

6ο Εργαστήριο Αρχιτεκτονικής Η/Υ: Κρυφές μνήμες (cache)

Α. Ευθυμίου

Παραδοτέο: Δευτέρα 27 Δεκέμβρη 2020, 23:59

Ο σκοπός αυτής της άσκησης είναι η εμβάθυνση της κατανόησης της λειτουργίας των κρυφών μνημών. Θα πρέπει να έχετε μελετήσει τα μαθήματα για το υποσύστημα μνήμης που αντιστοιχούν στις ενότητες 5.1-5.3 και τμήμα του 5.5 του συγγράμματος των Patterson, Hennessy.

Αυτή η εργαστηριακή άσκηση είναι διαφορετική από τις προηγούμενες. Θα τρέξετε ένα πρόγραμμα σε assembly αρκετές φορές για να δείτε πως συμπεριφέρονται διαφορετικές οργανώσεις κρυφής μνήμης. Οι περιγραφές των οργανώσεων κρυφών μνημών και τί πρέπει να παρατηρήσετε θα είναι σε μορφή ερωτήσεων σε ένα quiz στη σελίδα του μαθήματος στο ecourse. Οι απαντήσεις θα δοθούν σε quiz στο ecourse και δεν θα χρειαστεί να ανεβάσετε αρχεία στο Github. Επομένως, σε αυτή την άσκηση θα χρειαστεί μόνο να πάρετε τον σκελετό της από το Github:

https://github.com/UoI-CSE-MYY505/starter_lab06. Δεν θα έχετε το δικό σας ξεχωριστό αποθετήριο.

1 Πίνακες 2 διαστάσεων

Το πρόγραμμα που σας δίνεται για αυτή την άσκηση (cache.asm), χρησιμοποιεί ένα πίνακα 2 διαστάσεων και αυξάνει κάθε στοιχείο του κατά 1. Επειδή οι πίνακες αποθηκεύονται στην μνήμη του υπολογιστή που έχει μία μόνο διάσταση, χρησιμοποιείται η λεγόμενη row-major διάταξη (order): πρώτα αποθηκεύονται τα στοιχεία της πρώτης γραμμής από αριστερά προς τα δεξιά, μετά της δεύτερης, κ.ο.κ. Σε αυτή την διάταξη, η διεύθυνση του $A[i][j]$ (το i είναι αριθμός γραμμής-σειράς και το j στήλης), γνωρίζοντας ότι κάθε στοιχείο του πίνακα είναι 4 bytes, ότι το πρώτο στοιχείο ($A[0][0]$) βρίσκεται στη διεύθυνση A και ότι ο αριθμός των στηλών είναι c , είναι $A + (i * c + j) * 4$. Δοκιμάστε το: για $i=j=0$, η διεύθυνση, σύμφωνα με την παραπάνω έκφραση, είναι A , για $i=0, j=1$, η διεύθυνση είναι $A + 4$, και για $i=1, j=0$, η διεύθυνση είναι $A + c * 4$.

Παρατηρήστε επίσης ότι στη δήλωση της ετικέτας data του πίνακα χρησιμοποιείται η οδηγία `.word 0 : 256` που αρχικοποιεί τη μνήμη με την τιμή 0, για 256 (=16x16) λέξεις των 32 bit. Επομένως υπάρχει χώρος για ένα πίνακα μεγέθους μέχρι και 16 x 16.

Φορτώστε το πρόγραμμα cache.asm στον MARS και παρατηρήστε τον κώδικά του. Είναι ένα απλό πρόγραμμα που διατρέχει έναν πίνακα δύο διαστάσεων **κατά στήλες** και αλλάζει κάθε λέξη του. Υπάρχουν 3 «παράμετροι» που μπορούν να διαφοροποιήσουν το πρόγραμμα:

- Ο αριθμός γραμμών-σειρών του πίνακα σε λέξεις (32 bit) - στον καταχωρητή `$a2`.
- Ο αριθμός στηλών του πίνακα σε λέξεις (32 bit) - στον καταχωρητή `$a1`. Ο αριθμός θα πρέπει να είναι δύναμη του 2 για να αποφευχθεί η πράξη του πολλαπλασιασμού για τον υπολογισμό των διευθύνσεων.
- Ο λογάριθμος με βάση 2 του αριθμού στηλών - στον καταχωρητή `$a0`. Χρησιμοποιείται για να υλοποιηθεί τον πολλαπλασιασμό, που χρειάζεται στον υπολογισμό διευθύνσεων, με ολίσθηση.

