

Χρονοπρογραμματισμός 2

23-08-13
18:25

Στο πρώτο pdf είδαμε πώς μπορούμε να φτιάξουμε το διάγραμμα Gantt. Ας δούμε λίγο όμως και τι κάνει ο κάθε αλγόριθμος χρονοπρογραμματισμού.

1. **SJF (Shortest Job First)**. Όπως λέει και το όνομά του, επιλέγεται η διεργασία που της απομένει να εκτελέσει λιγότερος χρόνος. Χωρίζεται σε δύο κατηγορίες, **με διακοπές** και **χωρίς διακοπές**. Όταν λέμε **με διακοπές**, σημαίνει ότι **όταν έρχεται μια νέα διεργασία όσο εκτελείται** μια άλλη, γίνεται έλεγχος σε αυτήν την νέα διεργασία μήπως **έχει μικρότερο χρόνο εκτέλεσης** από αυτήν που είναι μέσα έτσι ώστε **να πάρει την θέση της**. Αντίθετα, όταν έχουμε **χωρίς διακοπές**, πρώτα **τελειώνει ολόκληρη** αυτή που έχει μπει μέσα **και μόνο τότε** γίνεται έλεγχος αν έχουν έρθει άλλες διεργασίες.
2. FCFS (First Come First Served). Με μια απλή μετάφραση, βλέπουμε ότι επιλέγονται οι διεργασίες με την σειρά άφιξής τους. Είναι φυσικά αλγόριθμος χωρίς διακοπή. Κάθε διεργασία που μπαίνει τερματίζει.
3. Με προτεραιότητες. Επιλέγεται η διεργασία με την μεγαλύτερη προτεραιότητα (μικρότερο αριθμό προτεραιότητας). Υπάρχουν κι εδώ οι κατηγορίες με διακοπές και χωρίς διακοπές. Ισχύουν τα ίδια με από πάνω.
4. RR (Round Robin) ή στα ελληνικά, Εκ Περιτροπής. Σε αυτόν τον αλγόριθμο, επιλέγονται όλες οι διεργασίες στην σειρά και τους δίνεται ένα συγκεκριμένο χρονικό διάστημα (κβάντο χρόνου) για να εκτελεστούν. Αν δεν τελειώσουν σε αυτό το κβάντο χρόνου, δίνεται το ίδιο κβάντο χρόνου στην επόμενη τους και μόλις φτάσουμε στην τελευταία, ξανα ξεκινάμε από την πρώτη.

Ας δούμε το διπλανό παράδειγμα.

Αφορά 2 ουρές με RR και SJF χωρίς διακοπές.

Θέμα 3° (20%) Σε ένα σύστημα με χρονοπρογραμματισμό πολυεπίπεδων ουρών φτάνουν πέντε διεργασίες P1, P2, P3, P4, P5 τις χρονικές στιγμές 0, 2, 4, 8, 20 αντίστοιχα. Το σύστημα περιλαμβάνει 2 ουρές τις Α και Β. Στην Α ο χρονοπρογραμματισμός γίνεται με βάση την πολιτική round robin, στην οποία εφαρμόζεται κβάντο χρόνου 4 ms, ενώ στη δεύτερη εφαρμόζεται πολιτική SJF χωρίς διακοπές. Όλες οι διεργασίες που εισάγονται στο σύστημα τοποθετούνται αρχικά στην ουρά Α με βάση την παραπάνω σειρά άφιξης, ενώ επιτρέπεται ανατροφοδότηση μεταξύ των δύο ουρών. Δηλαδή, αν μια διεργασία δεν προλάβει να εκτελεστεί πλήρως στο χρονικό διάστημα που τις δίνει το κβάντο χρόνου, τότε πηγαίνει στην ουρά Β. Όταν η ουρά Α είναι κενή το σύστημα επιλέγει διεργασία από την ουρά Β για εκτέλεση. Οι χρόνοι εκτέλεσης των παραπάνω διεργασιών εκτιμώνται σε 8,20,3,14 και 10 ms αντίστοιχα. Να υπολογίσετε τους μέσους χρόνους αναμονής, απόκρισης και διεκπεραίωσης για τις διεργασίες αυτές. Θεωρείστε ότι ο χρόνος εναλλαγής περιβάλλοντος (context switch) είναι μηδενικός.

Διεργασία	Χρ. Αφίξης	Χρ. Ξεσπάσματος (Εκτέλεσης)
P1	0	8
P2	2	20
P3	4	3
P4	8	14
P5	20	10

Αφού φτιάξαμε το πινακάκι με τα δεδομένα, μπορούμε να ξεκινήσουμε.

Αν θυμάστε, χωρίζουμε το υπόλοιπο της κόλλας μας σε τόσες στήλες όσες είναι οι ουρές που αναφέρονται κρατώντας μια μικρή στήλη ακόμα στα αριστερά για τον χρόνο. Πάμε λοιπόν.

Χρόνος	Ουρά Α (RR 4)	Ουρά Β (SJF χωρίς διακοπές)
0	P1(8)	
4	P1(4) , P2(20), P3(3)	P1(4)
8	P2(16) , P3(3) , P4(14)	P1(4), P2(16)
11	P4(14)	P1(4), P2(16)
15	P4(10)	P1(4), P2(16), P4(10)
19	-----	P2(16), P4(10)
29	P5(10)	P2(16)
33	P5(6)	P2(16), P5(6)
39	-----	P2(16)
55	-----	-----

Διάγραμμα Gantt

	P1	P2	P3	P4	P1	P4	P5	P2	
	0	4	8	11	15	19	29	39	55

Για τους χρόνους αναμονής, ολοκλήρωσης, απόκρισης και ρυθμό διεκπεραίωσης και πώς βγαίνουν, τα είπαμε στο πρώτο pdf. Εδώ απλά θα γράψουμε τα αποτελέσματα.

