

Μικροεπεξεργαστές & Εφαρμογές

Εξαμηνιαία Εργασία Ομάδας 33
Μπούρχα Ιωάννα Α.Μ. 58019

Επιβλέπων καθηγητής: Γ. Συρακούλης

Εισαγωγή

Στην παρούσα εργασία ζητείται η υλοποίηση ενός αλγορίθμου ο οποίος θα δείχνει πόσο θα πρέπει να πληρώσει ο ιδιοκτήτης ενός αυτοκινήτου το οποίο έχει αφήσει σε χώρο στάθμευσης για συγκεκριμένο χρόνο. Ο χρόνος προσδιορίζεται από έναν χρονομετρητή και παρουσιάζεται με πέντε ψηφία στο δεκαεξαδικό σύστημα αρίθμησης.

Χρόνος παραμονής στον χώρο στάθμευσης	Τιμή
Λιγότερο από μία ώρα	0,50 €
Περισσότερο από μία ώρα και λιγότερο από δύο	0,95 €
Περισσότερο από δύο ώρες και λιγότερο από τρεις	1,70 €
Περισσότερο από τρεις ώρες και λιγότερο από τέσσερις	2,60 €
Περισσότερες από τέσσερις ώρες	3,50 €

Πίνακας 1: Τιμοκατάλογος

Σύμφωνα με την παρατήρηση της εκφώνησης, ο χρόνος εισόδου του κάθε αυτοκινήτου αποθηκεύεται στην θέση μνήμης της οποίας η διεύθυνση είναι η τιμή της πινακίδας του αυτοκινήτου.

Παραδοχές

Θεωρώ ότι οι τιμές των χρόνων που συμμετέχουν στο πρόγραμμα είναι σε δευτερόλεπτα. Δεδομένου ότι μία μέρα αποτελείται από $24 \cdot 60 \cdot 60 = 86.400$ δευτερόλεπτα για την αναπαράσταση των οποίων απαιτούνται 5 ψηφία, όπως ακριβώς δίδεται και στην εκφώνηση. Συγκεκριμένα με την χρήση πέντε ψηφίων για την αναπαράσταση του χρόνου μπορώ να μετρήσω μέχρι το $(FFFFF)_{16} = (1.048.575)_{10}$ που αντιστοιχεί στα δευτερόλεπτα 12 ημερών.

Η εμφάνιση του αντίστοιχου αντιτίμου προκύπτει από τον συνδυασμό των περιεχομένων δύο συγκεκριμένων θέσεων μνήμης οι οποίες ορίζονται από τον προγραμματιστή. Στη συγκεκριμένη εφαρμογή οι τιμές που επιλέχτηκαν είναι η $0xFFFFFD50$ όπου αποθηκεύεται το πλήθος των ευρώ και η επόμενη, δηλαδή η $0xFFFFFD54$, όπου αποθηκεύεται το πλήθος των λεπτών.

Κώδικας

```
// Ergasia Omadas 33
// Ioanna Bourcha A.M. 58019

// time is declared in seconds

#include <MKL25Z4.H>

__asm void payment(int plate_number, int now_time){

    // r0 --> plate_number      r1 --> now_time in seconds

    ldr r11, =0xFFFFFD50 // default address to save euros
                          // default address to save cents 0xFFFFFD54

    movs r6, #60
    mul  r6, r6, r6      // r6 = 3600 --> sec in 1 h
    adds r7, r6, #0     // r7 = 3600

    ldr r2, [r0]         // r2 --> entry_time_in_parking (sec)

    subs r2, r1, r2      // diff = now - entry (sec)

    cmp r2, r7           // 3600 sec == 1 h
    bhi c2
    movs r3, #0          // euros
    movs r4, #50         // cents
    str r3, [r11]
    str r4, [r11, #4]
```

```

        bx lr

c2: add r7, r7, r6          // r7 = 7200
    cmp r2, r7              // 7200 sec == 2 h
    bhi c3
    movs r3, #0              // euros
    movs r4, #95             // cents
    str r3, [r11]
    str r4, [r11, #4]
    bx lr

c3: add r7, r7, r6 // r7 = 10800
    cmp r2, r7              // 10800 sec == 3 h
    bhi c4
    movs r3, #1              // euros
    movs r4, #70             // cents
    str r3, [r11]
    str r4, [r11, #4]
    bx lr

c4: add r7, r7, r6          // r7 = 14400
    cmp r2, r7              // 14400 sec == 4 h
    bhi c5
    movs r3, #2              // euros
    movs r4, #60             // cents
    str r3, [r11]
    str r4, [r11, #412]
    bx lr

c5: movs r3, #3              // euros
    movs r4, #50             // cents
    str r3, [r11]
    str r4, [r11, #4]
    bx lr
}

__asm void store_entry(int plate_number, int entry_time){    str r1, [r0] }

int main(void)
{
    int plate_number = 0xFFFF000;
    int entry_time = 3058;          // time in seconds
    int now_time = 6089;           // time in seconds

    store_entry(plate_number, entry_time);

    payment(plate_number, now_time);
}

// Ergasia Omadas 33