Αρχικά, το μέγεθος του πίνακα είναι 4x4, αλλά θα σας ζητηθεί να το αλλάξετε.

2 Προσομοιωτής κρυφής μνήμης του Mars

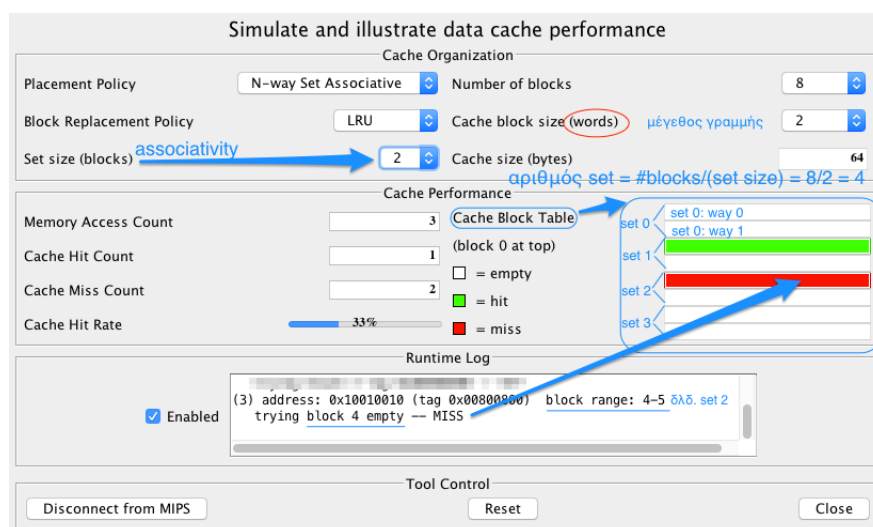
Θα χρησιμοποιήσετε τον MARS και κυρίως το “Data Cache Simulator” (βλ. Σχήμα 1) που βρίσκεται κάτω από το μενού Tools. Είναι ένας προσομοιωτής κρυφής μνήμης δεδομένων (όχι μνήμης εντολών). Μπορεί κανείς να ορίσει τον τρόπο οργάνωσης και τις βασικές παραμέτρους μιας κρυφής μνήμης και να δει πως γίνονται οι προσπελάσεις στις γραμμές της καθώς και να πάρει πληροφορίες σχετικά με τον αριθμό και το ποσοστό ευστοχιών.

Σε όλα τα εργαλεία του MARS πρέπει να πατηθεί το κουμπί “Connect to MIPS” για να αρχίσουν τα εργαλεία να «βλέπουν» τις εντολές του MIPS. Μπορεί κανείς να αλλάξει το πρόγραμμα, να φορτώσει άλλο, κλπ. χωρίς να κλείσει τα εργαλεία (ή να τα αποσυνδέσει από τον MIPS). Για να καθαριστούν οι προηγούμενες τιμές - πληροφορίες των εργαλείων, υπάρχει το κουμπί “Reset”. Αλλιώς, αν δεν αλλάξουν παράμετροι στον προσομοιωτή, συνεχίζει να συγκεντρώνει δεδομένα. Επομένως, **θα χρειαστεί να κάνετε Reset αρκετές φορές**, σε αυτό το εργαστήριο.

Οι παράμετροι οργάνωσης του cache simulator βρίσκονται στο επάνω μέρος του παραθύρου. Ακολουθούν την γνωστή, από τις διαλέξεις, ορολογία εκτός από το “Placement Policy” που στις διαλέξεις αναφέρεται ως οργάνωση της κρυφής μνήμης. Προσοχή στην παράμετρο “Set size (blocks)” που είναι ο αριθμός των block ενός set, γνωστός και ως associativity. Αν θέλετε να βρείτε τον αριθμό των set (π.χ. για να υπολογίσετε μετά πόσα bit είναι το index), θα πρέπει να διαιρέσετε το “Number of blocks” με το “Set size”. Επίσης προσέξτε ότι το μέγεθος γραμμής, “Cache block size” είναι σε λέξεις όχι bytes. Τέλος, συχνά μπερδεύει κανείς το μέγεθος γραμμής (“Cache block size”) με τον συνολικό αριθμό γραμμών της cache (“Number of blocks”).

