

ΑΝΑΦΟΡΑ PROJECT

Ομάδα:

Αμπλιανίτης Κωνσταντίνος
Σεργάκης Αλέξανδρος

Προεργασία

Για την διεξαγωγή του project θεωρείται αναγκαία η εξοικείωση με τη χρήση της στοίβας, των αναδρομικών συναρτήσεων, τις συμβάσεις των καταχωρητών, το polling και τα interrupts σε Assembly MIPS.

Περιγραφή ζητούμενων

Ο σκοπός της άσκησης είναι η βαθύτερη κατανόηση και παράλληλα η εξοικείωση με την χρήση στοίβας ,αναδρομικών συναρτήσεων και τις συμβασεις των καταχωρητών στην γλώσσα προγραμματισμού Assembly. Αρχικά ζητείται η υλοποίηση του παιχνιδιού του λαβύρινθου και στην συνέχεια η επίλυση του λαβύρινθου αφορά τη δεύτερη λειτουργικότητα που πρέπει να υλοποιηθεί.

Περιγραφή της εκτέλεσης

Στο 1ο ερώτημα ζητείται η υλοποίηση σε C του αλγόριθμου διάσχισης λαβυρίνθου. Το ερώτημα υλοποιήθηκε με την χρήση διαφόρων συναρτήσεων. Πιο αναλυτικά δημιουργήθηκε η printLabyrinth η οποία ψάχνει βάσει την τοποθεσία του παίκτη και βάζει ένα 'P' στην θέση που βρίσκεται ο παίκτης. Επιπλέον, δημιουργήθηκε η makeMove η οποία βρίσκει το ιδανικό μονοπάτι για την λύση του λαβυρίνθου μέσω της δοκιμής κάθε δυνατού μονοπατιού από έναν αλγόριθμο. Στην συνέχεια δημιουργήθηκε η find path η οποία μέσω των εντολών w,a,s,d μετακινεί τον χρήστη αντίστοιχα πάνω,αριστερά,κάτω και δεξιά με την προϋπόθεση ότι στην θέση που θέλει να μετακινηθεί ο χρήστης έχει . και όχι I. Αν υπάρχει I εμφανίζει μήνυμα λάθους επιλογής. Στο τέλος αν ο χρήστης φτάσει στο @ εμφανίζει μήνυμα Winner Winner Chicken Dinner.

Στο 2ο ερώτημα ζητείται μετατροπή του προγράμματος από C σε Assembly. Το πλήκτρο επιλογής του χρήστη θα γίνεται με syscall αρχικά και στην συνέχεια με μέθοδο Polling. Το ερώτημα υλοποιήθηκε με την βοήθεια της c δηλαδή δημιουργήθηκαν οι 3 ίδιες συναρτήσεις όπως στην c με την χρήση της στοίβας και αναδρομικών συναρτήσεων. Αρχικά στο data segment δηλώθηκαν όλα τα δεδομένα μας (Strings , ο πίνακες, width και height) . Στην συνέχεια υλοποιήθηκε η printLabyrinth στην οποία μέσω εντολών la περνάει η θέση μνήμης των πινάκων σε καταχωρητές και μέσω της επεξεργασίας τους και την χρήση των εντολών sb(store byte) και lb(load byte) γίνεται η αποθήκευση των τιμών που υπάρχουν στο map array στον temp array ώστε να γίνει η εκτύπωση. Κάθε φορά που ο καταχωρητής, που είναι ο αντίστοιχος μετρητής j, φτάνει στο width ο καταχωρητής γραμμών (i) αυξάνεται και καλούμε την enter ώστε να αφήσουμε μία γραμμή κενό. Στην makeMove αρχικά παίρνουμε τις περιπτώσεις να γυρίζει 0 ως τιμή επιστροφής. Πριν από αυτό, δηλώνουμε στοίβα με σκοπό να κρατάμε κάθε φορά την διεύθυνση επιστροφής και την τιμή του PlayerPos ώστε αν γυρίζει ποτέ μηδέν η συνάρτηση να επιστρέφει σε αυτή που την κάλεσε. Έπειτα παίρνουμε την περίπτωση που επιστρέψει ένα και εκτελούμε μία μία τις συνθήκες. Στην ουσία αυτό που κάνει η makemove είναι κάθε φορά να μπαίνει μέσα με ένα όρισμα (PlayerPos) και αν ικανοποιείται μια συνθήκη μπαίνει και

προχωράει στο ίδιο μονοπάτι, αν όχι βγαίνει και δοκιμάζει τις υπόλοιπες. Στο τέλος αυτό που γίνεται είναι να γίνει το ra το ra που είχε στην μείν και ουσιαστικά η συνάρτηση να κλείσει οριστικά.

