

Επέκταση Άσκησης 6.1

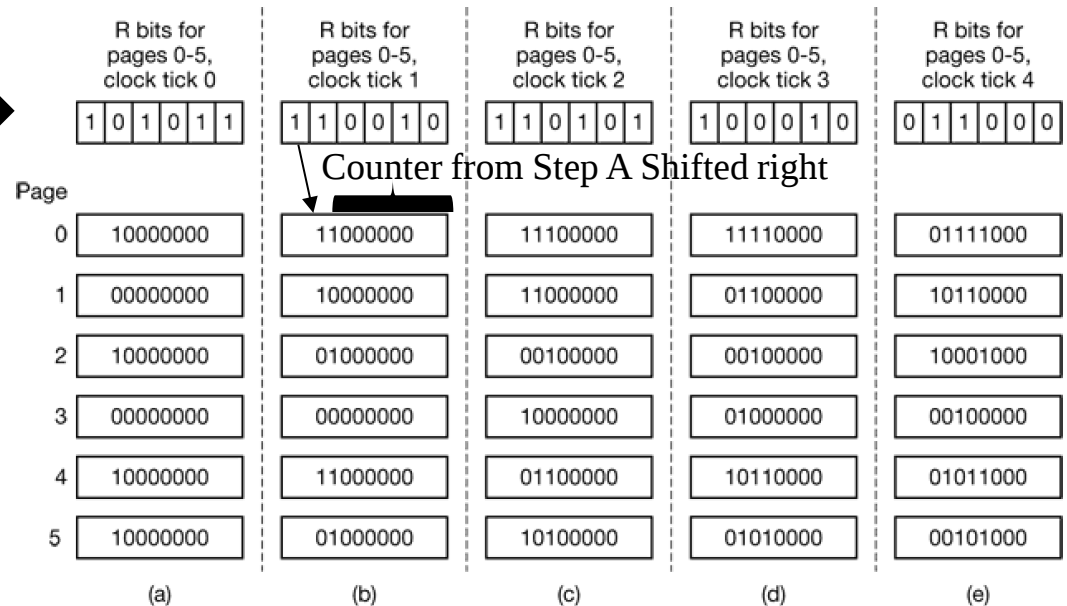
Υλοποίηση Aging λογικής

Aging λογική

- Σε πολλές περιπτώσεις είναι χρήσιμη η συνδυασμένη αριθμητική υλοποίηση/κωδικοποίηση ενός προβλήματος στο πεδίο του χρόνου
- Παράδειγμα από τα Λειτουργικά Συστήματα
 - Πώς δίνουμε προτεραιότητα σε συγκεκριμένα προγράμματα τα οποία έχουν συχνή χρήση κάποιων πόρων (π.χ. μνήμη) έναντι κάποιων άλλων που δεν αναμένεται να έχουν συχνή χρήση
 - Η μνήμη είναι περιορισμένη
 - Κόβεται σε κομμάτια (σελίδες)
 - Κάθε εφαρμογή παίρνει κάποιες σελίδες
 - Αν κάποια στιγμή μια εφαρμογή θέλει παραπάνω μνήμη, το ΛΣ πρέπει να βρει κάποια άλλη («θύμα») για να δώσει στην πρώτη παραπάνω
 - Να δώσει περισσότερες σελίδες στη μία
 - Πώς όμως μπορούμε να κρατάμε εύκολα ένα τέτοιο ιστορικό προσπελάσεων για να δούμε τη συχνότητα χρήσης ενός κομματιού/σελίδας, χωρίς να χρειάζεται να κρατάμε όλη τη πληροφορία προσπελάσεων?
 - Επίσης τα προγράμματα έχουν χωρική και χρονική τοπικότητα. Η χωρική αφορά το κομμάτι της μνήμης που κοιτάμε. Πώς μπορεί να αποτυπωθεί η χρονική εγγύτητα των προσπελάσεων με κάποιο αριθμητικό τρόπο?

Αντικατάσταση σελίδων - Aging

- Για να το κάνουμε αυτό, πρέπει να δίνουμε μεγαλύτερη βαρύτητα στις πιο πρόσφατες προσπελάσεις
- Πώς γίνεται εύκολα υπολογιστικά/αριθμητικά αυτό?
 - Με τα most/least significant bits ενός «μετρητή» ανά σελίδα
 - Σε κάθε χτύπο ρολογιού έχουν σημειωθεί με 1 οι σελίδες στις οποίες έχει γίνει αναφορά στο διάστημα ανάμεσα σε αυτό και το προηγούμενο χτύπο
 - Ο αντίστοιχος μετρητής σελίδας από το προηγούμενο βήμα γίνεται shift προς τα δεξιά
 - Το bit αναφοράς της σελίδας αντιγράφεται στο MSB του μετρητή αυτής της σελίδας (αν έχει γίνει αναφορά στο ενδιαμέσο διάστημα είναι 1)
 - Αυτή η αναφορά επειδή μπαίνει στο MSB έχει και τη μεγαλύτερη βαρύτητα
 - Αν προκύψει ανάγκη αντικατάστασης, διαλέγουμε τη σελίδα με τη μικρότερη αριθμητική τιμή μετρητή



Επέκταση Άσκησης 6.1

- Επεκτείνετε τη λογική της άσκησης 1 ώστε ο χρήστης να δίνει σε ένα while loop έναν αριθμό (0 ή 1) που θα δείχνει αν έγινε προσπάθεια το τελευταίο διάστημα
- Αυτός ο αριθμός θα πρέπει να επηρεάζει (μέσω μιας συνάρτησης) σε κάθε loop το πιο σημαντικό ψηφίο ενός μετρητή και να επιστρέφει τη νέα τιμή στο μετρητή
- Σε κάθε βήμα τυπώνετε τον μετρητή μέσω της display_bits ώστε να επιβεβαιώσετε την αλλαγή
- Σημείωση: ο μετρητής πρέπει να είναι unsigned int
- Συνάρτηση age:
 - unsigned int age(unsigned int counter,int x){