Οι πληροφορίες στο υπόλοιπο παράθυρο του cache simulator αλλάζουν κατά την εκτέλεση του προγράμματος. Στα δεξιά υπάρχει μια αναπαράσταση της δομής της κρυφής μνήμης, που θα χρησιμοποιήσετε πολύ σε αυτή την άσκηση. Κάθε σειρά αντιστοιχεί σε μία γραμμή κρυφής μνήμης (cache block). Οι λέξεις που περιέχονται στη γραμμή δεν φαίνονται. Το χρώμα δείχνει που έγιναν ευστοχίες ή αστοχίες. Στο σχήμα έχουν επισημανθεί τα set σε μια 2-way set associative οργάνωση, γιατί, δυστυχώς, τα set δεν ξεχωρίζουν στην αναπαράσταση της cache από το εργαλείο. Αν ήταν 4-way, οι τέσσερις πρώτες σειρές/γραμμές θα αντιστοιχούσαν στο set 0, κ.ο.κ. Αν η οργάνωση ήταν direct mapped, κάθε γραμμή θα ήταν και ξεχωριστό set, ενώ αν ήταν fully-associative, θα υπήρχε ένα μόνο set και επομένως όλες οι γραμμές θα ήταν τα ways του set αυτού. Στο αριστερό μέρος παρουσιάζονται τα αποτελέσματα επίδοσης της κρυφής μνήμης (αριθμοί ευστοχιών - αστοχιών και το ποσοστό ευστοχίας). Προσοχή το τελευταίο παρουσιάζεται στρογγυλοποιημένο σε ακέραιες μονάδες. Στο κάτω μέρος του παραθύρου υπάρχει το “Runtime Log”, που, αν είναι ενεργοποιημένο, δίνει πληροφορίες όπως η διεύθυνση προσπέλασης, το tag της, κ.α.

Μπορεί να σας φανεί χρήσιμο και το “Memory Reference Visualization” tool. Δείχνει ένα «χάρτη» μιας περιοχής μνήμης του MIPS και σε κάθε προσπέλαση αλλάζει το χρώμα της αντίστοιχης λέξης. Έτσι μπορεί να δεί κανείς αν υπάρχει κάποιο μοτίβο στις διευθύνσεις που προσπελούνται κάθε φορά.



Σχήμα 1: Data Cache Simulator.

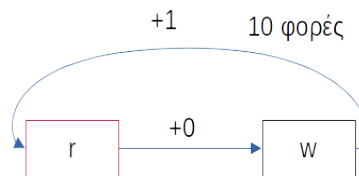
3 Μοτίβα προσπέλασης μνήμης

Για να καταλάβει κανείς πως οι προσπελάσεις μνήμης ενός προγράμματος επιδρούν στην κρυφή μνήμη, είναι πολύ σημαντικό να μπορεί να διακρίνει τα μοτίβα των προσπελάσεων αυτών.

Δείτε το παρακάτω πρόγραμμα σε C και θεωρήστε ότι ο πίνακας είναι πίνακας από bytes.

```
for (i = 0; i < 10; i++) a[i] = a[i]+1;
```

Υπάρχουν 2 προσπελάσεις μνήμης σε κάθε επανάληψη: η πρώτη είναι ανάγνωση και η δεύτερη εγγραφή. Και οι δύο προσπελάσεις είναι στην ίδια διεύθυνση. Σε κάθε επόμενη επανάληψη γίνονται πάλι 2 προσπελάσεις, αλλά σε διεύθυνση αυξημένη κατά 1 σε σχέση με την προηγούμενη επανάληψη.



Σχήμα 2: Μοτίβο 1.

Αρχική προσπέλαση			Μετάβαση 1			Προσπέλαση 2			Μετάβαση 2			Προσπέλαση 3			Μετάβαση 3		
R/W	Διεύθυνση (hex)		+/-	Βήμα (hex)	Επανάληψεις* (δεκαδικός)	R/W	+/-	Βήμα (hex)	Επανάληψεις* (δεκαδικός)	R/W	+/-	Βήμα (hex)	Επανάληψεις* (δεκαδικός)	R/W	+/-	Βήμα (hex)	Επανάληψεις* (δεκαδικός)
R	10010000		++	0	x	W	++	1	x 10								

Σχήμα 3: Το μοτίβο 1 στο quiz.

