Πανεπιστήμιο Θεσσαλίας
Τμήμα Πληροφορικής

Αρχιτεκτονική Υπολογιστών

Λυμένες Ασκήσεις

Άσκηση 11:

Έστω ένα σύστημα μνήμης, στο οποίο έχουμε προσθέσει μια κρυφή μνήμη θυμάτων 16 θέσεων μεταξύ της κρυφής μνήμης δεδομένων L1 και της κρυφής μνήμης L2. Μελετήστε την επίδραση της κρυφής μνήμης θυμάτων στην απόδοση του συστήματος μνήμης, τόσο ποιοτικά, όσο και ποσοτικά. Για την ποσοτική προσέγγιση, υποθέστε ότι η αξιολόγηση του συστήματος μνήμης γίνεται πάνω σε κάποια εφαρμογή, όπου η κρυφή μνήμη δεδομένων L1 εμφανίζει ποσοστό επιτυχίας 90% και η κρυφή μνήμη θυμάτων εμφανίζει ποσοστό επιτυχίας 50%. Ο χρόνος προσπέλασης σε επιτυχία για την L1 είναι 2cc, ο μέσος χρόνος προσπέλασης για την L2 είναι 20cc, ενώ ο χρόνος προσπέλασης της μνήμης θυμάτων είναι 3cc. Για απλούστευση θεωρήστε ότι μια αποτυχημένη προσπέλαση αυξάνει το χρόνο προσπέλασης σε επιτυχία κατά το μέσο χρόνο προσπέλασης του επόμενου επιπέδου.

Απάντηση

Η κρυφή μνήμη θυμάτων (victim cache) τοποθετείται ανάμεσα σε δύο επίπεδα της ιεραρχίας μνήμης, ώστε να συγκρατεί τα πρόσφατα εκδιωχθέντα μπλοκ του ενός, καθώς στέλνονται στο άλλο, και να προσφέρει μια ευκαιρία σύντομης ανάκτησης σε περίπτωση που κάποιο από αυτά ξαναζητηθεί. Η κρυφή μνήμη θυμάτων δεν έχει απεικόνιση διεύθυνσης, υλοποιείται δηλαδή ως πλήρως συσχετιστική μνήμη. Έτσι, τα πρώτα μπλοκ που διώχνονται από το μικρότερο από τα δύο επίπεδα θα καταλάβουν θέσεις της μνήμης θυμάτων, και όταν αυτή γεμίσει, τα επόμενα μπλοκ θα αντικαθιστούν τα προηγούμενα, με βάση κάποιον αλγόριθμο σαν τον LRU ή τον FIFO. Με κάθε αποτυχία στην κρυφή μνήμη που βρίσκεται πριν τη μνήμη θυμάτων, πρώτα ελέγχεται η μνήμη θυμάτων, ώστε αν το ζητούμενο μπλοκ βρίσκεται εκεί, να ανακτηθεί άμεσα, χωρίς να γίνει προσπέλαση στη μνήμη του επόμενου επιπέδου. Φυσικά, αν