



# ΔΙΚΤΥΑ II

## Project Switch 8x8 Simulation

ΓΚΟΥΤΣΙΔΗΣ ΝΙΚΟΛΑΟΣ  
ΑΕΜ: 335

## ΜΕΤΑΓΩΓΕΑΣ 8x8

### Σύντομη περιγραφή

Μεταγωγέας 8x8 που δουλεύει σε slots και σε κάθε slot ένα πακέτο μπορεί να μετακινηθεί από την είσοδο  $i$  στην έξοδο  $j$  αν και μόνο αν την ίδια χρονική στιγμή ούτε η  $i$  ούτε η  $j$  εμπλέκονται σε άλλη μεταφορά πακέτου. Σύμφωνα με τη περιγραφή του προβλήματος στην είσοδο  $i$  κατά την διάρκεια του slot  $t$  έρχονται πακέτα προς μετάδοση

$$A_i = \begin{cases} 0, & 1 - p_i \\ 3, & p_i = a(i \bmod 4) \end{cases}$$

Αναλυτικά για τις εισόδους 1 – 8:

- $A_1 = \begin{cases} 0, & 1 - \alpha \\ 3, & p_1 = a(1 \bmod 4) = \alpha \end{cases}$
- $A_2 = \begin{cases} 0, & 1 - 2\alpha \\ 3, & p_2 = a(2 \bmod 4) = 2\alpha \end{cases}$
- $A_3 = \begin{cases} 0, & 1 - 3\alpha \\ 3, & p_3 = a(3 \bmod 4) = 3\alpha \end{cases}$
- $A_4 = 0, \quad p_4 = a(4 \bmod 4) = 0$
- $A_5 = \begin{cases} 0, & 1 - \alpha \\ 3, & p_5 = a(5 \bmod 4) = \alpha \end{cases}$
- $A_6 = \begin{cases} 0, & 1 - 2\alpha \\ 3, & p_6 = a(6 \bmod 4) = 2\alpha \end{cases}$
- $A_7 = \begin{cases} 0, & 1 - 3\alpha \\ 3, & p_7 = a(7 \bmod 4) = 3\alpha \end{cases}$
- $A_8 = 0, \quad p_8 = a(8 \bmod 4) = 0$

Σύμφωνα με τον εκάστοτε μέγιστο μέσο φόρτο που θα δίνεται, θα πρέπει να υπολογίζεται το  $\alpha$ . Έτσι αν  $\rho$  = φόρτος έχουμε τον τύπο  $\rho = \frac{\sum \lambda_i}{\sum \mu_i}$  όπου  $\lambda_i$  ο ρυθμός αφίξεων στο μεταγωγέα και  $\mu_i$  ο ρυθμός εξυπηρέτησης. Έχουμε  $\lambda_i = E(x) = \sum x * p_x$ ,

έτσι το  $\sum_{i=1}^8 \lambda_i = (1 - \alpha) * 0 + 3 * \alpha + (1 - 2\alpha) * 0 + 3 * 2\alpha + (1 - 3\alpha) * 0 + 3 * 3\alpha + 0 + (1 - \alpha) * 0 + 3 * \alpha + (1 - 2\alpha) * 0 + 3 * 2\alpha + (1 - 3\alpha) * 0 + 3 *$

$3\alpha + 0 = 36\alpha$  και το  $\sum \mu_i = 8$ , άρα  $\rho = \frac{\sum \lambda_i}{\sum \mu_i} = \frac{36\alpha}{8} \Rightarrow \alpha = \frac{8 \cdot \rho}{36}$  και βρίσκουμε το  $\alpha$  για να μπορούμε να υπολογίσουμε τις πιθανότητες αφίξεων πακέτων.

Γίνεται επιλογή για κάθε είσοδο  $i$  τις εξόδους  $j$  στις οποίες θα στέλνει πακέτα και αυτό θα παραμείνει σταθερό καθ' όλη τη διάρκεια της λειτουργίας.

