

Λειτουργικά Συστήματα(K22): Δεύτερη Εργασία

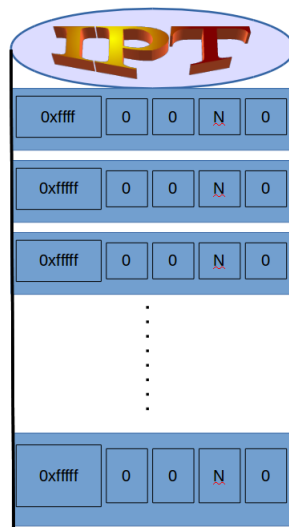
Δημήτρης Φωτεινός

AM:1115201700181

Επεξήγηση κώδικα και παραδοχές:

Η προσομοίωση έχει ως εξής:

Αρχικά, δημιουργούμε το IPT. Να αναφέρουμε πως η κάθε "καταχώρηση" στον ανεστραμένο πίνακα σελίδων περιέχει τα εξής δεδομένα:



- Η πρώτη "φέτα" , περιέχει την διεύθυνση της σελίδας όπου θα γίνεται η αναφορά για την συγκεκριμένη σελίδα. Την αρχικοποιούμε με μια τιμή που δεν περιέχεται σε κανένα απ'τα .trace αρχεία.

- Το επόμενο μέλος της καταχώρησης αναπαριστά τον αναγνωριστικό αριθμό της διεργασίας η οποία επενεργεί και αλληλεπιδρά κάθε φορά με την μνήμη. Ως **παραδοχή**,έχουμε ορίσει πως η πρώτη διεργασία έχει τον αναγνωστικό αριθμό **1** και η δεύτερη διεργασία έχει τον αναγνωριστικό αριθμό **2**. Αρχικοποιούμε τον αριθμό αυτόν με 0.
- Το επόμενο στοιχείο των καταχωρήσεων αυτών αναπαριστά το γνωστό σε όλους μας **valid bit**. Στην ουσία το συγκεκριμένο bit μας "μαρτυράει" αν μια συγκεκριμένη σελίδα βρίσκεται ήδη στην μνήμη. Κάθε φορά λοιπόν,που υπάρχει μια τέτοια ταύτιση αυτό το bit παίρνει την τιμή 1, αλλιώς αν δεν βρίσκεται στην μνήμη έχει την τιμή 0. Θα μας βοηθήσει ιδιαίτερα στην συνέχεια για να υλοποιήσουμε τον αλγόριθμο WS Clock.
- Το επόμενο bit (καθώς πρόκειται κυριολεκτικά για 1 bit), παίζει τον ρόλο της "εγγραφής ή της ανάγνωσης". Αλλάζει δηλαδή ανάλογα με το αν έχουμε R ή W στο τέλος μιας αναφοράς. Και ανάλογα αυξάνονται οι μετρητές στο πρόγραμμα.
- Το τελευταίο bit πρόκειται για την λεγόμενη "χρονοσφραγίδα",δηλαδή είναι ένας μετρητής όπου μετράει (στον LRU) την στιγμή όπου υπήρξε αναφορά σε μια συγκεκριμένη σελίδα στον IPT πίνακα. Στον WS αλγόριθμο αυτή η χρονοσφραγίδα αλλάζει,καθώς δεν αυξάνεται σε κάθε αναφορά μιας συγκεκριμένης σελίδας αλλά όταν υπάρχει μια καινούρια εισαγωγή στον πίνακα μας ή όταν "δίνουμε δεύτερη ευκαιρία" σε μια σελίδα του πίνακα,όπως άλλωστε αναφέρει και το βιβλίο (σελ. 418).

Αξίζει επίσης να δείξουμε την λογική πίσω απ'την επιλογή των 5 πρώτων δεκαεξαδικών ψηφίων απ'το αρχείο των αναφορών.Όπως γνωρίζουμε, κινούμαστε με βάση το 32-bitο σχήμα στη συγκεκριμένη εργασία:

32bit	
5x4=20	3x4=12

Σύμφωνα και με το βιβλίο, η διεύθυνση της σελίδας αναπαριστάται απο 20bits,δηλαδή από 5 bytes. Και για αυτό παίρνουμε τα 5 πρώτα. (Βέβαια,τα 3 τελευταία στοιχεία χρησιμοποιούνται για τον υπολογισμό του πλαισίου στο οποίο θα μπει η σελίδα,στην φυσική μνήμη δηλαδή (offset)). Απ'την στιγμή που κάνουμε εικονική υλοποίηση και δεν έχουμε φυσική μνήμη,το offset αυτό

δεν μας χρησιμεύει κάπου.

