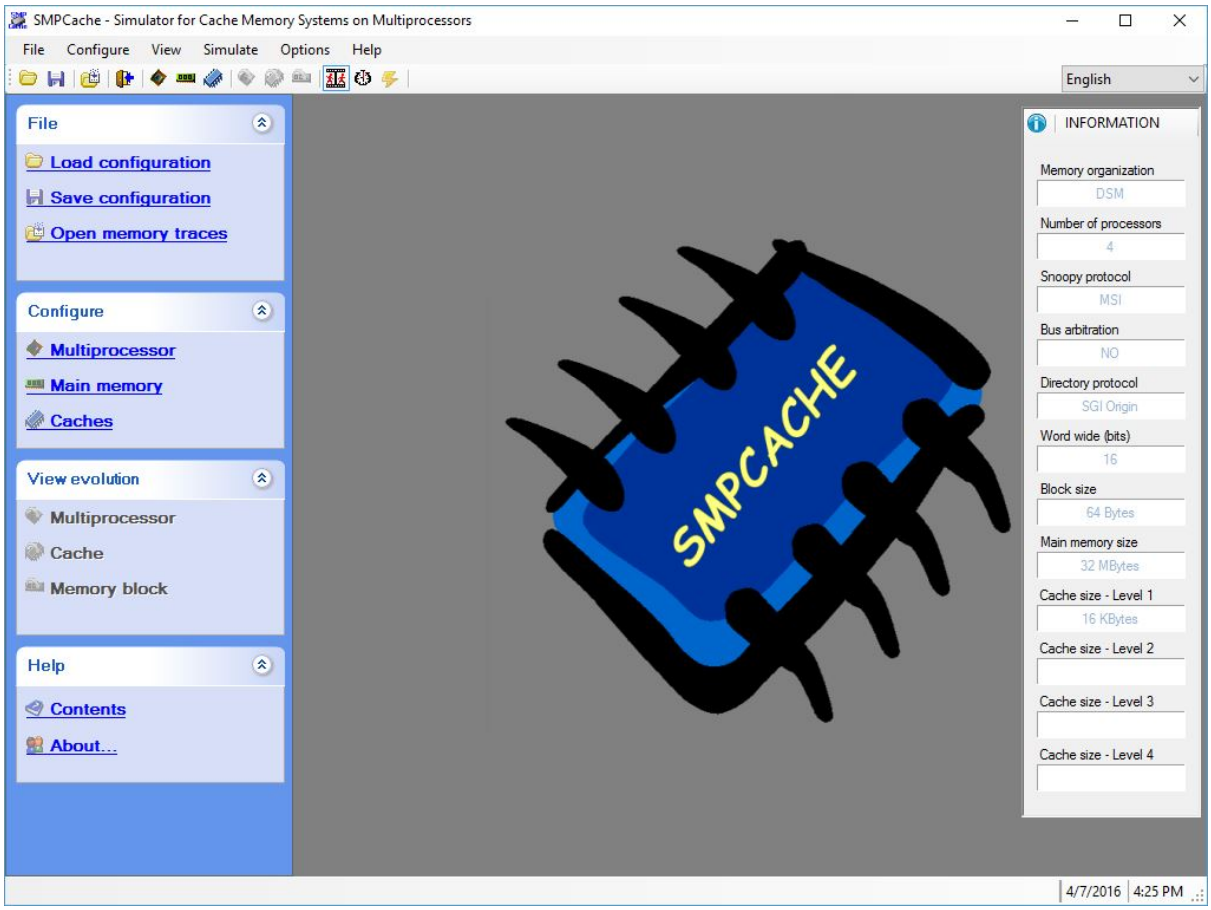
SMPcache

Ένα εργαλείο για προσομοίωση-οπτικοποίηση κρυφής μνήμης (Cache)

