

## 2<sup>η</sup> Εργασία στα Λειτουργικά Συστήματα

Το αντικείμενο της δεύτερης εργασίας αφορά στην προσομοίωση ενός βραχυπροθέσμου δρομολογητή που θα προωθεί διεργασίες προς τη Cpu , καθώς και την καταγραφή καποιων στοιχειωδων στατιστικων στοιχειων χρησης του.Ο προσομοιωτης θα περιεχει εναν κυριως βρογχο μεσα στον οποιο θα υλοποιειται ο αλγοριθμος RR με διαφορετικο quantum για την πρωτη ροη εκτελεσης ( $q=3$ ) και διαφορετικο για την δευτερη ( $q=9$ ).

Η Δομή του κωδικα αποτελείται απο διαφορες κλασεις.

- 1) Class NewProcessTemporaryList : Αυτη η κλαση αναπαριστα μια λιστα στην οποια τοποθετουνται νεες διεργασίες που μόλις έχουν δημιουργηθεί και βρίσκονται στην κατάσταση new.
- 2) Class CPU : Σαυτη την κλαση αναπαρισταται η μοναδα επεξεργασίας του συστηματος
- 3) Class Process :Η κλαση αυτη αναπαριστα μια διεργασία του συστηματος
- 4) Class ProcessGenerator: Η συγκεκριμενη κλαση αναπαριστα μια γεννητρια διεργασιων για την προσομοίωση
- 5) Class Clock : Κλαση που αναπαριστα το ρολοι του συστηματος
- 6) Class ReadyProcessesList : Λιστα στην οποια τοποθετουνται οι ετοιμες διεργασίες.Υλοποιειται με μια ουρα FIFO .
- 7) Class Statistics : Αυτη η κλαση υπολογιζει ορισμενα στατιστικα στοιχεια βασει των διεργασιων που εμφανιζονται στο συστημα και τα αποθηκευει σε ενα αρχειο
- 8) Class RRScheduler: Η κλαση αυτη αναπαριστα τη λειτουργια του δρομολογητη διεργασιων του συστηματος, μεταφεροντας διεργασίες απο και προς τη Cpu.Προσθετει διεργασίες στη λιστα ετοιμων διεργασιων συμφωνα με τον αλγοριθμο RR.

## ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΕΡΓΑΣΙΑΣ

Ο κωδικας ξεκιναι ελεγχοντας αν υπαρχει το αρχειο με τα δεδομενα των διεργασιων.

Εαν δεν υπαρχει τοτε δημιουργω ενα αρχειο και ενα αντικειμενο processgenerator οπου καλω την συναρτηση createprocess() και εκει μεσα με την κλαση Random δημιουργω ψευδοτυχειους χρονους αφιξης και χρησης του επεξεργαστη και τα αποθηκευω σε ενα process.(τα τυχαια νουμερα τα εχω βαλει στο διαστημα [1-20]).Η διαδικασια αυτη θα επαναληφθει 5 φορες αφου εχω δωσει ως αριθμο συνολικων διεργασιων το 5.(Δεν υπηρχε καποια διευκρινιση στην εκφωνηση) Αυτες οι διεργασίες με την σειρα τους θα αποθηκευτουν σε μια λιστα της class NewProcessGenerator οπου σαυτη την λιστα αποθηκευονται διεργασίες με κατασταση new.Επισης δημιουργω ενα αντικειμενο της RRScheduler με παραμετρο 3 οπου ειναι η τιμη του quantum.

Εαν υπάρχει το αρχείο τότε δημιουργώ ένα αντικείμενο της RRScheduler με παραμετρο 9 όπου είναι η τιμή του quantum. Ανοίγω το αρχείο και καλώ την συνάρτηση parseprocessfile() όπου διαβάζω μια μια τα χαρακτηριστικά της κάθε διεργασίας, φτιάχνω 5 διαφορετικά αντικείμενα Process και τα αποθηκεύω στην λίστα της κλάσης NewprocessTemporaryList.

Στην συνέχεια υπάρχει ο κυρίως βρόγχος του προγράμματος όπου εκτελείται ο αλγόριθμος RR. Εαν οι χτυποι του ρολογιού περάσουν το 2000 (μία τυχαία τιμή όπου είναι σίγουρο πως θα έχουν τελειώσει όλες οι διεργασίες για τα τυχαία χαρακτηριστικά που τους έχω δώσει) και εαν έχουν τελειώσει όλες οι διεργασίες τότε σταματάει ο βρόγχος.

