



ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΠΙΣΤΗΜΗ ΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

<http://eclass.aueb.gr/courses/INF511/>

Ενδεικτικές Επαναληπτικές Ασκήσεις

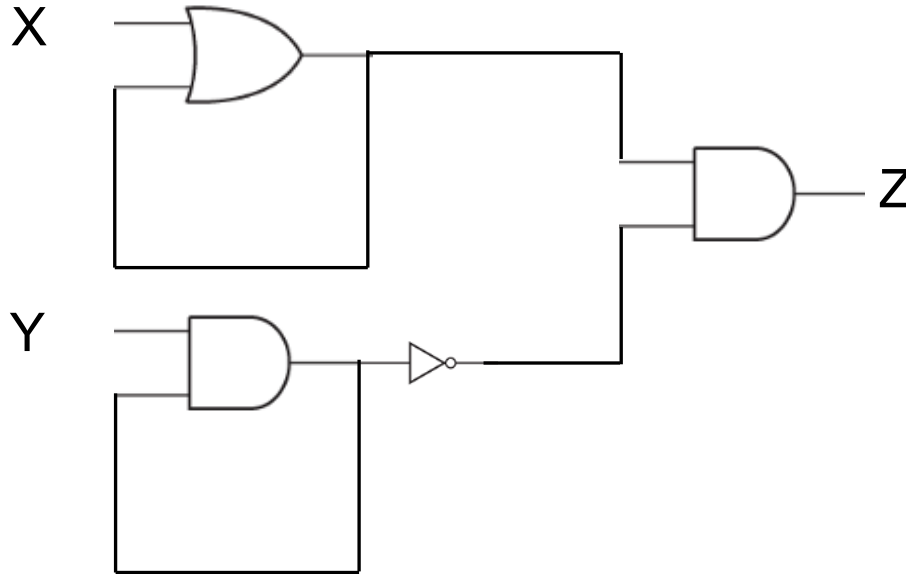
Οι Ασκήσεις Είναι Ενδεικτικές

Προσοχή!!!

Οι ασκήσεις που ακολουθούν εδώ είναι **ενδεικτικές** και **συμπληρωματικές** σε ότι έχουμε κάνει στις διαλέξεις.

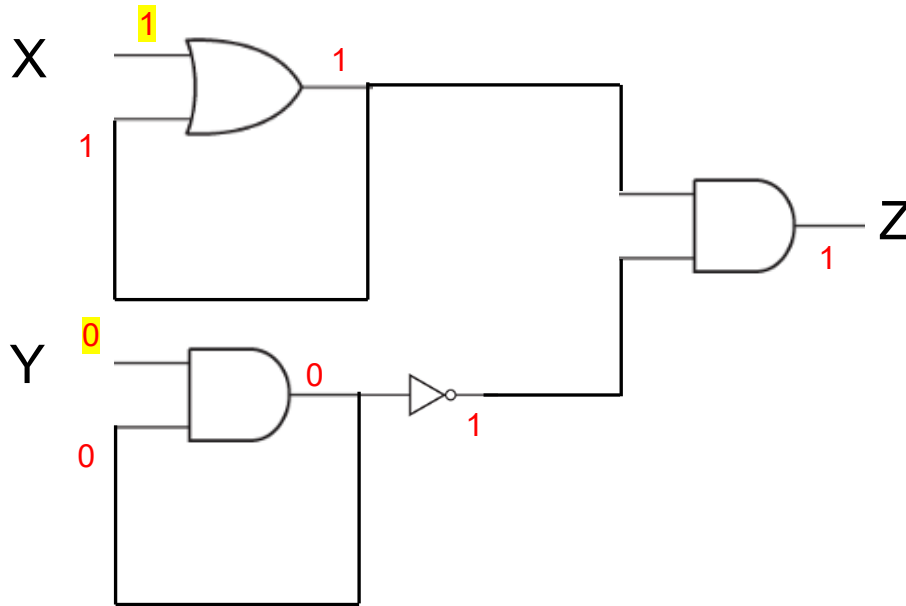
Για την εξεταστέα ύλη συμβουλευτείτε το αρχείο [Final exam material.pdf](#)

