

ΛΕΙΤΟΥΡΓΙΚΑ ΑΡΤΙΟΙ-project 1

Γάγγας Δημήτρης

: 1115201400024

Η εργασία υλοποιήθηκε σε linux ubuntu 16.04 64bit

Αρχεία που δημιουργήθηκαν για τις ανάγκες της άσκησης:

- **w1-5.c**
Μικρά προγράμματα που υλοποιήθηκαν που δημιουργούν φόρτο εργασίας.
Προσοχή! Θέλουν πάνω από 5-6 λεπτά.
- **Atext.txt**
Text αρχείο που δε ξεπερνά τα 50kbytes.
- **minHeapFor_xv6.c/.h**
Στην οποία βρίσκονται όλες οι συναρτήσεις που χρειάστηκαν να υλοποιηθούν για το min heap μας.
Συγκεκριμένα:
 1. `initminHeap ()`;
Αρχικοποίηση τον αριθμό των κόμβων του heap σε 0
 2. `getLeft (int index),getRight (int index),getParent (int index)`
Που επιστρέφουν $2*v+1$, $2*v+2$ και $(v-1)/2$ αντίστοιχα, όπου v ο αριθμός των κόμβων στο heap.
 3. `mHeap_push (struct proc * p) , pop ()`
Η `mHeap_push` καλείται στα 5 σημεία στο `proc.c` που το state γίνεται `RUNNABLE`.
Η `pop` καλείται στον `scheduler` αφού πρώτα ξέρουμε ότι το heap δεν είναι άδειο από τη `getHeapSize()`.
 4. `heapyfy_push (int index) , heapyfy_pop (int index)`
Καλούνται μέσα στις `mHeap_push` και `pop` αντίστοιχα για να κάνουν update τον σωρό μας με βάση τα `priorities`(Υπολογίζεται για κάθε process στην αρχή της `push`).
 5. `getHeapSize()`

Αρχεία που συμπληρώθηκαν για τις ανάγκες του minHeap technique στο λειτουργικό xv6:

- **proc.c**

1. Ο scheduler πλέον δε χρησιμοποιεί τη τεχνική round robin αλλά τα priorities με βάση το policy που μας υποδείχθηκε.
2. Στη allocproc γίνεται η αρχικοποίηση των πεδίων της proc(creation_time,running_time,priority,scheduledTimes etc.)
3. Στην pinit καλείται η initminHeap();
4. Δημιουργήθηκε η int printProcess(void) η οποία καλείται απ 'τη system call που φτιάξαμε και εκτυπώνει τα SLEEPING ,RUNNING,RUNNABLE processes .
**Για τα RUNNABLE processes εκτυπώνεται και το priority.
** Επίσης, εκτυπώνεται και το #Πλήθος μη ολοκληρωμένων time slices ,ο οποίος υπολογίζεται απ 'τη διαφορά timesScheduled- runningTime

- **Proc.h**

Προστέθηκαν τα απαιτούμενα πεδία για τις ανάγκες της άσκησης.
(creation_time,running_time,priority,scheduledTimes).

- **trap.c**

Πριν καλεστεί η yield που θα διώξει την process απο τη CPU λόγω τέλους time slice το running time του θα αυξηθεί κατά 1.

- **Makefile**

- **Τα αρχεία που χρειάστηκαν για την υλοποίηση της system call (sys_printp(void))**

Σχετικά με το Παραδοτέο-Σχολιασμοί

Όλα τα δείχνουν να λειτουργούν σωστά.