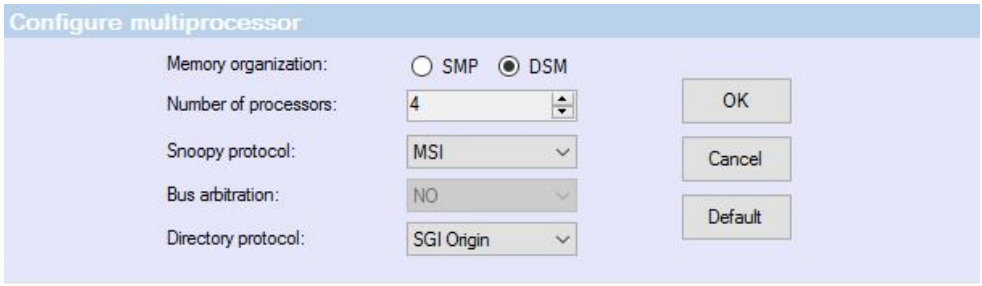
1. Βασικές ρυθμίσεις του συστήματος: δημιουργία μια δικής μας σύνθεσης συστήματος.

Το SMPcache είναι ένα εργαλείο με το οποίο μπορούμε να προσομοιώσουμε τη λειτουργία μιας μνήμης cache. Μάλιστα είναι αρκετά ισχυρό, καθώς μπορεί να υποστηρίξει πολύ πιο σύνθετα συστήματα μνημών και επεξεργαστών από αυτά που μελετούμε στο παρόν μάθημα. Έχει εγκατασταθεί στα μηχανήματα του τμήματος, και επειδή μας έχει παραχωρηθεί με ειδική άδεια ειδικά για χρήση από το πανεπιστήμιο, δυστυχώς δε μπορεί να δοθεί σε κάθε φοιτητή.

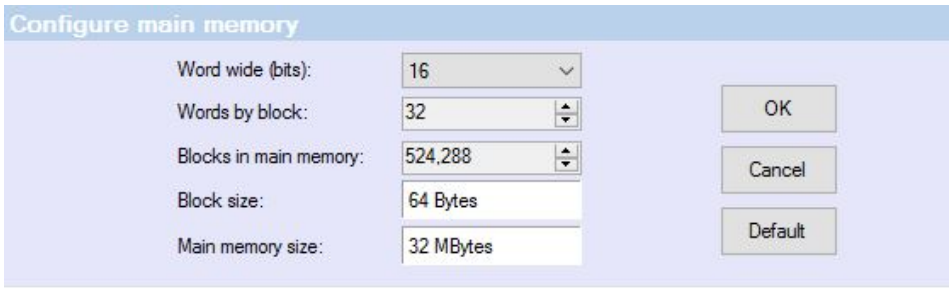
Ανοίγοντας το SMPcache, αρχικά βλέπουμε ένα παράθυρο όπως φαίνεται στην εικόνα:



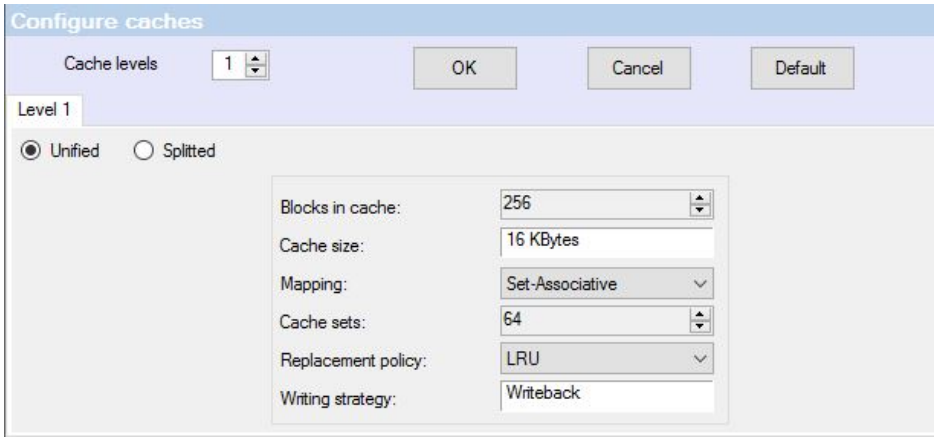
Στα αριστερά βλέπουμε τις διάφορες επιλογές, και στα δεξιά τη σύνθεση (configuration) του συστήματος που έχουμε δημιουργήσει. Τώρα που δεν έχουμε πειράξει τίποτε, έχουμε μια στάνταρ σύνθεση, την οποία και θα αλλάξουμε. Γι αυτό από την κατηγορία “Configure” θα επισκεφθούμε μία μία όλες τις επιλογές για να φτιάξουμε το σύστημά μας όπως το θέλουμε. Αυτά τα μενού υπάρχουν και στη γραμμή των μενού και στη γραμμή εργαλείων σαν κουμπάκια και στο αριστερό υπο-παράθυρο. Για να φτιάξουμε την πρώτη δική μας σύνθεση, πάμε αρχικά στο Configure→Multiprocessor:



Αυτό που μας ενδιαφέρει σε αυτό το σημείο είναι να βάλουμε αριθμό επεξεργαστών «1». Στη συνέχεια πάμε στο “Main memory” :



Εδώ πρέπει να βάλουμε το πλήθος των bits ανά block (στη δική μας άσκηση είναι 4 Bytes = 32 bits), το πλήθος των λέξεων ανά block (Words by block), και το πόσα blocks θέλουμε να έχει η κεντρική μας μνήμη. (Με βάση αυτά θα προκύψουν αυτόματα και τα Block size και Main memory size). Στη συνέχεια πηγαίνουμε στο “Caches”:



Εδώ μας ενδιαφέρει κυρίως το μέγεθος της Cache. Συγκεκριμένα βάζουμε το πόσα blocks χωράει η cache (το μέγεθος του block έχει ήδη καθοριστεί πριν, μέσω της κεντρικής μνήμης). Αν θέλουμε η κρυφή μας μνήμη να είναι προσεταιριστική (associative) επιλέγουμε Set-Associative, και από κάτω πόσα Set θέλουμε να υπάρχουν. Για τη δική μας άσκηση θα επιλέξουμε στο “Mapping” την επιλογή “Direct” γιατί θέλουμε μνήμη άμεσης απεικόνισης. (Εδώ μπορούμε επίσης να διαλέξουμε αν θέλουμε ξεχωριστή κρυφή μνήμη για εντολές και δεδομένα, καθώς και στρατηγική αντικατάστασης του block σε περίπτωση προσεταιριστικής μνήμης. Τέλος να πούμε ότι για τις πολιτικές εγγραφών, το εργαλείο μας δίνει μόνο την επιλογή “Writeback”).

Κάτι πολύ χρήσιμο που προφέρει το εργαλείο είναι η επιλογή να αποθηκεύσουμε τη σύνθεση που έχουμε φτιάξει. Αυτό θα μας βοηθήσει και για να μη χρειάζεται να ξανακάνουμε τις ρυθμίσεις κάθε φορά που ανοίγουμε το εργαλείο, αλλά και για να δοκιμάζουμε τις διαφορετικές συνθέσεις που μας ζητάει αυτή η άσκηση (ή πιθανόν και άλλες με τις οποίες θα θέλαμε εύκολα να πειραματιστούμε).

Για να το κάνουμε αυτό, αφού έχουμε τελειώσει με τις ρυθμίσεις της σύνθεσής μας, πατάμε “Save configuration”. Προσοχή σε ποιο φάκελο αποθηκεύονται αυτά τα αρχεία. Είναι προτιμότερο να αλλάξετε το φάκελο αυτό, να αντιστοιχεί σε κάποιο φάκελο με την άσκησή σας. Έπειτα όταν χρειαστεί μπορείτε να φορτώσετε μια σύνθεση από το “Load configuration”.

2. Εκτέλεση προσομοίωσης με τη χρήση αρχείων ιχνών (trace files).