το μπλοκ δε βρίσκεται στη μνήμη θυμάτων, είτε γιατί δεν πέρασε ποτέ από αυτήν, είτε γιατί αντικαταστάθηκε σε κάποια προηγούμενη αποτυχία, θα γίνει αναγκαστικά προσπέλαση του επόμενου επιπέδου. Η επίδραση της κρυφής μνήμης θυμάτων στην απόδοση ενός συστήματος μνήμης εξαρτάται από το ποσοστό επιτυχίας της, ιδιαίτερα σε σχέση με το ποσοστό επιτυχίας της κρυφής μνήμης που προηγείται, αλλά και από την επικάλυψη που έχει η προσπέλασή της με την επικάλυψη των δύο γειτονικών επιπέδων του συστήματος μνήμης. Γενικά, μπορούμε να περιμένουμε θετική επίδραση, επειδή με βάση την αρχή της τοπικότητας αναφορών, υπάρχει σημαντική πιθανότητα ένα μπλοκ που αντικαταστάθηκε από την κρυφή μνήμη πριν τη μνήμη θυμάτων να ξαναζητηθεί σύντομα, και πριν αυτό αντικατασταθεί από την τελευταία. Μάλιστα, αν η χρόνος προσπέλασης της κρυφής μνήμης θυμάτων καλύπτεται πλήρως από το χρόνο προσπέλασης της μνήμης του προηγούμενου ή του επόμενου επιπέδου, τότε όσο μικρή και αν είναι αυτή η πιθανότητα, θα υπάρχει πάντα βελτίωση στην απόδοση του συστήματος μνήμης, αφού στη χειρότερη περίπτωση κάθε αποτυχία στην πρώτη κρυφή μνήμη θα οδηγεί σε προσπέλαση της δεύτερης, η οποία όμως θα γίνεται χωρίς καθυστέρηση, κι επομένως θα παρέχει απόδοση ίση με αυτή ενός συστήματος χωρίς μνήμη θυμάτων. Αν όμως ο χρόνος προσπέλασης της μνήμης θυμάτων δεν καλύπτεται από τις προσπελάσεις των δύο επιπέδων του συστήματος μνήμης, κάτι που θα μπορούσε να είναι επιθυμητό, ώστε από τη μια να μην αυξάνεται η κατανάλωση ενέργειας και από την άλλη η διασύνδεση με τη μνήμη επόμενου επιπέδου να μην απασχολείται άσκοπα και να μπορεί να διατίθεται για άλλες προσπελάσεις, τότε ένα ιδιαίτερα χαμηλό ποσοστό επιτυχίας στη μνήμη θυμάτων θα μπορούσε να οδηγήσει σε μείωση στην απόδοση του συστήματος μνήμης αντί για αύξηση.

