι κενοί χαρακτήρες ανάμεσα στις εντολές και τα ορίσματά τους δεν επηρεάζουν την λειτουργία του προγράμματος και οι εντολές εκτελούνται κανονικά χωρίς να συναντάται κάποιο πρόβλημα. Ακόμη, οποτεδήποτε η προσπάθειά μου να ανοίξει κάποιο αρχείο δεν ικανοποιείται τυπώνεται κατάλληλο μήνυμα λάθους. Τέτοιο μήνυμα τυπώνεται και κάθε φορά που θέλω να δεσμεύσω μνήμη αλλά δεν μπορώ. Θα πρέπει επιπροσθέτως να τονίσω ότι η περίπτωση άδειου batchfile υπάγεται στην περίπτωση άδειας γραμμής, οπότε δεν συμβαίνει τίποτα και το πρόγραμμα τερματίζεται. Τέλος, θα πρέπει να υπενθυμίσω πως δεν θέτω σαν υποχρέωση στο batchfile να τελειώνει με την εντολή quit αφού στο τέλος του εγώ στο πρόγραμμά μου το υποχρεώνω να τερματιστεί με αυτήν την εντολή.