Κεφάλαιο 1 (Αποθήκευση Δεδομένων)



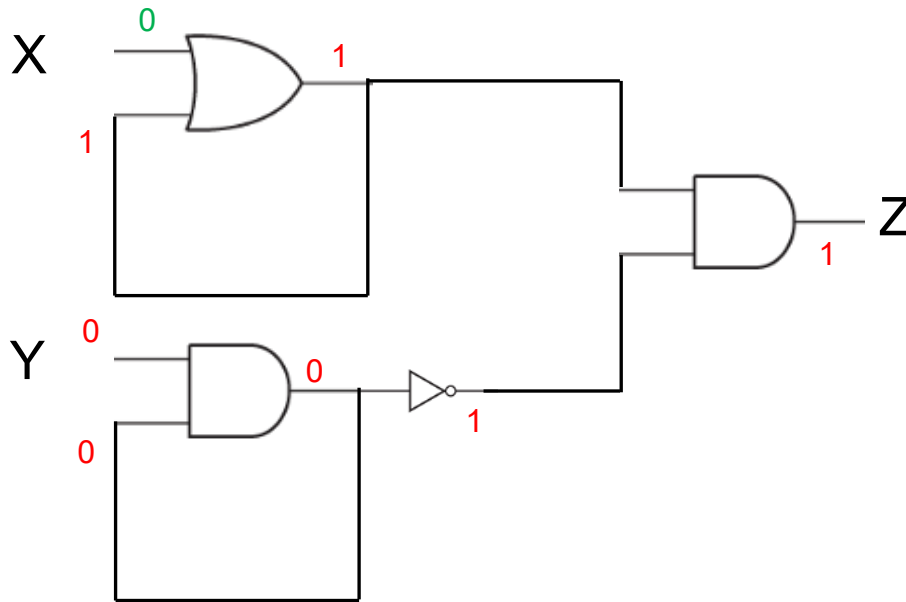
- Πόσο είναι η έξοδος Z αν $X=1$ και $Y=0$;
- Τι θα γίνει αν μετά το X γίνει 0; Τι θα γίνει αν το Y γίνει 1;

Κεφάλαιο 1 (Αποθήκευση Δεδομένων)



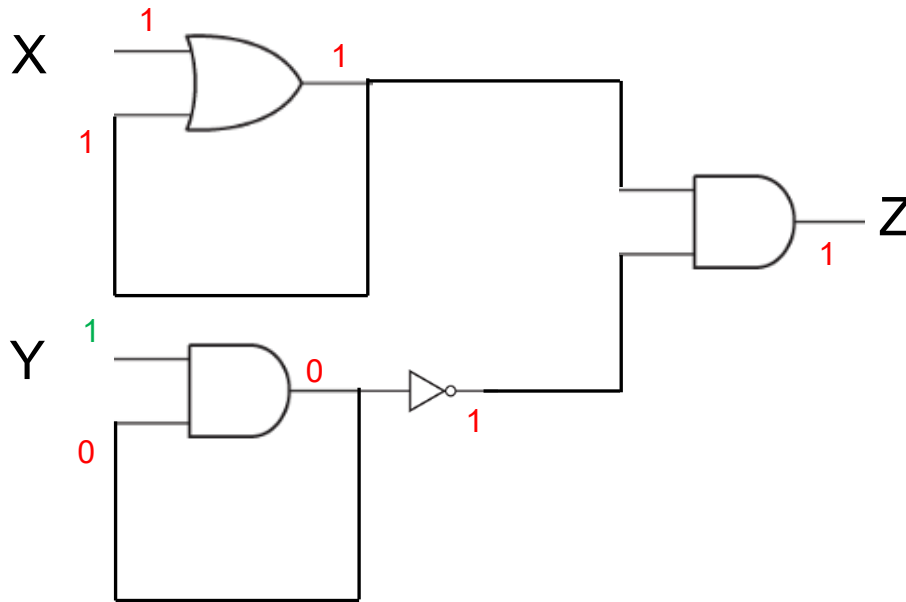
- Πόσο είναι η έξοδος Z αν $X=1$ και $Y=0$;
- Τι θα γίνει αν μετά το X γίνει 0; Τι θα γίνει αν το Y γίνει 1;
- Αν $X=1$ και $Y=0$, τότε θα έχουμε $Z=1$.
- Το Z θα παραμείνει 1 όπως και να αλλάξουν μετά τα X και Y.