Ποιοι όμως είναι οι παράγοντες που ρυθμίζουν το ποσοστό επιτυχίας στην κρυφή μνήμη θυμάτων; Φυσικά, όπως προαναφέραμε, η τοπικότητα αναφορών αυξάνει αυτό το ποσοστό. Αυτός όμως ο παράγοντας εξαρτάται από την εφαρμογή, και σίγουρα υπάρχουν εφαρμογές, όπου είτε η τοπικότητα αναφορών είναι πολύ μικρή στο χώρο, είτε οι αναφορές επαναλαμβάνονται υπερβολικά αραιά. Για παράδειγμα, μια εφαρμογή που χειρίζεται μεγάλους πίνακες με μέτριο βαθμό πλήρωσης δεδομένων είναι πιθανό να εμφανίζει και τα δύο αυτά συμπτώματα. Το μέγεθος της κρυφής μνήμης θυμάτων είναι σίγουρα ένας σημαντικός παράγοντας ρύθμισης του ποσοστού επιτυχίας της, ιδιαίτερα εφ' όσον αυτή υλοποιείται ως πλήρους συσχέτισης κι επομένως δεν εμφανίζει αποτυχίες από συγκρούσεις απεικόνισης. Για σταθερό μέγεθος μνήμης θυμάτων, η οργάνωση της κρυφής μνήμης που προηγείται ίσως να είναι ο πιο καθοριστικός παράγοντας ρύθμισης του ποσοστού επιτυχίας της μνήμης θυμάτων, ανεξάρτητα από την εφαρμογή. Πιο συγκεκριμένα, μια κρυφή μνήμη άμεσης απεικόνισης έχει γενικά χαμηλότερη απόδοση σε αποθήκευση δεδομένων από μια συνολοσυσχετιστική κρυφή μνήμη, επειδή η δεύτερη παρέχει περισσότερες ευκαιρίες αποθήκευσης σε μπλοκ που έχουν την ίδια διεύθυνση συνόλου από ό,τι η πρώτη. Η ύπαρξη πολλαπλών ταυτόχρονα ενεργών περιοχών μνήμης είναι περισσότερο συνηθισμένη σε δεδομένα από ό,τι σε εντολές, εκτός εάν ο επεξεργαστής υποστηρίζει πολλαπλές ροές ελέγχου, κι έτσι μια κρυφή μνήμη άμεσης απεικόνισης θα αντικαθιστά τα μπλοκ της πιο συχνά από ό,τι μια συνολοσυσχετιστική κρυφή μνήμη. Έτσι, για τον ίδιο χώρο αποθήκευσης, μια μνήμη θυμάτων για δεδομένα θα γεμίζει γρηγορότερα με μια κρυφή μνήμη άμεσης απεικόνισης, και θα έχει επομένως συχνότερες αντικαταστάσεις, δηλαδή χαμηλότερο ποσοστό επιτυχίας. Αντίθετα, με μια συνολοσυσχετιστική κρυφή μνήμη, τα μπλοκ που θα καταλήγουν στη μνήμη θυμάτων θα είναι γενικά αυτά που δε θα χωρούν στα σύνολα της πρώτης, κάτι που θα συμβαίνει όταν οι ταυτόχρονα ενεργές περιοχές μνήμης είναι περισσότερες από το βαθμό συσχέτισης της μνήμης. Με κατάλληλο βαθμό συσχέτισης, συνήθως όχι μεγαλύτερο από 4, αυτό θα συμβαίνει ιδιαίτερα σπάνια, κι έτσι η μνήμη θυμάτων θα διατηρεί τα δεδομένα για μεγαλύτερο χρονικό διάστημα, κι έτσι θα έχει υψηλότερο ποσοστό επιτυχίας. Από την άλλη μεριά, τέλος, για αποθήκευση εντολών, δε θα πρέπει να αναμένεται διαφορά στην απόδοση μιας κρυφής μνήμης άμεσης απεικόνισης από μια συνολοσυσχετιστική, εκτός αν υποστηρίζονται πολλαπλές ροές ελέγχου, οπότε η συμπεριφορά τείνει να μοιάζει με αυτή της αποθήκευσης δεδομένων.
Πάντως, αν ασχοληθούμε με το σχετικό ποσοστό επιτυχίας μεταξύ της κρυφής μνήμης θυμάτων και της κρυφής μνήμης που προηγείται, μια καλύτερη συμπεριφορά συνολοσυσχετιστικής κρυφής μνήμης ίσως να οδηγεί σε μικρότερη βελτίωση στην απόδοση, από ό,τι για μια μνήμη άμεσης απεικόνισης. Μ' άλλα λόγια, μια κρυφή μνήμη άμεσης απεικόνισης με μια σχετικά μεγάλη κρυφή μνήμη θυμάτων μπορεί να παρουσιάσει δραματική βελτίωση στην απόδοση του συστήματος μνήμης, ενώ μια συνολοσυσχετιστική κρυφή μνήμη που ήδη έχει υψηλά ποσοστά επιτυχίας πιθανό να μην παρουσιάσει τέτοια βελτίωση, όσο μεγάλη κι αν είναι η κρυφή μνήμη θυμάτων που συνδέεται μ' αυτήν.

