

Προγραμματιστική εργασία υλοποίησης εντολών για μικρό υπολογιστή

1. Βασικά

Θεωρήστε πως έχετε στην διάθεσή σας έναν περιορισμένων δυνατοτήτων υπολογιστή ο οποίος διαθέτει σαν μνήμη του 10 διευθύνσεις (0-9). Σε κάθε διεύθυνση μπορούν να υπάρχουν 4 δυαδικά ψηφία (αριθμοί 0 ή 1). Στην μνήμη μπορούμε να αποθηκεύσουμε το αποτέλεσμα μιας πράξης ή να ανακαλέσουμε μια τιμή από κάποια θέση της. Επιπλέον ο υπολογιστής διαθέτει και δύο καταχωρητές για αριθμητικές πράξεις με τα συμβολικά ονόματα R1 και R2. Κάθε ένας από αυτούς διαθέτει 4 δυαδικά ψηφία.

2. Συναρτήσεις

Η υλοποίηση των καταχωρητών και της μνήμης μπορεί να γίνει με την χρήση ακεραίων τιμών στο [0,1]. Για παράδειγμα ο καταχωρητής R1 μπορεί να δηλωθεί σαν `int R1[4]`.

Για την προσομοίωση των βασικών λειτουργιών του υπολογιστή να υλοποιηθούν οι παρακάτω συναρτήσεις:

1. `void boot()`. Η συνάρτηση αυτή εκκινεί τον υπολογιστή και κάνει εκκαθάριση της μνήμης και των καταχωρητών. Σε αυτήν την κατάσταση η μνήμη του υπολογιστή γεμίζει με τυχαίες τιμές. Για την παραγωγή μιας τυχαίας τιμής στο διάστημα [0,15] μπορείτε να χρησιμοποιήσετε τον παρακάτω κώδικα

```
int x= rand() % 16;if(x<0) x=-x;
```
2. `int loadR1(int address)`. Η συνάρτηση αυτή φορτώνει στον καταχωρητή R1 το περιεχόμενο της θέσης μνήμης `address`. Αν η θέση μνήμης δεν είναι στο διάστημα [0..9] η συνάρτηση θα επιστρέφει 0 αλλιώς θα επιστρέφει 1.
3. `int loadR2(int address)`. Η συνάρτηση αυτή φορτώνει στον καταχωρητή R2 το περιεχόμενο της θέσης μνήμης `address`. Αν η θέση μνήμης δεν είναι στο διάστημα [0..9] η συνάρτηση θα επιστρέφει 0 αλλιώς θα επιστρέφει 1.
4. `int storeR1(int address)`. Η συνάρτηση αυτή αποθηκεύει στην θέση μνήμης `address` το περιεχόμενο του καταχωρητή R1. Αν η θέση μνήμης δεν είναι στο διάστημα [0..9] δεν θα γίνεται καμία καταχώρηση και η συνάρτηση θα επιστρέφει 0 αλλιώς θα γίνεται η καταχώρηση και η συνάρτηση θα επιστρέφει 1.
5. `int storeR2(int address)`. Η συνάρτηση αυτή αποθηκεύει στην θέση μνήμης `address` το περιεχόμενο του καταχωρητή R2. Αν η θέση μνήμης δεν είναι στο διάστημα [0..9] δεν θα γίνεται καμία καταχώρηση και η συνάρτηση θα επιστρέφει 0 αλλιώς θα γίνεται η καταχώρηση και η συνάρτηση θα επιστρέφει 1.
6. `void addRegistersR1()`. Η συνάρτηση αυτή προσθέτει τα περιεχόμενα των δυο καταχωρητών R1 και R2 και το αποτέλεσμα μπαίνει στον καταχωρητή R1. Μπορείτε να πραγματοποιήσετε την πρόσθεση μετατρέποντας πρώτα τους αριθμούς σε δεκαδικούς και αφού τους προσθέσετε να τους μετατρέψετε πάλι σε δυαδικούς. Αν μετά την πρόσθεση

