

7ο Εργαστήριο Αρχιτεκτονικής Η/Υ: Υπολογισμός πεδίων διεύθυνσης κρυφής μνήμης

Α. Ευθυμίου

Παραδοτέο: Παρασκευή 8 Μάη 2015, 23:00

Ο σκοπός αυτής της άσκησης είναι η κατανόηση της οργάνωσης κρυφών μνημών και πως αυτή επηρεάζεται από τις βασικές παραμέτρους μιας κρυφής μνήμης.

Θα γράψετε ένα πρόγραμμα σε Java που δέχεται το μέγεθος γραμμής, συσχετιστικότητα (associativity) και συνολική χωρητικότητα μιας κρυφής μνήμης και, για κάθε διεύθυνση μνήμης, θα υπολογίζει τα πεδία: tag, index, block offset. Στην επόμενη εργαστηριακή άσκηση θα χρησιμοποιήσετε τον περισσότερο κώδικα αυτής της άσκησης για να υλοποιήσετε έναν προσομοιωτή κρυφής μνήμης.

Θα πρέπει να έχετε μελετήσει τα μαθήματα σχετικά με κρυφές μνήμες που αντιστοιχούν στις ενότητες 5.1 - 5.3 και 5.5 του συγγράμματος.

1 Η άσκηση

Για να ξεκινήσετε θα χρειαστείτε τα αρχεία του εργαστηρίου δίνοντας τις εξής εντολές:

```
git remote add lab07_starter https://github.com/UoI-CSE-MYY402/lab06_starter.git
git fetch lab07_starter
git merge lab07_starter/master -m "Fetched lab07 starter files"
```

Στο αρχείο Lab07.java θα βρείτε το σκελετό του προγράμματος που μεταγλωττίζεται και τρέχει, αλλά δεν υπολογίζει σωστά τα πεδία tag, index, block offset.

Μεταγλωτίστε το (javac Lab07.java) και τρέξτε το με την εντολή: java Lab07 8 1 4096 traceFile. Ο πρώτος αριθμός της παραπάνω εντολής είναι το μέγεθος της γραμμής κρυφής μνήμης σε bytes. Ο επόμενος αριθμός είναι η συσχετιστικότητα (associativity ή number of ways) και ο τρίτος αριθμός είναι η συνολική χωρητικότητα (δεδομένων) της κρυφής μνήμης σε bytes. Από αυτές τις 3 βασικές παραμέτρους, μπορεί κανείς να υπολογίσει τον αριθμό των σετ, και πώς χωρίζεται μια διεύθυνση στα **3 πεδία: tag, index, block offset**.

Το τελευταίο όρισμα στην εντολή (traceFile) είναι ένα όνομα αρχείου, που περιέχει **διευθύνσεις των 48bit**, μία σε κάθε γραμμή. (Για την ώρα έχει μία μόνο γραμμή, αλλά μπορείτε να προσθέσετε και άλλες αν θέλετε). Όταν τρέξει, θα δείτε την έξοδο:

Address: 7f588cd9f718 => Tag: abcd, Index: 10, Boff: 4

που είναι η διεύθυνση, από το αρχείο και πως αυτή χωρίζεται στα 3 πεδία, μόνο που οι τιμές είναι λάθος! Αν το αρχείο είχε περισσότερες σειρές, θα υπήρχαν αντίστοιχα πολλές γραμμές εξόδου παρόμοιες με την παραπάνω.

2 Τί πρέπει να κάνετε

Ανοίξτε το αρχείο Lab07.java και αλλάξτε τον κώδικα ώστε να υπολογίζει σωστά τα 3 πεδία. Υπάρχουν σχόλια στον κώδικα που καθοδηγούν σε ποιά σημεία χρειάζονται αλλαγές. Όλες οι αλλαγές θα γίνουν στην κλάση Cache. Μην κάνετε αλλαγές στη Lab07.

Επειδή οι διευθύνσεις είναι 48 bit, και ο απλός τύπος int της Java είναι 32 bit, χρησιμοποιείτε μεταβλητές τύπου long (που είναι 64 bit) για ακέραιους αριθμούς, όπου περιμένετε μεγάλους αριθμούς (π.χ. το πεδίο tag). Δεν θα χρειαστεί να χρησιμοποιήσετε ακέραιους με περισσότερα από 64 bit.

Επειδή η Java έχει πολλαπλές βιβλιοθηκές και ενδέχεται να υπάρχει κάτι έτοιμο, για τον υπολογισμό των πεδίων, μη χρησιμοποιήσετε τίποτα εκτός από απλές αριθμητικές πράξεις, μάσκες (λογικές πράξεις and), ολισθήσεις και λογάριθμους.

Το πρόγραμμά σας θα πρέπει να δουλεύει σωστά για οποιαδήποτε οργάνωση κρυφής μνήμης, από direct mapped μέχρι fully-associative. Αυτές οι οργανώσεις προκύπτουν από κατάλληλες τιμές στα 3 (αριθμητικά) ορίσματα. Κάντε μερικές δοκιμές για να βεβαιωθείτε ότι δουλεύει σωστά. Θα χρειαστείτε τον κώδικα για την επόμενη άσκηση και, πιθανότατα, δεν θα έχει ολοκληρωθεί η βαθμολόγηση αυτής μέχρι τότε.

Οι αλλαγές που χρειάζονται είναι πολύ λίγες (κάτω από 10 γραμμές κώδικα και μερικές δηλώσεις μεταβλητών), αρκεί να γνωρίζει κανείς πως είναι οργανωμένη μια κρυφή μνήμη και πώς να χειριστεί δυαδικούς αριθμούς.

3 Παραδοτέο

Το παραδοτέο της άσκησης είναι το αρχείο Lab07.java. Η παράδοση θα γίνει μέσω GitHub, όπως πάντα.