Η επιλογή του προορισμού του πακέτου που αφικνείται γίνεται με τυχαίο ομοιόμορφο τρόπο ανάμεσα στην ομάδα των εξόδων που έχουν προεπιλεγεί.

Μετά με τη μέθοδο maximum weighted matching δημιουργείται ένας permutation matrix για την επιλογή των ζευγών εισόδων-εξόδων για την μετάδοση των πακέτων σε ένα slot.

### **Αναλυτική περιγραφή κώδικα:**

#### Γλώσσα προγραμματισμού C.

Η προσομοίωση και τα πειράματα έγιναν σε εικονική μηχανή Ubuntu 14.10 x64 και compiler τον gcc (Ubuntu 4.8.2-19ubuntu1) 4.8.2.

compile: gcc switch.c -o switch

run: ./switch

Είσοδοι προσομοιώνονται με ουρές - διασυνδεδεμένες λίστες (linked lists) FIFO (First In First Out). Συνολικός αριθμός ουρών 64 (8x8) όλοι οι πιθανοί συνδυασμοί και όλες αυτές οι διασυνδεδεμένες λίστες προσομοιώνονται σε ένα πίνακα δεικτών - διευθύνσεων αυτών, 8x8. Δηλαδή η θέση  $i, j$  «δείχνει» την ουρά με είσοδο  $i$  και έξοδο  $j$ .

Το κάθε πακέτο που αφικνείται προσομοιώνεται σε κόμβο ο οποίος περιέχει το μοναδικό id του πακέτου, τον αριθμό της ουράς που εισάγεται, τον αριθμό της εξόδου που προορίζεται και τον αριθμό του slot που αφιχθεί και έναν δείκτη που «δείχνει» στον επόμενο κόμβο για την ουρά που θα εισαχθεί.

Υπάρχουν οι συναρτήσεις queueEmpty που αρχικοποιεί μια ουρά. Η queueInsert η οποία εισάγει ένα πακέτο στο τέλος της ουράς. Η queueRemove που προσομοιώνει την εξυπηρέτηση του πακέτου που βρίσκεται στην αρχή της ουράς, αφαιρώντας το. Και η print\_queue που εκτυπώνει σε ένα αρχείο τα πακέτα των ουρών. Στο κύριο πρόγραμμα υπάρχει πίνακας που κρατάει το μέγεθος κάθε ουράς.

Στο κύριο (main) πρόγραμμα αφού δώσουμε από την κονσόλα τον μέγιστο μέσο φόρτο (σε μορφή δεκαδικού π.χ. 0.85 δηλ. 85%) γίνεται η αρχικοποίηση των πινάκων που χρησιμοποιούνται. Έπειτα υπολογίζονται το  $\alpha$  και οι πιθανότητες αφίξεων πακέτων για κάθε είσοδο 1-8 όπως περιγράφηκε παραπάνω. Στη συνέχεια υπολογίζεται ο πίνακας εισόδων-εξόδων με πιθανότητα 0.5 ο οποίος παραμένει σταθερός.

Ξεκινώντας την προσομοίωση στο αρχικό slot = 0 έχουμε μόνο αφίξεις πακέτων. Έτσι για κάθε είσοδο 1-8 και με την πιθανότητα που υπολογίστηκε πιο πάνω για κάθε

είσοδο ελέγχεται αν γίνει εισαγωγή πακέτων ή όχι. Αν υπάρξει αυτή η πιθανότητα τότε ανάμεσα στις υπάρχουσες εξόδους που υπάρχουν για τη συγκεκριμένη είσοδο (από τον πίνακα εισόδων-εξόδων) επιλέγεται τυχαία μια έξοδος και γίνεται εισαγωγή 3 πακέτων και υπάρχει η αντίστοιχη αύξηση του αριθμητικού μεγέθους της κάθε ουράς. Αφού τελειώσουν όλες οι εισοδοί προχωράει στο επόμενο slot στο οποίο εκτός από εισαγωγή πακέτων θα γίνει και εξυπηρέτηση.