2.1. Δημιουργία των αρχείων

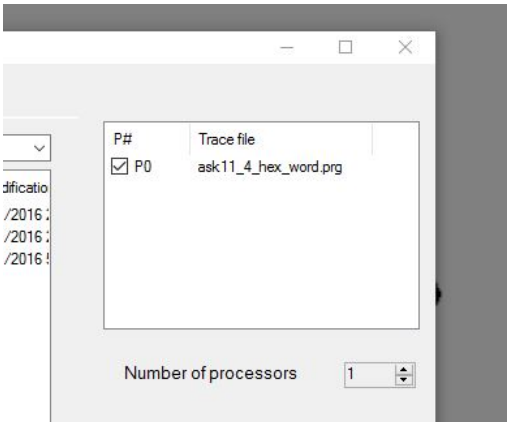
Για να πειραματιστούμε με τον τρόπο λειτουργίας του συστήματός μας, χρειαζόμαστε ένα (ή περισσότερα) αρχεία που μας δίνουν μια σειρά από προσπελάσεις στη μνήμη. Αυτά τα αρχεία συνήθως προκύπτουν από κανονικά προγράμματα, από τα οποία κρατούμε για το σκοπό μας μόνο το κομμάτι που αφορά τις προσπελάσεις του προγράμματος στη μνήμη. Ένα τέτοιο αρχείο που περιέχει τις προσπελάσεις αυτές, με κατάλληλο συντακτικό, λέγεται trace file. (Μάλιστα το εργαλείο SMPcache συνοδεύεται από μεγάλα trace files που έχουν προκύψει από γνωστά benchmarks, τόσο για απλούς (μονούς) επεξεργαστές όπως της άσκησής μας, όσο και για πολυεπεξεργαστές. Αυτά φυσικά είναι πολύ μεγάλα και πολύπλοκα για τους σκοπούς και τα μεγέθη αυτής της άσκησης.) Το εργαλείο αυτό έχει το εξής συντακτικό για ένα trace file: μια προσπέλαση μνήμης σε κάθε γραμμή. Κάθε γραμμή έχει δύο αριθμούς. Ο 1^{ος} αριθμός μας λέει το είδος τις προσπέλασης (0 = Instruction Fetch, 2 = Read, 3 = Write), και ο 2^{ος} μας λέει τη διεύθυνση μνήμης. Μιας και εμείς για να βρούμε cache misses δεν ενδιαφερόμαστε για το είδος της προσπέλασης, αρκεί σε αυτή την άσκηση να βάλουμε σε όλες τις προσπελάσεις σαν εντολή την Read. Εδώ πρέπει να προσέξουμε, γιατί το εργαλείο αυτό περιμένει διευθύνσεις word (και όχι Byte), δηλαδή το σύστημά μας δεν είναι Byte-addressable, αλλά word-addressable. Άρα για προσπελάσεις (όπως της άσκησής μας,) που δίνονται διευθύνσεις Byte για ένα σύστημα με word-size=4Bytes, θα πρέπει να γίνει μετατροπή σε δ/νση Word, δηλαδή διαίρεση με το 4. Π.χ. έστω ότι έχουμε μια προσπέλαση στη μνήμη που αντιστοιχεί στο Byte 100 (όπως η πρώτη δ/νση της άσκησης 11.4). Αυτό επειδή αντιστοιχεί στο 100/4=25^{ο} word, θα πρέπει στο trace file να χρησιμοποιήσουμε τη διεύθυνση του word, δηλαδή $(25)_{10}$. Εδώ όμως είναι κι ένα ακόμη σημείο που θέλει προσοχή. Το εργαλείο SMPcahce περιμένει τις διευθύνσεις μνήμης στο δεκαεξαδικό, επομένως πρέπει να κάνουμε κι αυτή τη μετατροπή! Δηλαδή στο παράδειγμά μας να γράψουμε 19 (έτσι το SMPcache καταλαβαίνει 19 στο δεκαεξαδικό, αλλά πρέπει να το γράφουμε έτσι, σκέτο, ούτε 0x19 ούτε φυσικά κάτι του στυλ 19_{16})