Ο Αλγόριθμος RR:

- 1) Στην αρχή γίνεται έλεγχος για το αν ο χρόνος της τελευταίας διακοπής στον επεξεργαστή είναι ίσος με το ρολόι. Αν είναι ίσος τότε γίνεται έλεγχος αν η διεργασία έχει τερματίσει. Αν έχει τερματίσει (current state=3) τότε γίνεται ο υπολογισμός του waiting time και του turnaround time. Αν δεν έχει τερματίσει τότε η διεργασία ξαναμπάνει στο τέλος της ουράς του readylist.
- 2) Αν ο χρόνος της τελευταίας διακοπής στον επεξεργαστή δεν είναι ίσος με το ρολόι τότε συνεχίζω παρακάτω όπου γίνεται ακόμα ένας έλεγχος αυτή τη φορά για τον αν ο χρόνος αφίξης από της διεργασίες που βρίσκονται σε κατάσταση new είναι ίσος με το ρολόι. Αν είναι τότε εκχωρώ την διεργασία στην readylist και αλλάζω την κατάσταση της σε waiting.
- 3) Στην συνέχεια του προγράμματος γίνεται ένας έλεγχος σχετικά με το μέγεθος της readylist. Βρίσκω το μεγαλύτερο μέγεθος που μπορεί να πάρει.
- 4) Τέλος γίνεται ακόμα ένας έλεγχος για το αν ο χρόνος διακοπής της τελευταίας διεργασίας από τον επεξεργαστή είναι ίσος ή μικρότερος από το ρολόι. Αν ισχύει αυτό, τότε σημαίνει ότι ο επεξεργαστής είναι ελεύθερος και μπορώ να δρομολογήσω την επόμενη διεργασία. Το κάνω αυτό και η διεργασία που είναι στην αρχή της ουράς φεύγει και πάει στον επεξεργαστή.

Αυτός ο αλγόριθμος βρίσκεται σε έναν βρόγχο και εκτελείται συνέχεια μέχρι να φτάσει το ρολόι σε ένα συγκεκριμένο χρονικό σημείο όπου όλες οι διεργασίες θα έχουν τελειώσει.

Waiting time: Όταν έχει τερματιστεί μια διεργασία τότε αφαιρώ από την ώρα που έχει το ρολόι εκείνη την στιγμή, το χρόνο αφίξης της διεργασίας και τον χρόνο που βρισκόταν η διεργασία μέσα στην cpu.

$WaitingTime = Clock - arrivalTime - quantum * (\text{ποσες φορές μπήκε})$

Το ποσες φορές μπήκε το υπολογίζω με έναν counter (int c) που βάζω στον επεξεργαστή και κάθε φορά που μπαίνει αυξάνεται κατά ένα.

TurnAroundTime: Όταν έχει τερματιστεί μια διεργασία τότε αφαιρώ από την ώρα που έχει το ρολόι εκείνη την στιγμή, το χρόνο αφίξης της διεργασίας

$TurnAroundTime = Clock - arrivalTime$

ResponsesTime: Όταν μια διεργασία μπαίνει για πρώτη φορά στον επεξεργαστή (αν ο counter (int c) δηλαδή που τον έχω για αυτό τον σκοπό είναι ίσος με 0) τότε αφαιρώ από την ώρα του ρολογιού εκείνης της στιγμής, το arrivalTime

$ResponseTime = Clock - arrivalTime$

Για να βρω όλα τα παραπάνω ως μεσου χρόνους όλων των διεργασιών απλά τα προσθέτω όλα και τα διαιρώ με το συνολικό αριθμό των διεργασιών.

Πχ .  $AverageWaitingTime = (waitingTime + waitingTime + \dots) / numberOfTotalProcesses$

Εγγραφή Στατιστικών στο αρχείο: Αφού βγω έξω από τον κύριο βρόχο φτιάχνω ένα αντικείμενο Statistics και περνάω το όνομα του αρχείου που θέλω να ανοίξω. Στην συνέχεια καλώ την WriteStatistics2File και γράφω όλα τα αποτελέσματα με τους χρόνους που έχω βρει, στο αρχείο εξόδου.

Την πρώτη φορά που θα τρέξει το πρόγραμμα θα τρέξει με quantum=3 και την δεύτερη με quantum=9. Οπότε τα στατιστικά την πρώτη φορά θα είναι διαφορετικά από την δεύτερη.

Τα αρχεία με τα δεδομένα και τα στατιστικά δεν τα στέλνω. Αυτό είναι ένα παράδειγμα που έτρεξα εγώ.!!!

Example of the program.

Το αρχείο με τα δεδομένα. Το πρώτο στοιχείο είναι ο χρόνος αφίξης της διεργασίας ενώ το δεύτερο στοιχείο είναι ο χρόνος χρήσης στον επεξεργαστή και το τρίτο στοιχείο είναι για την άλλη διεργασία ο χρόνος αφίξης και παει λεγοντας...

8 13 8 3 1 9 14 10 14 12

Το αρχείο με τα στατιστικά. Το αποτέλεσμα αυτό είναι με την δεύτερη ροή εκτέλεσης

Το quantum της δεύτερης ροής είναι μεγάλο. Όπως φέρεται και στο αποτέλεσμα το waiting time και το turnaroundtime είναι αυξημένα αφού η κάθε διεργασία τρέχει στον επεξεργαστή για αρκετά μεγάλο χρόνο με αποτέλεσμα να δημιουργείται μεγάλη ουρά στην

readylist .Το αποτέλεσμα της πρώτης ροής εκτέλεσης θα είχε μικρότερα αποτελέσματα.(quantum=3)

\*\*\* STATISTICS OF 5 PROCESSES RUNNING IN CPU WITH ALGORITHM RR [QUANTUM=9]  
\*\*\*

Average Waiting Time [20.8 sec ]

Average Response Time [4 sec ]

Maximum Length Of Ready Processes List [3 sec]

Average TurnAround Time [35.2 sec ]