Ας δούμε τώρα πόση είναι η βελτίωση στην απόδοση του δεδομένου συστήματος μνήμης με την παρουσία της κρυφής μνήμης θυμάτων. Έστω τ_1 ο χρόνος προσπέλασης της κρυφής μνήμης L1, τ ο χρόνος προσπέλασης της μνήμης θυμάτων και T_2 ο μέσος χρόνος προσπέλασης της κρυφής μνήμης L2, συμπεριλαμβανομένων όλων των προσπελάσεων στα επόμενα επίπεδα του συστήματος μνήμης. Έστω ακόμα p_1 το ποσοστό επιτυχίας της μνήμης L1 και p το ποσοστό επιτυχίας της μνήμης θυμάτων. Έτσι, αν T_χ είναι ο μέσος χρόνος προσπέλασης της μνήμης χωρίς κρυφή μνήμη θυμάτων, τότε ο χρόνος αυτός θα είναι:

$$T_\chi = \tau_1 + (1 - p_1) \times T_2$$

Στη συγκεκριμένη άσκηση θα υποθέσουμε ότι η προσπέλαση της κρυφής μνήμης θυμάτων επικαλύπτεται με την προσπέλαση της L1, αλλά δεν επικαλύπτεται με την προσπέλαση της L2. Μ' άλλα λόγια, κάθε προσπέλαση που γίνεται στην κρυφή μνήμη L1 πηγαίνει παράλληλα και στη μνήμη θυμάτων, αλλά η προσπέλαση της L2 δεν ξεκινάει πριν ελεγχθεί και η τελευταία. Με $\tau \geq \tau_1$ ο μέσος χρόνος προσπέλασης της μνήμης με κρυφή μνήμη θυμάτων T_μ θα είναι:

$$T_\mu = \tau_1 + (1 - p_1) \times (\tau - \tau_1) + (1 - p) \times (1 - p_1) \times T_2$$

όπου ο όρος που ακολουθεί το χρόνο προσπέλασης της L1 είναι ο πρόσθετος χρόνος προσπέλασης της μνήμης θυμάτων, ο οποίος προστίθεται σε κάθε αποτυχία στην L1 – άρα με πιθανότητα $(1 - p_1)$, και ο τελευταίος όρος είναι ο μέσος χρόνος προσπέλασης της L2, που προστίθεται σε κάθε αποτυχία της L1 που αποτυγχάνει και στη μνήμη θυμάτων – άρα με πιθανότητα ίση με το γινόμενο $(1 - p) \times (1 - p_1)$.
Η βελτίωση στο μέσο χρόνο προσπέλασης της μνήμης Sp θα είναι ο λόγος των δύο χρόνων:

$$Sp = \frac{T_\chi}{T_\mu} = \frac{\tau_1 + (1 - p_1) \times T_2}{\tau_1 + (1 - p_1) \times (\tau - \tau_1) + (1 - p) \times (1 - p_1) \times T_2}$$

Μπορούμε να δούμε ότι όσο αυξάνει ο χρόνος τ, αυξάνει ο δεύτερος όρος του χρόνου T_μ, κι επομένως η βελτίωση στην απόδοση του συστήματος μνήμης μειώνεται. Έτσι, η οριακή τιμή του χρόνου τ για την οποία παύει να υπάρχει βελτίωση θα προκύπτει ως εξής:

$$\begin{aligned} Sp \le 1 &\Leftrightarrow \tau_1 + (1 - p_1) \times T_2 \le \tau_1 + (1 - p_1) \times (\tau - \tau_1) + (1 - p) \times (1 - p_1) \times T_2 \Leftrightarrow \\ &\Leftrightarrow p \times (1 - p_1) \times T_2 \le (1 - p_1) \times (\tau - \tau_1) \Leftrightarrow \\ &\Leftrightarrow \tau \ge \tau_1 + p \times T_2 \end{aligned}$$

Ισοδύναμα, η οριακή τιμή του ποσοστού p για την οποία παύει να υπάρχει βελτίωση θα δίνεται από τη σχέση:

$$p \le \frac{\tau - \tau_1}{T_2}$$

Όταν $\tau \le \tau_1$, οπότε η προσπέλαση της μνήμης θυμάτων καλύπτεται πλήρως από την προσπέλαση της L1, ο χρόνος T_μ θα είναι:

$$T_\mu = \tau_1 + (1 - p) \times (1 - p_1) \times T_2$$

οπότε η βελτίωση Sp θα γίνεται:

$$Sp = \frac{T_\chi}{T_\mu} = \frac{\tau_1 + (1 - p_1) \times T_2}{\tau_1 + (1 - p) \times (1 - p_1) \times T_2} = 1 + \frac{p \times (1 - p_1) \times T_2}{\tau_1 + (1 - p) \times (1 - p_1) \times T_2}$$