Ένας τρόπος να αναπαρασταθεί το μοτίβο φαίνεται στο σχήμα 2. Το κόκκινο τετράγωνο είναι η αρχική προσπέλαση και το r σημαίνει ανάγνωση. Το βέλος δείχνει στην επόμενη προσπέλαση και το +0 παριστάνει ότι γίνεται στην ίδια διεύθυνση. Η δεύτερη προσπέλαση είναι εγγραφή, και συμβολίζεται με ένα w. Η τρίτη προσπέλαση είναι ξανά η αρχική του μοτίβου. Η μετάβαση, εκτός από την αλλαγή στην διεύθυνση (το +1 στο βέλος), έχει και τον αριθμό επαναλήψεων του μοτίβου (10 φορές).

Για τους σκοπούς της άσκησης θα πρέπει να περιγράψετε μοτίβα σε ένα quiz στο ecourse. Το παραπάνω μοτίβο στη μορφή του quiz φαίνεται στο σχήμα 3. Η αρχική προσπέλαση περιλαμβάνει και την διεύθυνση, ενώ οι επόμενες προσπελάσεις δεν την χρειάζονται. Οι διευθύνσεις τους μπορούν να υπολογιστούν από το βήμα και την προηγούμενη διεύθυνση. Ο πίνακας του quiz, για να είναι γενικής χρήσης, έχει σε κάθε μετάβαση το πρόσημο, το βήμα και τον αριθμό επαναλήψεων. Τα δύο πρώτα είναι ίδια με το προηγούμενο σχήμα. Ο αριθμός επαναλήψεων πρέπει να συμπληρώνεται μόνο όταν αυτή η μετάβαση οδηγεί στην αρχή του μοτίβου, αλλιώς πρέπει να έχει την τιμή 0. Έτσι, για το παράδειγμα, στη μετάβαση 1 ο αριθμός επαναλήψεων είναι 0, ενώ στην μετάβαση 2 είναι 10. Οι επόμενες στήλες είναι χρήσιμες για μεγαλύτερα μοτίβα. Αν το μοτίβο τελειώνει πριν χρησιμοποιηθούν όλες οι στήλες, τις αφήνουμε κενές.

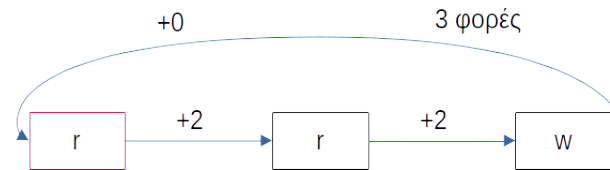
Χρειάζεται προσοχή στη μορφή που ζητούνται οι αριθμοί. Αυτοί που σχετίζονται με διευθύνσεις (αρχ. διεύθυνση, βήμα) είναι δεκαεξαδικόι ενώ οι επαναλήψεις θέλουν δεκαδικό (με βάση το 10) αριθμό. Μη βάλετε προθήματα (0x πριν τον δεκαεξαδικό) ή άλλη μορφή από αυτή που ζητείται.

Το παρακάτω πρόγραμμα έχει ένα πιο σύνθετο μοτίβο.

```
for (i = 0; i < 12; i += 4) a[i+4] = a[i] + a[i+2];
```

Θεωρούμε ξανά ότι ο πίνακας είναι από bytes και ότι οι προσπελάσεις μνήμης για το a[i] γίνεται πριν από αυτήν για το a[i+2]. Το μοτίβο που φαίνεται στην εικόνα, ξεκινάει με την προσπέλαση που αντιστοιχεί

στο $a[i]$ μετά στην $a[i+2]$ και τέλος στην εγγραφή στο $a[i+4]$. Στη συνέχεια το μοτίβο επαναλαμβάνεται και η επόμενη διεύθυνση που αντιστοιχεί στο νέο $a[i]$ είναι ίδια με την προηγούμενη (την εγγραφή στο $a[i+4]$) καθώς το i αυξάνεται κατά 4. Έτσι υπάρχει το +0 στο βέλος που δείχνει την επανάληψη του μοτίβου. Το μοτίβο επαναλαμβάνεται 3 φορές.



Σχήμα 4: Μοτίβο 2.

Αρχική προσπέλαση		Μετάβαση 1			Προσπέλαση 2	Μετάβαση 2			Προσπέλαση 3	Μετάβαση 3		
R/W	Διεύθυνση (hex)	+/-	Βήμα (hex)	Επαναλήψεις* (δεκαδικός)	R/W	+/-	Βήμα (hex)	Επαναλήψεις* (δεκαδικός)	R/W	+/-	Βήμα (hex)	Επαναλήψεις* (δεκαδικός)
R ▾	10010000	"+" ▾	2	x	R ▾	"+" ▾	2	x 0	W ▾	"+" ▾	0	x 3

Σχήμα 5: Το μοτίβο 2 στο quiz.

