

Λειτουργικά Συστήματα - 2η εργασία

Μαραζιάρης Χαράλαμπος - sdi1800105

Ιανουάριος 2020

1 Παραδοτέο

Έχουν υλοποιηθεί όλα τα ζητούμενα της εργασίας. Ο πηγαίος κώδικας είναι γραμμένος στη γλώσσα C και περιλαμβάνει εκτενή σχολιασμό. Το παραδοτέο αρχείο *.tar* αποτελείται από την ακόλουθη ιεραρχία αρχείων:

- *./memory/*
 - * *memory.c* , *memory.h* , *memory_structs.h*
Συναρτήσεις διαχείρισης και πρόσβασης του Memory Segment.
 - * *ipt_management.c* , *ipt_management.h*
Συναρτήσεις χειρισμού του Inverted Page Table.
- *./page_repl_algorithms/*
 - * *page_repl.c* , *page_repl.h*
Υλοποιήσεις των αλγορίθμων αντικατάστασης σελίδας, Least Recently Used (LRU) και Working Set (WS).
- *./queue/*
 - * *queue.c* , *queue.h* : Υλοποίηση απλά συνδεδεμένης ουράς.
- *./traces/** : Αρχεία με ίχνη αναφορών
- *./simulator.c* : Ο πυρήνας του προσομοιωτή.
- *./Makefile* : Εύκολο compile, run, clean.

2 Μεταγλώττιση και Εκτέλεση

Μεταγλώττιση:

```
1 $ make
```

Εκτέλεση: Τα 3 πρώτα ορίσματα είναι υποχρεωτικά.

```
1 $ ./mem_sim <page_replacent_algorithm> <frames> <q> <window_size>  
    <max_references>
```

Διαγραφή αντικειμενικών και εκτελέσιμου:

```
1 $ make clean
```

3 Λειτουργία

Το πρόγραμμα προσομοιώνει τη λειτουργία ενός συστήματος διαχείρισης μνήμης με βάση 2 ίχνη αναφορών.

Τα ορίσματα της γραμμής εντολών καθορίζουν:

1. τον αλγόριθμο αντικατάστασης σελίδας (επιλογή ανάμεσα σε “LRU” και “WS”)
2. τον αριθμό πλαισίων μνήμης (**frames**)
3. το σύνολο εγγραφών που θα διαβάζονται εναλλάξ από κάθε αρχείο (**q**)
4. το μέγιστο μέγεθος του ιστορικού παραθύρου αναφορών, εφόσον έχει επιλεγεί ο αλγόριθμος WS (**window_size**)
5. το μέγιστο πλήθος αναφορών σε σελίδες που θα διαβαστούν συνολικά και από τα 2 αρχεία (**max_references**). Αν δεν καθοριστεί, τότε διαβάζονται όλες οι αναφορές.

Θεωρούμε ότι όλες οι σελίδες στις οποίες γίνονται οι αναφορές, βρίσκονται αρχικά στο σκληρό δίσκο (**Hard Drive**).

Αφού αρχικοποιηθεί το memory segment, διαβάζονται q στο πλήθος αναφορές από κάθε αρχείο-ίχνος και για κάθε μια γίνεται απόπειρα ανάκτησης της σχετικής σελίδας από την εικονική μνήμη. Η εικονική μνήμη αναπαρίσταται από έναν *Αντίστροφο Πίνακα Σελίδων* (**Inverse Page Table**). Αρχικά, η σελίδα αναζητείται στο IPT και εάν βρεθεί, ενημερώνονται οι πληροφορίες που τη συνοδεύουν. Διαφορετικά, φορτώνεται από τον σκληρό δίσκο (**Page Fault**).

Εάν το IPT **δεν** είναι γεμάτο, τότε η αναφορά αποθηκεύεται εκεί. Σε αντίθετη περίπτωση, καλείται ο αλγόριθμος αντικατάστασης (ορισμένος στη γραμμή εντολών) προκειμένου να δημιουργήσει χώρο στο IPT, ώστε να τοποθετηθεί η πιο πρόσφατη σελίδα. Εάν οι σελίδες που απομακρύνονται έχουν τροποποιηθεί (δηλαδή έχουν γίνει **Write** σε κάποια αναφορά), τότε γράφονται (ξανά) στον σκληρό δίσκο.

Μόλις τελειώσει η διαδικασία ανάκτησης σελίδων από τη μνήμη, εκτυπώνονται μερικά στατιστικά για τη χρήση της και το πρόγραμμα **τερματίζει**.

4 Αλγόριθμοι Αντικατάστασης Σελίδας

4.1 Least Recently Used (LRU)

Ο αλγόριθμος *LRU* επιλέγει και διαγράφει από την κύρια μνήμη την σελίδα στην οποία δεν έγινε αναφορά για το μεγαλύτερο χρονικό διάστημα, από τις σελίδες που βρίσκονται ήδη στο IPT. Αυτό επιτυγχάνεται διατηρώντας μια πληροφορία, τη χρονική στιγμή τελευταίας αναφοράς, σε κάθε καταχώρηση της κύριας μνήμης.

4.2 Working Set (WS)

Ο αλγόριθμος *WS* διατηρεί για κάθε διεργασία ένα ιστορικό παράθυρο αναφορών, μεγέθους **window_size**, με τις πιο πρόσφατες αναφορές μνήμης της διεργασίας. Όταν το ιστορικό παράθυρο γεμίσει και γίνει μια καινούργια αναφορά από τη διεργασία, τότε αφαιρείται η παλαιότερη χρονικά αναφορά, ώστε να προστεθεί η πιο πρόσφατη (*window roll*). Για τη λειτουργία αυτή έχει χρησιμοποιηθεί η δομή της **ουράς**.

Όταν η κύρια μνήμη είναι πλήρης, και έρθει η στιγμή της αντικατάστασης σελίδων, τότε δημιουργείται ένα σύνολο με όλες τις **διακριτές** σελίδες που βρίσκονται στο ιστορικό παράθυρο αναφορών της διεργασίας. Στη συνέχεια, **διαγράφονται** από το IPT όσες σελίδες της διεργασίας δεν περιλαμβάνονται στο σύνολο που μόλις δημιουργήθηκε, ελευθερώνοντας χώρο για την πιο πρόσφατη σελίδα.