όπου διαπιστώνουμε ότι πάντα $Sp \ge 1$.
Η μέγιστη βελτίωση εμφανίζεται όταν $\tau \le \tau_1$ και $p \approx 1$, και θα είναι:

$$Sp_{\max} = 1 + (1 - p_1) \times \frac{T_2}{\tau_1}$$

Για τις τιμές που μας δίνονται η βελτίωση θα είναι:

$$Sp = \frac{2cc + 0.1 \times 20cc}{2cc + 0.1 \times 1cc + 0.5 \times 0.1 \times 20cc} = 1.29$$

Με διαφορετικές υποθέσεις στην επικάλυψη των προσπελάσεων στις τρεις μνήμες θα προέκυπταν διαφορετικά συμπεράσματα, χρησιμοποιώντας όμως παρόμοια λογική με την παραπάνω. Ειδικότερα, χωρίς επικάλυψη στις προσπελάσεις της μνήμης θυμάτων και της L1, στο χρόνο T_μ θα προσθέταμε όλο το χρόνο τ, αντί της διαφοράς $\tau - \tau_1$, ενώ με επικάλυψη στις προσπελάσεις της μνήμης θυμάτων και της L2, ο χρόνος προσπέλασης της μνήμης θυμάτων θα συμμετείχε μόνο σε περίπτωση επιτυχίας στην προσπέλασή της, αφού σε άλλη περίπτωση καλύπτεται από το χρόνο προσπέλασης της L2.

Άσκηση 12:

Θεωρήστε ένα σύστημα συμμετρικού πολυεπεξεργαστή, στο οποίο θέλουμε να υλοποιήσουμε συγχρονισμό με βάση την αδιάσπαστη λειτουργία compare&swap, η οποία συγκρίνει το περιεχόμενο μιας διεύθυνσης μνήμης με το περιεχόμενο ενός καταχωρητή γενικού σκοπού, και σε περίπτωση ισότητας, το ανταλλάσσει με το περιεχόμενο ενός δεύτερου καταχωρητή γενικού σκοπού. Αν ο

πολυεπεξεργαστής είναι βασισμένος σε επεξεργαστές αρχιτεκτονικής MIPS, να γράψετε μια συνάρτηση σε συμβολική γλώσσα MIPS που να υλοποιεί τη λειτουργία compare&swap με τη βοήθεια των εντολών LL και SC.

Απάντηση

Ένα από τα σημαντικότερα θέματα παράλληλης επεξεργασίας είναι ο συγχρονισμός μεταξύ των κωδίκων που εκτελούνται παράλληλα. Ο συγχρονισμός αυτός επιτυγχάνεται με κατάλληλη υποστήριξη από το υλικό, με τη μορφή λειτουργιών που ενεργούν ατομικά ή αδιάσπαστα. Τέτοιες λειτουργίες επιτρέπουν σε ένα κέντρο επεξεργασίας να ελέγχει και να τροποποιεί κάποια θέση μνήμης σε μια ενιαία λειτουργία, χωρίς δηλαδή να υπάρχει ο κίνδυνος μεταξύ του ελέγχου και της τροποποίησης να έχει μεσολαβήσει επέμβαση – δηλαδή προσπάθεια εγγραφής – κάποιου άλλου κέντρου επεξεργασίας. Οι αδιάσπαστες λειτουργίες χρησιμοποιούνται τυπικά για απόκτηση κλειδιών σε υλοποίηση φραγμάτων (barriers) ή αμοιβαία αποκλειστικής εκτέλεσης κρίσιμου κώδικα (mutual exclusion).
Η εξασφάλιση της ατομικότητας γίνεται από το υλικό, συνήθως με έναν από δύο τρόπους:

- είτε μέσα στην ίδια εντολή μέσω συνεργασίας του επεξεργαστή με το μέσο διασύνδεσης,
- είτε με ζεύγος εντολών που εξομοιώνει την ατομικότητα μέσα στον επεξεργαστή.