Τέλος η findPath υλοποιήθηκε με πολλά labels τα οποία χρησιμοποιήθηκαν για να προσωμοιώσουν την switch στην c η οποία περιέχει τα cases w,a,s,d. Στο case w ο χρήστης ανεβαίνει 1 θέση πάνω εφόσον υπάρχει ' το συγκεκριμένο case υλοποιήθηκε με μια beq η οποία κοιτάει αν ο χρήστης μπορεί να ανέβει μια θέση πάνω και μία beq που κοιτάζει αν ο χρήστης έφτασε στο τέλος και εκτυπώνει Winner Winner Chicken Dinner. Στο case d ο χρήστης πηγαίνει 1 θέση δεξιά εφόσον υπάρχει ' το συγκεκριμένο case υλοποιήθηκε με μια beq η οποία κοιτάει αν ο χρήστης μπορεί να ανέβει μια θέση στο τέλος και εκτυπώνει Winner Winner Chicken Dinner.

πάνω και μία beq που κοιτάζει αν ο χρήστης έφτασε στο τέλος και εκτυπώνει Winner Winner Chicken Dinner.

Στο case a ο χρήστης πηγαίνει μια θέση αριστερά εφόσον υπάρχει ' το συγκεκριμένο case υλοποιήθηκε με μια beq η οποία κοιτάει αν ο χρήστης μπορεί να ανέβει μια θέση πάνω και μία beq που κοιτάζει αν ο χρήστης έφτασε στο τέλος και εκτυπώνει Winner Winner Chicken Dinner.

Στο case s ο χρήστης κατεβαίνει 1 θέση εφόσον υπάρχει ' το συγκεκριμένο case υλοποιήθηκε με μια beq η οποία κοιτάει αν ο χρήστης μπορεί να ανέβει μια θέση πάνω και μία beq που κοιτάζει αν ο χρήστης έφτασε.

Τέλος το polling έγινε με τον εξής τρόπο. κάναμε li δύο καταχωρητές (0xffff0000, 0xffff0004) που είναι το trasmitter reciever και transmitter control αντιστοίχα. Αν ο χρήστης δεν πατήσει πλήκτρο τότε ο control έχει την τιμή 0 και μπαίνει μέσω and σε μία λούπα που συνεχίζει μέχρι να πατηθεί κάποιο πλήκτρο. Αν πατηθεί πλήκτρο τότε αυτός ο καταχωρητής γίνεται 1 η λούπα σταματάει και ο \$v0 που έχει το όρισμα απλά μεταφέρει την τιμή του σε ένα t καταχωρητή ώστε να γίνουν μετά οι απαραίτητες ρυθμίσεις.

Συμπεράσματα

Μέσω του project έγινε υλοποίηση του πρώτου ολοκληρωμένου προγράμματος σε Assembly Mips που βοήθησε στην κατανόηση και εξοικείωση της γλώσσας σε αρκετά μεγάλο βαθμό. Επήλθε εξοικείωση με τις εντολές συνθήκης, την στοίβα, τις αναδρομικές συναρτήσεις, polling και της σωστής χρήσης των καταχωρητών μέσω των συμβάσεων που υπάρχουν αλλά και τον τρόπο που χρησιμοποιούνται. Επιπλέον έγινε επίσης μία πρώτη ουσιαστική γνωριμία με το polling και τα interrupts που μας βοηθούν στην καλύτερη λειτουργία του προγράμματος. πχ μέσω polling γίνεται ακαριαία η εισαγωγή χαρακτήρα και δεν χρειάζεται να πατάμε enter η να βλέπουμε τον χαρακτήρα που πατήσαμε.