Για την εξυπηρέτηση θα πρέπει να βρεθεί ένας permutation matrix έτσι ώστε να πληροί η προσομοίωση τις προϋποθέσεις. Αυτός ο πίνακας βρίσκεται με τη μέθοδο maximum weighted matching. Αναλυτικά (επειδή δε βρήκα κάποιον έτοιμο κώδικα που να προσομοιώνει κατάλληλα) αντιγράφει σε έναν προσωρινό πίνακα τα μεγέθη όλων των ουρών. Διατρέχει όλο τον πίνακα και βρίσκει το μεγαλύτερο και το αποθηκεύουμε το σημείο  $i,j$  και βάζουμε 1 στην αντίστοιχη θέση του permutation matrix. Έπειτα στον προσωρινό πίνακα, την αντίστοιχη γραμμή και στήλη του μέγιστου αριθμού που βρήκε, τις γεμίζει με μηδενικά. Στην ουσία αποκλείουμε την είσοδο και την έξοδο που επιλέξαμε πριν έτσι ώστε να μην επιλεγούν ξανά. Και προχωράμε βρίσκοντας πάντα τα υπόλοιπα μέγιστα μέχρις ότου μηδενίσει ο προσωρινός πίνακας. Με αυτόν τον τρόπο βρίσκεται ο permutation matrix των μέγιστων – πιο βαριών ουρών προς εξυπηρέτηση. Σύμφωνα με αυτόν τον πίνακα όπου υπάρχει 1 επιλέγονται οι ουρές που θα εξυπηρετηθούν από ένα πακέτο η κάθε μία.

Αυτό προσομοιώνεται για 5000 slots. Κάθε φορά που εξυπηρετείτε ένα πακέτο αποθηκεύεται η καθυστέρηση (delay, διαφορά του slot εξυπηρέτησης μείον το slot άφιξης του). Στο τέλος υπολογίζεται η μέση καθυστέρηση για κάθε ζεύγος εισόδου-εξόδου αλλά και η συνολική καθυστέρηση πακέτου ανεξάρτητα εισόδου-εξόδου. Η μέση καθυστέρηση για κάθε ζεύγος εισόδου-εξόδου υπολογίζεται με το πηλίκο της συνολικής καθυστέρησης των πακέτων που εξυπηρετήθηκαν προς των αριθμό αυτών των πακέτων.

Τα αποτελέσματα εκτυπώνονται σε 5 αρχεία.

Στο final\_results.txt εκτυπώνονται ο αριθμός όλων των πακέτων που αφίχθηκαν στο σύστημα, ο συνολικός αριθμός πακέτων που εξυπηρετήθηκαν, ο συνολικός αριθμός των slots που έτρεξε η προσομοίωση, ο μέγιστος μέσος φόρτος του συστήματος και το  $\alpha$  που υπολογίστηκε, η συνολική μέση καθυστέρηση πακέτου ανεξάρτητα εισόδου-εξόδου και η μέση καθυστέρηση για κάθε ζεύγος εισόδου-εξόδου (64 στο σύνολο όλα τα πιθανά ζεύγη).

Στο results.txt εκτυπώνονται τα πακέτα που εξυπηρετήθηκαν. Το id του πακέτου, ο αριθμός της ουράς εισόδου, ο αριθμός της εξόδου, το slot άφιξης, το slot εξυπηρέτησης και η καθυστέρηση (delay).

Στο permutation\_matrices.txt εκτυπώνονται σε κάθε slot ο permutation matrix που υπολογίστηκε με τον max-weighted matching και που δείχνει όπως είπαμε παραπάνω τα ζεύγη εισόδου-εξόδου που επιλέχθηκαν για να εξυπηρετηθούν τα πακέτα σύμφωνα με τους περιορισμούς.

Στο `queues_size.txt` εκτυπώνονται σε κάθε slot τα μεγέθη όλων των ουρών και στο `queues_remain.txt` εκτυπώνονται στο τέλος όλα τα πακέτα που έμειναν στις ουρές και δεν εξυπηρετήθηκαν αφού τελείωσε η προσομοίωση των 5000 slots.