Τα σχήματα 4 και 5 δείχνουν την αναπαράσταση του δεύτερου μοτίβου.

3.1 Μοτίβα ευστοχιών - αστοχιών κρυφής μνήμης

Για να υπολογίσουμε το ποσοστό αστοχιών (ή ευστοχιών) είναι χρήσιμο να αναγνωρίζουμε μοτίβα ευστοχιών - αστοχιών που προκαλούνται από τις προσπελάσεις μνήμης ενός προγράμματος. Η αναπαράσταση που θα χρησιμοποιηθεί για αυτά τα μοτίβα είναι απλούστερη από την παραπάνω. Θα χρησιμοποιηθεί μια σειρά από τα γράμματα **H**, **M** που αντιστοιχούν σε ευστοχία, αστοχία, παρενθέσεις για να ομαδοποιούν μοτίβα και ο χαρακτήρας **x** και ένας αριθμός που είναι ο αριθμός επαναλήψεων του μοτίβου.

Για παράδειγμα το μοτίβο **(HMx2H)x2** (πλήρης ακολουθία: HMMHHMMH) σημαίνει ότι έγινε μια ευστοχία, 2 συνεχόμενες αστοχίες και άλλη μια ευστοχία και το μοτίβο επαναλήφθηκε δύο φορές. Συνολικά έγιναν 8 προσπελάσεις μνήμης. Το 'x' σχετίζεται με το αμέσως προηγούμενο γράμμα (το πρώτο x2 αναφέρεται στο M μόνο) και έτσι χρειάζονται παρενθέσεις για να ομαδοποιηθούν υπο-μοτίβα. Από το μοτίβο μπορούμε να υπολογίσουμε το ποσοστό ευστοχίας ως 50% καθώς σε 8 προσπελάσεις υπάρχουν 4 ευστοχίες (ή, αν το προσέξετε καλύτερα, σε κάθε 4 προσπελάσεις συμβαίνουν 2 ευστοχίες).

Σε μερικές ερωτήσεις, θα σας ζητηθεί να εξειδικεύσετε τις αστοχίες σε ένα από τα τρία είδη: Υποχρεωτικές, Χωρητικότητα, Σύγκρουσης και στα αντίστοιχα μοτίβα, θα χρησιμοποιήσετε αντί του M ένα από τα Y, X, S. Συνεχίζοντας το προηγούμενο παράδειγμα, αν οι δύο πρώτες αστοχίες ήταν υποχρεωτικές και οι δύο επόμενες χωρητικότητας, το μοτίβο θα πρέπει να αλλάξει σε **HYx2Hx2Xx2H**. Δεν μπορεί κανείς να επαναλάβει 2 φορές το αρχικό μοτίβο και τα δύο συνεχόμενα H από το τέλος του πρώτου και την αρχή του δεύτερου μοτίβου, έχουν παρασταθεί με **Hx2**.

4 Η άσκηση

Ξεκινήστε τον Mars, φορτώστε το cache.asm, περάστε το από τον assembler και ανοίξτε το Data Cache Simulation Tool.

Διαλέξτε προσεκτικά την οργάνωση: Fully Associative, LRU, Number of blocks 16, cache block size (words) 1. Το συνολικό μέγεθος της κρυφής μνήμης θα είναι 64 bytes και το Set size (blocks) 16. Επιλέξτε το enabled στο Runtime Log για να βλέπετε τη διεύθυνση στην οποία γίνεται προσπέλαση και αν προκαλεί ευστοχία ή αστοχία. **Πατήστε το "Connect to MIPS"**.

Αν θέλετε, ξεκινήστε και το Memory Reference Visualizer tool για να παρατηρείτε πως κάνει προσπελάσεις μνήμης το πρόγραμμα. Διαλέξτε τις τιμές κατάλληλα ώστε να φαίνεται μόνο ο πίνακας. Μη ξεχάσετε να πατήσετε το “Connect to MIPS”. Αναδιατάξτε τα παράθυρα ώστε να βλέπετε όσο περισσότερες πληροφορίες μπορείτε. Αν δεν έχετε μεγάλη οθόνη ίσως να χρειαστεί να μετακινείτε τα παράθυρα εργαλείων για να μπορείτε να δείτε τις τιμές των καταχωρητών ή το πρόγραμμα που εκτελείται (text segment στο tab execute).