Στην συνέχεια,εφαρμόζουμε σε γενικές γραμμές τον αλγόριθμο που βρίσκεται στο βιβλίο σχετικά με την αντικατάσταση σελίδας (σελ. 411).

Οι αλγόριθμοι LRU και WS Clock υλοποιήθηκαν με βάση το βιβλίο. Έγιναν πληθώρα παραδειγμάτων για την πιστοποίηση της λειτουργικότητας και των 2 αλγορίθμων.Τέλος να σημειώσουμε πως όλα τα στατιστικά εμφανίζονται στην main εκτός απ'τις καταχωρήσεις απ'το αρχείο αναφορών,οι οποίες και βρίσκονται στο .txt αρχείο: Entries.txt.

Παραδοχές και μεταγλώττιση του προγράμματος

Για να μεταγλωττιστεί το πρόγραμμα γράφουμε διαδοχικά:

- make

1 Και είτε επιλέγουμε τον LRU, π.χ: LRU frames q MaxReports

2 Είτε επιλέγουμε τον WS, π.χ WS frames q MaxReports WindowSize

Έχω υλοποιήσει 2 αλγόριθμους για το WS:

- 1 Ο πρώτος είναι η απλή εκδοχή του Working Set και έγινε με βάση με τον αλγόριθμο στην παρακάτω φωτογραφία:

- **WS:** Αν ο αλγόριθμος αντικατάστασης που επιλέξαμε είναι ο WS τότε:
 Αρχικά ελέγχεται αν η μνήμη είναι γεμάτη.
 - Αν δεν είναι, τότε σε κάθε εισαγωγή ελέγχουμε αν το #p υπάρχει ήδη στο Page Table.

-
- Αν δεν υπάρχει, τότε το εισάγουμε και στη συνέχεια ενημερώνουμε το Working Set.
 - Αν το #p δεν υπάρχει στο Working Set, τότε:
 - Αν έχουν γίνει ήδη οι πρώτες n εισαγωγές (όπου n = μέγεθος παραθύρου), δημιουργείται page fault. Στην συνέχεια, το #p με το μικρότερο time signature αντικαθιστάται από το νέο και έτσι ενημερώνεται το Working Set.
 - Αν δεν έχουν γίνει οι πρώτες n εισαγωγές, τότε το νέο #p εισάγεται απευθείας στο Working Set.
 - Αν το #p υπάρχει στο Working Set, τότε:
 - Αν έχουν γίνει ήδη οι πρώτες n εισαγωγές (όπου n = μέγεθος παραθύρου), διαγράφεται το #p με το μικρότερο time signature και στην συνέχεια αν το νέο #p εξακολουθεί να βρίσκεται στο Working Set, δεν ξαναισάγεται. Σε αντίθετη περίπτωση, το νέο #p εισάγεται στο Working Set.
 - Αν δεν έχουν γίνει οι πρώτες n εισαγωγές, τότε το νέο #p δεν ξαναισάγεται στο Working Set.
 - Αν το #p υπάρχει στο Page Table, τότε απλά κάνουμε ενημέρωση του Working Set όπως περιγράφηκε παραπάνω.
 - Αν η μνήμη είναι γεμάτη, τότε διαγράφονται από τη μνήμη όλες οι σελίδες που δεν βρίσκονται στο Working Set.

2 Ο δεύτερος είναι το WS Clock (δεν έχω βάλει την παράμετρο του Window Size μέσα).

Επίσης υπάρχει και ένα δοκιμαστικό .trace αρχείο μέσα σε κάθε υλοποίηση.

Εν κατακλείδι, έχω 3 φακέλους, ο ένας είναι για το απλό WS (αυτή η εκδοχή επιλέγω να εξεταστεί), ο άλλος για το WSClock και ο τελευταίος είναι μια δοκιμαστική main.c στην οποία υλοποιείται ένα συγκεκριμένο παράδειγμα με το δοκιμαστικό .trace. (./main WS 4 20 30 2)

Ταύτιση έχουμε μόνο όταν μια σελίδα που πάει να εξετάσει μια διεργασία αν υπάρχει στο IPT, υπάρχει ήδη στο IPT, με το **ίδιο** pid (τοπική αναζήτηση).