Για διευκόλυνση σας δίνεται μια εντολή command line για Linux, με την οποία αν έχετε ένα αρχείο text με τις προσπελάσεις μνήμης σε Byte-addresses στο δεκαδικό, μία ανά γραμμή, σας εκτυπώνει το trace file, μετατρέποντας από δεκαδικό σε δεκαεξαδικό, και από Byte-address σε Word-address (word-size=4), ενώ βάζει και τον αριθμό 2 σε κάθε γραμμή (Read):

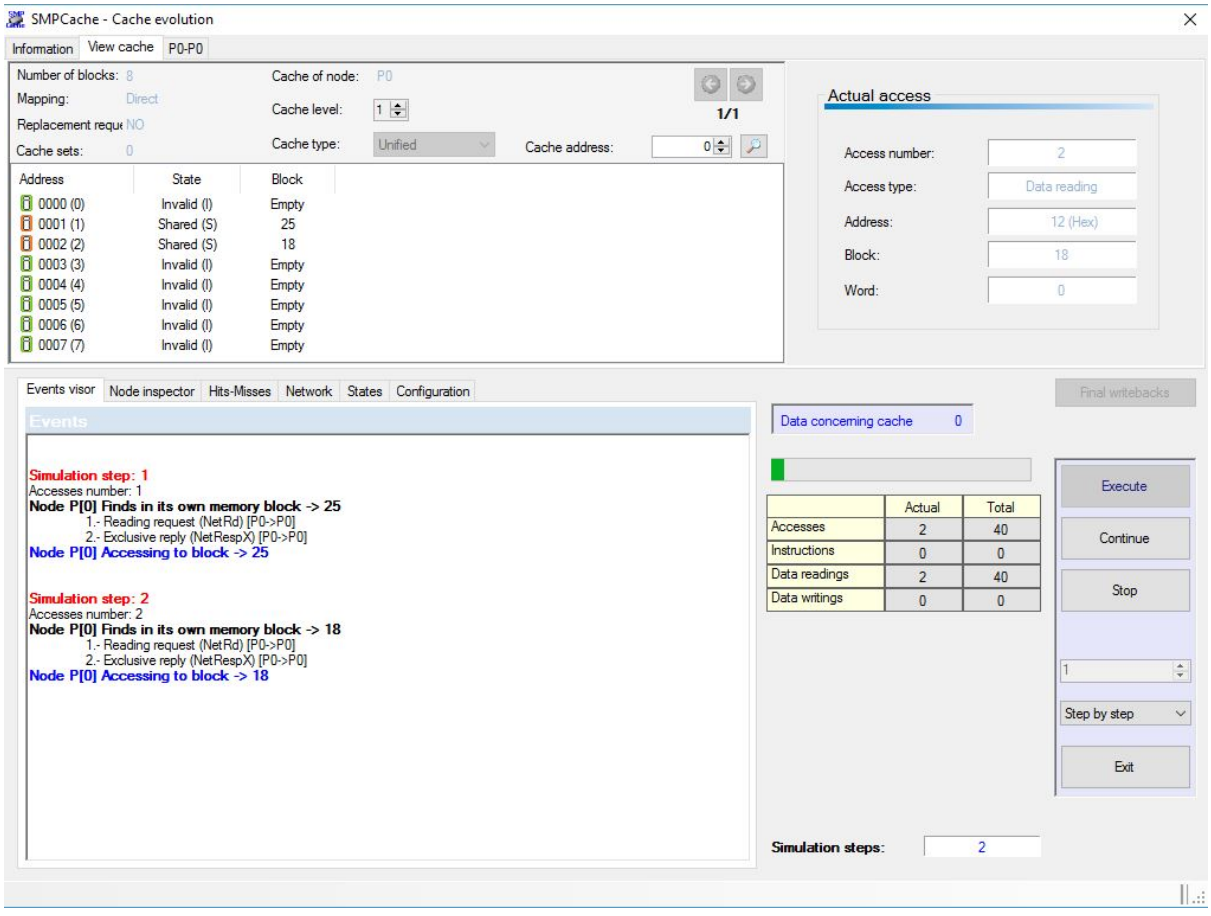
```
cat myFile.txt |awk '{printf "2  %x\n",$1/4}'
```