Βάλτε ένα breakpoint στην εντολή lw που διαβάζει ένα στοιχείο του πίνακα. Πατήστε το Go (F5) που τρέχει το πρόγραμμα μέχρι το τέλος ή το πρώτο breakpoint. Όταν η εκτέλεση φτάσει στο breakpoint τρέξτε την lw με εκτέλεση εντολή προς εντολή (Step - F7). Παρατηρήστε τι συμβαίνει στα 2 εργαλεία. Η lw αστοχεί στην κρυφή μνήμη και θα δείτε κόκκινο χρώμα στην πρώτη γραμμή του Cache block table (το δεξί μέρος στο κέντρο του Data Cache Simulator). Συνεχίστε με εκτέλεση εντολή-προς-εντολή και δείτε τι κάνει η εντολή sw. Η sw ευστοχεί στη κρυφή μνήμη: φαίνεται με πράσινο χρώμα στην ίδια, πρώτη γραμμή του Cache block table. Η ευστοχία οφείλεται στην χρονική τοπικότητα αναφορών μνήμης (temporal locality): η sw γράφει στην ίδια διεύθυνση με την lw της ίδιας επανάληψης. Συνεχίστε την εκτέλεση με Go (F5) μέχρι την lw της επόμενης επανάληψης και πολλαπλά Step (F7) μέχρι και την sw που ακολουθεί.

Σε αυτό το σημείο ανοίξτε το quiz για να δείτε τί πρέπει να καταγράψετε. Το quiz αποτελείται από πέντε μέρη που είναι σχεδιασμένα να ολοκληρωθούν με τη σειρά. Μπορείτε να αφήσετε κενό αν δεν μπορείτε να απαντήσετε, αλλά μην προχωρήσετε χωρίς να έχετε προσπαθήσει τις προηγούμενες ερωτήσεις.

Στην αρχή κάθε μέρους περιγράφεται η οργάνωση της κρυφής μνήμης και τυχόν αλλαγές στις διαστάσεις του πίνακα του προγράμματος. Μετά ακολουθούν οι ερωτήσεις που πρέπει να απαντήσετε.

Προσοχή στην μορφή που ζητείται στις απαντήσεις. Όπου ζητείται δεκαεξαδικός αριθμός αυτός θα πρέπει γράφεται απευθείας χωρίς το συνηθισμένο πρόθεμα (0x). Σε μερικά σημεία (αριθμοί επαναλήψεων) ζητούνται δεκαδικοί (με βάση το δέκα) αριθμοί

4.1 Πρακτικά θέματα του Quiz

Οι βαθμοί που εμφανίζονται δίπλα στις ερωτήσεις του quiz δεν είναι σωστοί. Μή τους δίνετε σημασία.

Η βαθμολόγηση δεν θα γίνει αυτόματα, αλλά με υποβοήθεια από λογισμικό (gradescope).

Αν δεν συμπληρώσετε όλες μαζί τις ερωτήσεις, για παράδειγμα αν συμπληρώνετε ένα κομμάτι κάθε μέρα, υπάρχει ένα μικρό ενδεχόμενο το quiz να χάσει τις προηγούμενες τιμές, αν και έχει ρυθμιστεί να κρατάει τις προηγούμενες απαντήσεις. Γι'αυτό το λόγο κρατήστε τις απαντήσεις και σε χαρτί ή σε αρχείο για να μπορείτε να τις περάσετε όλες μαζί πριν την οριστική υποβολή του, χωρίς να χρειάζεται να ξαναλύσετε όλα τα ερωτήματα, αν συμβεί αυτό το πρόβλημα.

Το quiz είναι προγραμματισμένο να κλείνει αυτόματα την καταληκτική ημερομηνία και, επειδή είναι αυτόματο, είναι αυστηρό! Όποιος χρειάζεται και έχει διαθέσιμα «πάσα» για να πάρει παράταση, θα πρέπει να τη ζητήσει μέσω της φόρμας όπως στα προηγούμενα εργαστήρια. Όμως οι παρατάσεις περνούν χειροκίνητα στο Quiz και θα πρέπει η συμπλήρωση της φόρμας να γίνει νωρίς και θα πρέπει να είστε διαθέσιμοι να συμπληρώσετε το quiz την επόμενη μέρα. Μην περιμένετε ότι θα δώ την αίτηση για παράταση, θα αλλάξω την προθεσμία για εσάς και θα μπορέσετε να ολοκληρώσετε το quiz εκείνη τη στιγμή!