Στην πρώτη περίπτωση, το μέσο διασύνδεσης πρέπει να εμποδίζει άλλες λειτουργίες να ενεργούν στη διεύθυνση μνήμης όπου ενεργεί η αδιάσπαστη λειτουργία, πιθανά με πλήρη αποκλεισμό του μέσου από άλλους επεξεργαστές μέχρι την ολοκλήρωση της λειτουργίας. Για την υλοποίηση με αυτόν τον τρόπο απαιτείται συνεργασία και του μηχανισμού συνοχής της κρυφής μνήμης, κάτι που αυξάνει την πολυπλοκότητα του πρωτοκόλλου συνοχής. Ο πλήρης αποκλεισμός του μέσου διασύνδεσης διευκολύνει την υλοποίηση του μηχανισμού συνοχής, αλλά βέβαια έχει ως αποτέλεσμα μεγαλύτερους χρόνους προσπέλασης μνήμης όταν εκκρεμούν λειτουργίες συγχρονισμού.
Στη δεύτερη περίπτωση, η ατομικότητα δεν εξασφαλίζεται από το μέσο διασύνδεσης, το οποίο έτσι δεν επιβαρύνεται με τέτοιες λειτουργίες, όμως ο επεξεργαστής πρέπει να παρουσιάζει στον κώδικα μια συμπεριφορά που να είναι ισοδύναμη με αυτή των πραγματικά αδιάσπαστων λειτουργιών. Εφ' όσον το μέσο διασύνδεσης δε συμμετέχει, δεν μπορεί να εξασφαλιστεί ότι κάποιος άλλος επεξεργαστής δε θα προσπαθήσει να τροποποιήσει τη θέση μνήμης όπου εκκρεμεί μια τέτοια λειτουργία, κι επομένως ο επεξεργαστής που εξέδωσε τη λειτουργία πρέπει να παρακολουθεί, ώστε να την ακυρώσει αν υπάρξει μεσολάβηση από άλλον επεξεργαστή. Ο μηχανισμός παρακολούθησης υλοποιείται συνήθως με δύο εντολές, μία που εκδίδει τη λειτουργία, και μία που ελέγχει την ολοκλήρωσή της.

Τα συστήματα πολυεπεξεργασίας που βασίζονται στην αρχιτεκτονική MIPS υλοποιούν τη δεύτερη από τις πιο πάνω τεχνικές για ατομικές λειτουργίες, με το ζεύγος εντολών LL και SC. Η πρώτη εντολή διαβάζει τη θέση μνήμης, σηματοδοτώντας την έναρξη της αδιάσπαστης λειτουργίας, ενώ η δεύτερη την τροποποιεί, ολοκληρώνοντας την αδιάσπαστη λειτουργία, με την προϋπόθεση ότι δεν έχει μεσολαβήσει άλλη προσπάθεια εγγραφής στην ίδια θέση. Αν έχει συμβεί κάτι τέτοιο, η εντολή SC επιστρέφει τιμή 0, υποδεικνύοντας την αποτυχία του συστήματος στην προσπάθεια εκτέλεσης των δύο εντολών ως μία αδιάσπαστη λειτουργία. Διαφορετικά, μη μηδενική τιμή επιστροφής σημαίνει ότι η λειτουργία είχε επιτυχία, ότι δηλαδή δεν υπήρξε επέμβαση άλλου επεξεργαστή μεταξύ της εκτέλεσης των δύο εντολών. Τυπικά, κάποια εντολή διακλάδωσης ελέγχει το αποτέλεσμα της SC, και οδηγεί σε επανάληψη της προσπάθειας, αν η τιμή επιστροφής είναι 0.
Μια υλοποίηση της λειτουργίας compare&swap με τη βοήθεια του ζεύγους εντολών LL και SC, υποθέτοντας ότι αρχικά η διεύθυνση μνήμης λαμβάνεται από τον καταχωρητή $4, η τιμή προς σύγκριση από τον καταχωρητή $5 και η τιμή προς ανταλλαγή από τον καταχωρητή $6, είναι η ακόλουθη:

```
CSwp: ll $8,0($4)
      bne $8,$5,Cswp
```

```
or $7,$6,$0
sc $7,0($4)
beq $7,$0,CSwp
```

