

Νίκος Μασούρας

sdi1800112

Λειτουργικά Συστήματα Άσκηση 2

Το παρών πρόγραμμα προσομοιώνει τη συμπεριφορά συστήματος διαχείρισης μνήμης βάσει πραγματικού ίχνους αναφορών.

Αυτό μπορεί να γίνει μέσω του αλγορίθμου LRU ή του αλγορίθμου 2ης ευκαιρίας. Αρχικά για την εκτέλεση του το πρόγραμμα

χρειάζεται 4 παραμέτρους, τον αλγόριθμο αντικατάστασης, το πλήθος των frames της κύριας μνήμης, το πλήθος των αναφορών

q και το πλήθος των αναφορών που θα εξεταστούν από κάθε αρχείο, με αυτή τη σειρά. Το όρισμα του αλγορίθμου αντικατάστασης

πρέπει να είναι "LRU" ή "2nd" ενώ τα υπόλοιπα είναι αριθμοί. Για τη δημιουργία εκτελέσιμου θα πρέπει να χρησιμοποιηθεί

η εντολή make. Το πρόγραμμα δίνει ως έξοδο τον αριθμό εγγραφών στον δίσκο, τον αριθμό των αναγνώσεων από τον δίσκο, τον

αριθμό των page faults, το πλήθος των καταχωρήσεων που εξετάστηκαν από κάθε αρχείο αναφορών και το μέγεθος της μνήμης.

Για να λειτουργήσει το πρόγραμμα θα πρέπει τα αρχεία αναφορών να ονομάζονται "gcc.trace" και "bzip.trace". Ένα παράδειγμα

δημιουργίας και εκτέλεσης του προγράμματος είναι το εξής:

make

./askisi LRU 200 100 10000

Ο αλγόριθμος LRU αρχικά ψάχνει αν υπάρχει η σελίδα στο hash table και κατ' επέκταση και στη μνήμη. Αν υπάρχει τότε αφού

τροποποιήσει την τιμή read -η οποία συμβολίζει αν η αναφορά είναι R ή W- στο node του hash table τοποθετεί τον ήδη υπάρχοντα

κόμβο μιας απλής συνδεδεμένης λίστας LRU που αντιστοιχεί στην σελίδα στο τέλος αυτής της λίστας. Αν δεν υπάρχει αυτή η σελίδα

στη μνήμη τότε υπάρχει page fault και χρειάζεται να διαβάσει από τον δίσκο. Αν η μνήμη δεν είναι κορεσμένη τότε ο αλγόριθμος

προσθέτει στο hash table που αντιστοιχεί στο συγκεκριμένο αρχείο ένα κόμβο με πληροφορίες για την σελίδα και επίσης προσθέτει

στο τέλος της λίστας LRU που αντιστοιχεί στο αρχείο ένα κατάλληλο κόμβο για τη

σελίδα. Αντίθετα αν η μνήμη είναι κορεσμένη τότε

πρέπει να γίνει page replacement. Η σελίδα που αντικαθίσταται είναι αυτή η οποία υποδεικνύεται από τον πρώτο κόμβο από την αρχή

της λίστας LRU. Έτσι αντικαθίσταται η σελίδα που χρησιμοποιήθηκε τελευταία και ο κόμβος που αντιστοιχεί στη νέα σελίδα θα μπει

στο τέλος της λίστας. Τέλος αν η σελίδα είναι W τότε πρέπει να γραφεί στο δίσκο.

Ο αλγόριθμος δεύτερης ευκαιρίας αν βρει την σελίδα στη μνήμη αλλάζει ένα reference bit r, που βρίσκεται στον κόμβο μιας λίστας clock

και αντιστοιχεί στη σελίδα, αν είναι 0 σε 1 και τοποθετεί τον κόμβο στο τέλος της λίστας clock. Αν δεν υπάρχει η σελίδα στη μνήμη

και αυτή δεν είναι κορεσμένη τοποθετούμε σχετικό κόμβο στο hash table και στο τέλος της λίστας clock με r αχρικοποιημένο στο 0. Αν η

μνήμη είναι κορεσμένη η σελίδα που αντικαθίσταται είναι η πρώτη που βρίσκεται να έχει r bit 0 αρχίζοντας την αναζήτηση από την αρχή της

λίστας ενώ σε κάθε κόμβο που περνάει η αναζήτηση το r bit θέτετε σε 0. Αν για παράδειγματα r bit όλων των κόμβων είναι 1 τότε με το τέλος

της αναζήτησης όλα θα έχουν γίνει 0 και η σελίδα που θα αντικατασταθεί είναι αυτή που υποδεικνύεται από τον κόμβο στην αρχή της λίστας.
Ο κόμβος που αντιστοιχεί στη νέα σελίδα θα μπει στο τέλος της λίστας clock. Η γενικότερη δομή του αλγορίθμου είναι ίδια με του LRU.

Η κύρια μνήμη προσομοιάζεται από έναν απλό πίνακα.