Στο `errors.txt` εκτυπώνονται τυχόν λάθη που μπορεί να προκύψουν όπως αφαίρεση πακέτου από άδεια ουρά, αποτυχία δέσμευσης χώρου στη μνήμη.

Τέλος στην οθόνη αφού δώσει ο χρήστης το μέγιστο μέσο φόρτο, εκτυπώνεται το α, ο πίνακας με τις πιθανότητες αφίξεων για κάθε είσοδο, ο σταθερός πίνακας εισόδου-εξόδου, ο συνολικός αριθμός πακέτων που εξυπηρετήθηκαν και ο μέσος χρόνος καθυστέρησης πακέτου ανεξάρτητα εισόδου-εξόδου. Για κάθε ζεύγος εκτυπώνεται η μέση καθυστέρηση στο αρχείο `final_results.txt`, όπως είπαμε παραπάνω.

Στο κώδικα μέσα υπάρχει επαρκής σχολιασμός που περιγράφει σε κάθε σημείο το τι πραγματοποιείται.

### Αποτελέσματα πειραμάτων:

Για μέγιστους μέσους φόρτους 60%, 70%, 85%, 110%, 130%, 150%, 180%.

Επανάληψη πειραμάτων 5 φορές για κάθε φόρτο και υπολογισμός μ.ο.

Αφού υπολογίσω για κάθε ζεύγος τη μέση καθυστέρηση βρίσκω το μ.ο για όλες (64) ουρές.

Μέγιστος Μέσος Φόρτος 60%, $\alpha = 0.133333$	
Επαναλήψεις	Μέσος χρόνος καθυστέρησης ζευγών εισόδου-εξόδου (Slots)
1	50.102404
2	57.627950
3	72.432266
4	65.733708
5	40.981821
M.O	57.3756298

Μέγιστος Μέσος Φόρτος 60%, $\alpha = 0.133333$	
Επαναλήψεις	Μέσος χρόνος καθυστέρησης Συστήματος (Slots)
1	240.895715
2	213.221998
3	195.955878
4	243.775919
5	181.503798
M.O	215.0629486

Μέγιστος Μέσος Φόρτος 70%, $\alpha = 0.155556$	
Επαναλήψεις	Μέσος χρόνος καθυστέρησης ζευγών εισόδου-εξόδου (Slots)
1	77.774932
2	82.147980
3	106.689779
4	128.010310
5	148.424904
M.O	108.609581

Μέγιστος Μέσος Φόρτος 70%, $\alpha = 0.155556$	
Επαναλήψεις	Μέσος χρόνος καθυστέρησης Συστήματος (Slots)
1	354.813719
2	418.538263
3	310.277718
4	316.945925
5	371.337302
M.O	352.5625788

Μέγιστος Μέσος Φόρτος 85%, $\alpha = 0.188889$	
Επαναλήψεις	Μέσος χρόνος καθυστέρησης ζευγών εισόδου-εξόδου (Slots)
1	139.338326
2	193.217851
3	203.916816
4	214.692120
5	197.470353
M.O	189.7270932

Μέγιστος Μέσος Φόρτος 85%, $\alpha = 0.188889$	
Επαναλήψεις	Μέσος χρόνος καθυστέρησης Συστήματος (Slots)
1	511.667043
2	554.385988
3	557.222112
4	562.256885
5	605.556291
M.O	558.2176638

Μέγιστος Μέσος Φόρτος 110%, $\alpha = 0.244444$	
Επαναλήψεις	Μέσος χρόνος καθυστέρησης ζευγών εισόδου-εξόδου (Slots)
1	296.282843
2	348.761120
3	375.554128
4	221.175889
5	275.943417
M.O	303.5434788