όπου: Η πρώτη εντολή διαβάζει τη θέση μνήμης όπου θέλουμε να εκτελέσουμε την αδιάσπαστη λειτουργία. Η δεύτερη εντολή συγκρίνει την τιμή που διαβάζεται με το περιεχόμενο του καταχωρητή $5, και επανεκκινεί τη λειτουργία αν οι τιμές είναι διαφορετικές. Η τρίτη εντολή αντιγράφει την τιμή προς αποθήκευση από τον καταχωρητή $6 σε κάποιον άλλο καταχωρητή – έστω τον $7, επειδή η εντολή αποθήκευσης SC αλλάζει τον καταχωρητή που περιέχει την τιμή προς αποθήκευση. Η τέταρτη εντολή ολοκληρώνει τη λειτουργία με εγγραφή στη μνήμη του περιεχομένου του καταχωρητή $7. Τέλος, η πέμπτη εντολή εξετάζει το αποτέλεσμα της προηγούμενης, και αν είναι 0, επιστρέφει τον έλεγχο στην πρώτη εντολή.

Παρατηρήστε ότι πιθανή εκτέλεση άλματος στην πρώτη από τις δύο εντολές διακλάδωσης αφήνει την αδιάσπαστη λειτουργία ημιτελή. Αυτό δε μας ενοχλεί, αφού η επόμενη εκτέλεση της εντολής LL θα ξεκινήσει άλλη τέτοια λειτουργία. Αν για παράδειγμα, ο παραπάνω κώδικας είναι μέρος ενός βρόχου αναμονής κάποιου κλειδιού, όσο το κλειδί δεν είναι ελεύθερο, η τιμή που διαβάζεται από τη μνήμη θα οδηγεί σε επιστροφή στην εντολή LL. Μόλις το κλειδί απελευθερωθεί, θα γίνει προσπάθεια εκτέλεσης της SC, και αν αυτή αποτύχει, επειδή κάποιος άλλος επεξεργαστής πρόλαβε και πήρε το κλειδί, τότε θα επιστρέψουμε και πάλι στην εντολή LL, διαφορετικά θα συνεχίσουμε με τον κώδικα που ακολουθεί.

Ο παραπάνω κώδικας θα πρέπει να συμπληρωθεί με κάποιον κώδικα καθυστέρησης, ώστε να εξασφαλιστεί ότι δε θα συμβεί αδιέξοδο στην ταυτόχρονη προσπάθεια πολλών επεξεργαστών να εκτελέσουν εντολή SC στην ίδια θέση μνήμης. Ο κώδικας αυτός θα πρέπει να εισάγει διαφορετική καθυστέρηση σε κάθε επεξεργαστή, ώστε αυτός με τη μικρότερη καθυστέρηση να προλάβει να ολοκληρώσει την αδιάσπαστη λειτουργία πριν την επέμβαση κάποιου άλλου. Ένας καλός μηχανισμός καθυστέρησης είναι ο μηχανισμός της εκθετικής υπαναχώρησης που περιγράφεται στο βιβλίο, και δε θα επαναλάβουμε εδώ. Ο μηχανισμός αυτός αυξάνει δυναμικά με εκθετικό τρόπο το χρόνο καθυστέρησης σε κάθε αποτυχία της εντολής SC, κι έτσι δίνεται η ευκαιρία σε ανταγωνιστές να διεκδικήσουν την ολοκλήρωση της δικής τους αδιάσπαστης λειτουργίας.