Καραισκου Μαρια Ελεννη 1103077 up1103077@ac.upatras.gr

Παναγοπουλου Βασιλικη 1100651 up1100651@ac.upatras.gr

Βιτσιου Ελενη 1100502 [up1100502@ac.upatras.gr](mailto:up1100502@ac.upatras.gr)

Μποζικη Φαιδρα 1100638 up1100638@ac.upatras.gr

**Πρώτη Φάση: υποστήριξη πολλαπλών επεξεργαστών**

**1.1 Εξυπηρέτηση με βάση τη σειρά άφιξης (FCFS)**

1.1.1 Περιγραφή Σχεδιασμού.

Διατήρησα τον ίδιο κώδικα με αυτόν που μας δώσατε, με μερικές αλλαγές.

* struct proc\_desc: Πρόσθεσα την μεταβλητή num\_procs η οποία αποθηκεύει τον αριθμό των επεξεργαστών που χρειάζεται η κάθε διεργασία για να εκτελεστεί.
* main: Στην ανάγνωση από την γραμμή εντολών, διαβάζει και τον αριθμό των διαθέσιμων επεξεργαστών.
* FCFS:
* Δέχεται ως παράμετρο τον αριθμό των διαθέσιμων επεξεργαστών (numprocs).
* Χρησιμοποιήσαμε δυναμική διαχείριση μνήμης για την παρακολούθηση των διεργασιών(processors ) και την κατάσταση του επεξεργαστή (processor\_status , ελεύθερος ή κατειλημμένος).
* Έπειτα εφόσον υπάρχουν διαθέσιμοι επεξεργαστές, τους αναθέτει διεργασίες. Όταν γεμίσουν όλοι οι επεξεργαστές οι διεργασίες παραμένουν στην αναμονή μέχρι να βρεθεί πάλι κάποιος ελεύθερος επεξεργαστής.

1.1.2 Παραδείγματα Υλοποίησης.

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, κατάλογος, γραμματοσειρά

Περιγραφή που δημιουργήθηκε αυτόματα

Παραδειγμα εκτελεσης προγραμματος με έναν απαιτουμενο επεξεργαστη.

1.1.3 Δυσκολίες.

* Δεν αντιμετώπισα κάποιο ιδιαίτερο πρόβλημα κατά την συγγραφή του κώδικα. Μου πήρε περισσότερο χρόνο να σκεφτώ σαν ιδέα πως θα το υλοποιήσω, παρά να γράψω τον κώδικα.

**Δεύτερη Φάση: υποστήριξη αιτήσεων πολλαπλών επεξεργαστών**

2.1 Τεκμηρίωση Κώδικα.

Η δεύτερη φάση σχεδιασμού του FCFS έχει υλοποιηθεί έτσι ώστε να δέχεται διεργασίες με πολλούς επεξεργαστές από ένα text file. Το πρόγραμμα έχει φτιαχτεί έτσι ώστε να εκτελούνται όσες διεργασίες χωράνε στους επεξεργαστές και στην συνέχεια αποδεσμεύονται όλοι για να ξεκινήσει η εκτέλεση των επομένων διεργασιών.

2.2 Περιγραφή Σχεδιασμού.

Το πρόγραμμα αποτελείται από δυο βασικές συναρτήσεις την main και την fcfs μεταξύ άλλων.

* Η συνάρτηση fcfs έχει υλοποιηθεί σε ένα loop μέχρι η λίστα διεργασιών να αδειάσει. Ελέγχει αν υπάρχουν διαθέσιμοι επεξεργαστές και διεργασίες που δεν έχουν εκτελεστεί και ακολουθεί ο παρακάτω κώδικας . Χρησιμοποιείται η συνάρτηση fork() έτσι ώστε όταν το Pid=0 να δημιουργηθεί μια δεύτερη διαδικασία όπου αντικαθιστά την τρέχουσα διεργασία με μια νέα εκτελέσιμη. Στην συνέχεια αναζητάμε τις ολοκληρωμένες διεργασίες με ένα Loop, ενημερώνεται η κατάσταση της διεργασίας και καταγράφονται τα στοιχεία ολοκλήρωσης της. Τέλος, επιστέφει τους δεσμευμένους επεξεργαστές για να δεχτούν τις επόμενες διεργασίες προς εκτέλεση.
* Η συνάρτηση main() έχει τροποποιηθεί έτσι ώστε στο αρχείο διεργασιών να διαβάζονται οι απαιτούμενοι επεξεργαστές της κάθε διεργασίας και να τις αποθηκεύονται σε μια μεταβλητή processors\_needed. Επίσης δέχεται ως όρισμα τον διαθέσιμο αριθμό επεξεργαστών και τον αποθηκεύει στην μεταβλητή total\_processors.

2.3 Παραδείγματα Υλοποίησης.

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

Παράδειγμα εκτέλεσης προγράμματος

όταν οι διεργασίες 3 και 7 απαιτούν δυο επεξεργαστές.

2.4 Δυσκολίες.

Αντιμετωπίσαμε δυσκολίες στην υλοποίηση του ερωτήματος καθώς την τελευταία φορά που τρέχει το while της συνάρτησης fcfs η τελευταία διεργασία διαγραφόταν από την λίστα και έτσι δεν μπορούσαμε να ανακτήσουμε τις πληροφορίες εκτέλεσης της.