Μέγιστος Μέσος Φόρτος 110%, $\alpha = 0.244444$	
Επαναλήψεις	Μέσος χρόνος καθυστέρησης Συστήματος (Slots)
1	1003.180302
2	852.664421
3	794.558244
4	755.977266
5	801.823387
M.O	841.640724

Μέγιστος Μέσος Φόρτος 130%, $\alpha = 0.288889$	
Επαναλήψεις	Μέσος χρόνος καθυστέρησης ζευγών εισόδου-εξόδου (Slots)
1	438.910692
2	383.495430
3	213.308637
4	388.066189
5	400.898729
M.O	364.9359354

Μέγιστος Μέσος Φόρτος 130%, $\alpha = 0.288889$	
Επαναλήψεις	Μέσος χρόνος καθυστέρησης Συστήματος (Slots)
1	1077.190126
2	992.290985
3	846.214719
4	995.141480
5	1023.347522
M.O	986.8369664

Μέγιστος Μέσος Φόρτος 150%, $\alpha = 0.333333$	
Επαναλήψεις	Μέσος χρόνος καθυστέρησης ζευγών εισόδου-εξόδου (Slots)
1	350.107807
2	483.002000
3	474.848047
4	366.578192
5	460.356250
M.O	426.9784592

Μέγιστος Μέσος Φόρτος 150%, $\alpha = 0.333333$	
Επαναλήψεις	Μέσος χρόνος καθυστέρησης Συστήματος (Slots)
1	1070.622083
2	1170.101705
3	1034.793889
4	1086.151778
5	1100.303795
M.O	1092.39465



Μέγιστος Μέσος Φόρτος 180%, $\alpha = 0.400000$	
Επαναλήψεις	Μέσος χρόνος καθυστέρησης ζευγών εισόδου-εξόδου (Slots)
1	414.777255
2	391.705208
3	367.543527
4	226.413290
5	282.181879
M.O	336.5242318

Μέγιστος Μέσος Φόρτος 180%, $\alpha = 0.400000$	
Επαναλήψεις	Μέσος χρόνος καθυστέρησης Συστήματος (Slots)
1	1271.997941
2	1194.161223
3	1151.710333
4	1055.563203
5	1041.669292
M.O	1143.0203984

Συνολικός Πίνακας M.O των 5 πειραμάτων:

M.O		
	Μέσος χρόνος καθυστέρησης Συστήματος (Slots)	Μέσος χρόνος καθυστέρησης Συστήματος (Slots)
Μέγιστος Μέσος Φόρτος 60%, $\alpha = 0.133333$	57.3756298	215.0629486
Μέγιστος Μέσος Φόρτος 70%, $\alpha = 0.155556$	108.609581	352.5625788
Μέγιστος Μέσος Φόρτος 85%, $\alpha = 0.188889$	189.7270932	558.2176638
Μέγιστος Μέσος Φόρτος 110%, $\alpha = 0.244444$	303.5434788	841.640724
Μέγιστος Μέσος Φόρτος 130%, $\alpha = 0.288889$	364.9359354	986.8369664
Μέγιστος Μέσος Φόρτος 150%, $\alpha = 0.333333$	426.9784592	1092.39465
Μέγιστος Μέσος Φόρτος 180%, $\alpha = 0.400000$	336.5242318	1143.0203984

Όπως προκύπτει όσο αυξάνεται ο φόρτος αυξάνεται και η μέση καθυστέρηση εξυπηρέτησης.

Όταν ο φόρτος υπερβεί ένα σημείο το 150%, τότε το  $\alpha$  γίνεται 0,33 και οι ουρές 3 και 7 έχουν αφίξεις πακέτων με πιθανότητα 1. Δηλαδή έρχονται συνέχεια πακέτα κάτι το οποίο κάνει το σύστημα ασταθές, διότι δε προλαβαίνει να εξυπηρετήσει τα πακέτα που έρχονται και οι ουρές αυτές θα συνεχίσουν να αυξάνονται στο άπειρο θεωρητικά, πρακτικά μέχρι να τελειώσει η μνήμη.