Ρυθμός διεκπεραίωσης = 5/55

Μέσος χρόνος απόκρισης = [(0-0) + (4-2) + (8-4) + (11-8) + (29-20)] / 5 = 18 / 5 = 3,6

Μέσος χρόνος ολοκλήρωσης = [(19-0) + (55-2) + (11-4) + (29-8) + (39-20)] / 5 = 119 / 5 = 23,8

Μέσος χρόνος αναμονής = [(0-0) + (15-4) + (4-2) + (39-8) + (8-4) + (11-8) + (19-15) + (29-20)] / 5 = 64 / 5 = 12,8

Καλή επιτυχία. Τώρα γνωρίζετε ό,τι χρειάζεστε.

Μια μικρή βοήθεια ακόμα μόνο.

Όταν έχετε αλγόριθμο που έχει να κάνει με την προτεραιότητα την διεργασία να την γράψετε ως εξής για να

Την χρονική στιγμή 0 η μόνη διεργασία είναι η P1. Εκτελείται λοιπόν για το κβάντο χρόνου 4. Συνεπώς θα εκτελεστεί μέχρι την χρονική στιγμή 4. Επειδή δεν προλαβαίνει να εκτελεστεί ολόκληρη, σύμφωνα με την εκφώνηση πρέπει να πάει στην ουρά Β με το νέο της υπόλοιπο. Αφού λοιπόν η P1 έφυγε από την ουρά Α, την χρονική στιγμή 4 υπάρχουν μόνο οι P2 και P3. Σύμφωνα με τον αλγόριθμο θα πρέπει να πάρουν με την σειρά τους από 4 χρονικές μονάδες. Έτσι επιλέγουμε πρώτα την P2 και εκτελείται μέχρι το 8. **Προσοχή!! Δεν μεταφέρουμε και την P3 στην ουρά Β γιατί δεν έχει ακόμα εκτελεστεί.. Σύμφωνα με την εκφώνηση, μεταφέρονται μόνο αυτές που ΔΙΑΚΟΠΤΟΝΤΑΙ..** Μετά με το νέο της υπόλοιπο 16 πηγαίνει στην ουρά Β. Την χρονική στιγμή 8 επιλέγουμε την διεργασία P3 που έχει σειρά για να πάρει το κβάντο χρόνου. **Σημείο προσοχής Νο.2 !! Η P3 δεν εκτελείται μέχρι το 12 μιας και έχει να εκτελέσει μόνο 3 χρονικές μονάδες και άρα δεν χρειάζεται ολόκληρο το κβάντο χρόνου (4) που της δίνεται.**

Την χρονική στιγμή 11 υπάρχει μόνο η P4 την οποία και επιλέγουμε. Δεν κάνουμε το λάθος να νομίζουμε ότι θα εκτελεστεί παραπάνω από 4 επειδή δεν έρχεται άλλη... Ο αλγόριθμος είναι υποχρεωτικός. Θα εκτελεστεί μόνο για 4 μέχρι δηλαδή το 15 και μετά θα περάσει απέναντι στην ουρά Β. Την χρονική στιγμή 15 λοιπόν, η ουρά Α είναι άδεια και πρέπει να επιλέξουμε διεργασία από την ουρά Β με βάση τον αλγόριθμό της. Έτσι μας και η Β έχει SJF χωρίς διακοπές, επιλέγουμε αυτήν με το μικρότερο υπόλοιπο P1(4) και την αφήνουμε να εκτελεστεί μέχρι να τελειώσει. Την χρονική στιγμή 19 που τελειώνει, η ουρά Α εξακολουθεί να είναι άδεια, μιας και δεν έρχεται άλλη διεργασία μέχρι εκείνη την στιγμή. Συνεχίζουμε λοιπόν με την ουρά 2 και επιλέγουμε πάλι τον μικρότερο χρόνο ξεσπάσματος P4(10) που τερματίζει την χρονική στιγμή 29. Την χρονική στιγμή 29, έχει έρθει η P5 που σύμφωνα με την εκφώνηση μπαίνει στην ουρά Α. Συνεπώς μιας και δεν είναι άδεια η Α πρέπει να πάρουμε διεργασία από εκεί. Επιλέγεται η P5 και εκτελείται κι αυτή μόνο για 4 χρονικές μονάδες και μετά μεταφέρεται στην ουρά Β.

Μια μικρή βοήθεια ακόμα μόνο.

Όταν έχετε αλγόριθμο που έχει να κάνει με την προτεραιότητα την διεργασία να την γράφετε ως εξής για να μη μπερδεύεστε.. Ραριθμός(υπολοιπο)^{προτεραιότητα} για παράδειγμα $P1(4)^2$

Τώρα είστε πια εντελώς έτοιμοι..

Συνεπώς μιας και δεν είναι άδεια η Α πρέπει να πάρουμε διεργασία από εκεί. Επιλέγεται η P5 και εκτελείται κι αυτή μόνο για 4 χρονικές μονάδες και μετά μεταφέρεται στην ουρά Β. Από εδώ και πέρα δεν έρχεται άλλη διεργασία για αυτό και επιλέγουμε συνέχεια την μικρότερη διεργασία από την Β και την τερματίζουμε.