2.2. Εκτέλεση της προσομοίωσης

Έχοντας ένα trace file, μπορούμε να ξεκινήσουμε την προσομοίωσή μας, και να πάρουμε αποτελέσματα για την τοποθέτηση των blocks, τα miss rate και hit rate κ.α. Από το “Open memory traces” επιλέγουμε το αρχείο με τις προσπελάσεις μνήμης που έχουμε φτιάξει. Κατά το άνοιγμα, στο δεξί-πάνω μέρος τους παραθύρου, θα δούμε την επιλογή που φαίνεται στην παρακάτω εικόνα. Εκεί θα πρέπει να επιλέξουμε το P0 (ουσιαστικά εδώ επιλέγουμε να γίνουν οι προσπελάσεις από τον επεξεργαστή P0. Θυμηθείτε ότι το εργαλείο μπορεί να υποστηρίξει προσομοίωση πολυεπεξεργαστών).



Στη συνέχεια για να δούμε τι θα γίνει τρέχοντας την προσομοίωση, θα πρέπει να διαλέξουμε ποιο μέρος του συστήματος θέλουμε να δούμε. Από το “View evolution” λοιπόν επιλέγουμε το “Cache”, ενώ στην αμέσως επόμενη ερώτηση στο αναδυόμενο παράθυρο επιλέγουμε “Text”. Αυτό που θα μας εμφανιστεί στη συνέχεια είναι το παράθυρο που βλέπετε παρακάτω:



Μέσω αυτού του παραθύρου θα πάρουμε όλες τις πληροφορίες που θέλουμε. Αρχικά θα πρέπει να πατήσουμε το κουμπί “Execute”, με το οποίο θα ξεκινήσει η προσομοίωση, και θα γίνει και η πρώτη προσπέλαση. Στη συνέχεια για να εκτελέσουμε κι άλλες προσπελάσεις, πατάμε το κουμπί Continue”. Κάθε φορά που το πατάμε, θα εκτελούνται προσπελάσεις (με βάση το trace file), τόσες όσες μας λέει η πιο κάτω επιλογή. Στη φωτογραφία έχω επιλέξει Step by Step, επομένως εκτελεί μία μία. Αν θέλω επιλέγω Complete, ώστε να τρέξει όλο το trace. Ας ασχοληθούμε τώρα με το τι άλλο βλέπουμε στο παράθυρό μας. Στο πάνω μέρος βλέπουμε το τι περιέχει η cache. Ποιο μπλοκ βρίσκεται σε ποια θέση της cache. Στο κάτω μέρος βλέπουμε μια εξήγηση της μετακίνησης του μπλοκ που έγινε. Αν από τα κουμπιά που βρίσκονται στη μέση του παραθύρου, επιλέξουμε “Hits-Misses” μπορούμε να δούμε

πόσες προσπελάσεις είναι Hits και πόσες Misses, καθώς και τα ποσοστά που αντιστοιχούν. Όπως βλέπετε υπάρχουν και πολλές άλλες επιλογές, όπως π.χ. η κατάσταση των blocks με βάση τα πρωτόκολλα MSI/MESI, η διασύνδεση των επεξεργαστών κ.α., αλλά δε θα μας χρειαστούν για τους σκοπούς αυτής της άσκησης. Στο τέλος μια προσομοίωσης, θα ερωτηθείτε αν θέλετε να ολοκληρωθούν τα Write Back. Ότι και να επιλέξετε (Ναι ή Όχι) δε μας ενδιαφέρει, γιατί και δε θα χρειαστεί να μετρήσουμε το χρόνο για κάτι τέτοιο στην παρούσα άσκηση, και εκτός αυτού, αν στο trace μας είχαμε μόνο αναγνώσεις, δεν υπάρχουν dirty blocks για να γραφτούν πίσω στην κεντρική μνήμη.