```

Η εκτέλεση του προγράμματος σύμφωνα με την εκφώνηση πρέπει να γίνει σε περιβάλλον που υποστηρίζει επεξεργαστή ARM και συγκεκριμένα προτείνεται το Keli Uvision. Δυστυχώς, η εκτέλεσή του στο περιβάλλον αυτό δεν ήταν εφικτή λόγω αδυναμίας συμβατότητας του compiler. Για αυτό αποφάσισα να χρησιμοποιήσω έναν online simulator. Ύστερα από αναζήτηση διαπίστωσα ότι ο CPUlator (<https://cpulator.01xz.net/?sys=arm>) είναι ο καλύτερος για την συγκεκριμένη περίπτωση. Στην περίπτωση αυτή, τα δεδομένα που βρίσκονται στην main σε γλώσσα c θα πρέπει να γραφτούν σε assembly.

Ο αντίστοιχος κώδικας στο περιβάλλον αυτό είναι:

```

.global _start

// time in seconds

_start:
    ldr r0, =0xFFFF000 // plate_number
    ldr r1, =0xBF2      // entry_time 3058
    str r1, [r0]

```

```

ldr r1, =0x17C9          // now_time    6089

//-----
// r0 --> plate_number
// r1 --> now_time in sec
ldr r11, =0xFFFFFD50     // address to save euros
                        // address to save cents 0xFFFFFD504

movs r6, #60
mul r6, r6, r6            // r6 = 3600 --> sec in 1 h
adds r7, r6, #0          // r7 = 3600

ldr r2, [r0]             // r2 --> entry_time_in_parking (sec)

subs r2, r1, r2          // now - entry (sec)

cmp r2, r7               // 3600 sec == 1 h
bhi c2
movs r3, #0              // euros
movs r4, #50             // cents
str r3, [r11]
str r4, [r11, #4]
bx lr

c2: add r7, r7, r6 // r7 = 7200
cmp r2, r7        // 7200 sec == 2 h
bhi c3
movs r3, #0       // euros
movs r4, #95     // cents
str r3, [r11]
str r4, [r11, #4]
bx lr

c3: add r7, r7, r6 // r7 = 10800
cmp r2, r7        // 10800 sec == 3 h
bhi c4
movs r3, #1       // euros
movs r4, #70     // cents
str r3, [r11]
str r4, [r11, #4]
bx lr

c4: add r7, r7, r6 // r7 = 14400
cmp r2, r7        // 14400 sec == 4 h
bhi c5
movs r3, #2       // euros
movs r4, #60     // cents
str r3, [r11]
str r4, [r11, #4]
bx lr

c5: movs r3, #3    // euros
movs r4, #50     // cents
str r3, [r11]
str r4, [r11, #4]
bx lr

```

Τα περιεχόμενα των καταχωρητών είναι:

r0	0ffff000
r1	000017c9
r2	00000bd7
r3	00000000
r4	00000032
r5	00000000
r6	00000e10
r7	00000e10
r8	00000000
r9	00000000
r10	00000000
r11	fffffd50
r12	00000000
sp	00000000
lr	00000000
pc	00000000
cpsr	000001d3
spsr	00000000

Πίνακας 2: Περιεχόμενα καταχωρητών

fffffd50	00000000	00000032	aaaaaaaa	aaaaaaaa
----------	----------	----------	----------	----------

Πίνακας 3: Περιεχόμενα μνήμης