όμως υπάρχει αποτέλεσμα μεγαλύτερο του 15 (γιατί ο μέγιστος ακέραιος που μπορεί να αποθηκευτεί σε 4 δυαδικά στοιχεία είναι ο 1111 δηλαδή το 15), τότε η συνάρτηση θα αποθηκεύει στον καταχωρητή R1 την σειρά [0,0,0,0]

7. void addRegistersR2(). Η συνάρτηση αυτή προσθέτει τα περιεχόμενα των δυο καταχωρητών R1 και R2 και το αποτέλεσμα μπαίνει στον καταχωρητή R2. Και πάλι ισχύουν οι ίδιοι περιορισμοί με την προηγούμενη συνάρτηση.
8. void subRegistersR1(). Η συνάρτηση αυτή αφαιρεί τα περιεχόμενα των δυο καταχωρητών R1 και R2 και το αποτέλεσμα μπαίνει στον καταχωρητή R1. Αν μετά την αφαίρεση προκύπτει αρνητική τιμή τότε στον καταχωρητή R1 θα αποθηκεύεται η σειρά [1,1,1,1]
9. void subRegistersR2(). Η συνάρτηση αυτή αφαιρεί τα περιεχόμενα των δυο καταχωρητών R1 και R2 και το αποτέλεσμα μπαίνει στον καταχωρητή R2. Αν μετά την αφαίρεση προκύπτει αρνητική τιμή τότε στον καταχωρητή R2 θα αποθηκεύεται η σειρά [1,1,1,1]
10. void displayR1(). Η συνάρτηση αυτή εμφανίζει στην οθόνη το περιεχόμενο του καταχωρητή R1.
11. void displayR2(). Η συνάρτηση αυτή εμφανίζει στην οθόνη το περιεχόμενο του καταχωρητή R2.
12. void displayMemory(). Η συνάρτηση αυτή εμφανίζει στην οθόνη το περιεχόμενο της μνήμης.

3. Διερμηνευτής

Στην συνάρτηση main() μέσα σε έναν ατέρμονα βρόγχο θα γίνεται ανάγνωση από τον χρήστη μιας εντολής και θα εκτελείται μια συνάρτηση από τις αυτές της προηγούμενης ενότητας. Οι αποδεκτές εντολές είναι (πάντα με κεφαλαία γράμματα)

1. BOOT, εκκίνηση υπολογιστή.
2. LOAD R N, αποθήκευση στον καταχωρητή R της μνήμης N. Η τιμή του R μπορεί να είναι R1 ή R2.
3. STORE R N. Αποθήκευση στην θέση μνήμης N του καταχωρητή R (όπου R είναι είτε R1 είτε R2).
4. ADD R. Πρόσθεση των καταχωρητών και αποθήκευση στον καταχωρητή R (όπου R είναι είτε R1 είτε R2).
5. SUB R. Αφαίρεση των καταχωρητών και αποθήκευση στον καταχωρητή R (όπου R είναι είτε R1 είτε R2).
6. SHOW R. Εμφάνιση στην οθόνη του καταχωρητή R (όπου R είναι είτε R1 είτε R2).
7. MEM. Εμφάνιση στην οθόνη του περιεχομένου της μνήμης

Αν σε κάποια εντολή υπάρχει σφάλμα (πχ λάθος ορίσματα να εμφανίζεται ένα κατάλληλο μήνυμα λάθους στην οθόνη.

Υπόδειξη: Θα πρέπει να χρησιμοποιήσετε τις συναρτήσεις `strlen()`, `strcpy()` και `strcmp()` από την βιβλιοθήκη `string.h` καθώς και την βιβλιοθήκη `stdlib.h` για τυχαίες τιμές.

Παράδοση εργασίας: Το τελικό παραδοτέο πρέπει να είναι ένα μόνο αρχείο το οποίο θα είναι ο κώδικας σε c της εφαρμογής μαζί με τα απαραίτητα σχόλια. Η ημερομηνία παράδοσης ορίζεται η 10/4/2021 και θα πρέπει να την στείλετε από το πανεπιστημιακό σας email στο πανεπιστημιακό email του διδάσκοντα, itsoulos@uoi.gr Στο σώμα του μηνύματος πρέπει να υπάρχει το όνομα, το επίθετό σας και ο αριθμός μητρώου σας.