Κεφάλαιο 1 (Αποθήκευση Δεδομένων)



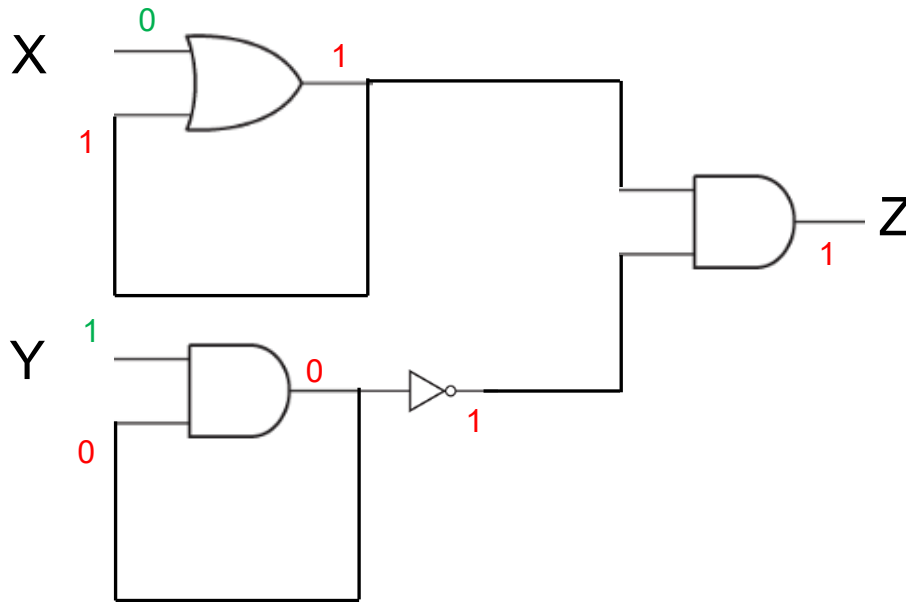
- Πόσο είναι η έξοδος Z αν $X=1$ και $Y=0$;
- Τι θα γίνει αν μετά το X γίνει 0; Τι θα γίνει αν το Y γίνει 1;
- Αν $X=1$ και $Y=0$, τότε θα έχουμε $Z=1$.
- Το Z θα παραμείνει 1 όπως και να αλλάξουν μετά τα X και Y.

Κεφάλαιο 1 (Αποθήκευση Δεδομένων)



- Πόσο είναι η έξοδος Z αν $X=1$ και $Y=0$;
- Τι θα γίνει αν μετά το X γίνει 0; Τι θα γίνει αν το Y γίνει 1;
- Αν $X=1$ και $Y=0$, τότε θα έχουμε $Z=1$.
- Το Z θα παραμείνει 1 όπως και να αλλάξουν μετά τα X και Y.

Κεφάλαιο 1 (Αποθήκευση Δεδομένων)



- Πόσο είναι η έξοδος Z αν $X=1$ και $Y=0$;
- Τι θα γίνει αν μετά το X γίνει 0; Τι θα γίνει αν το Y γίνει 1;
- Αν $X=1$ και $Y=0$, τότε θα έχουμε $Z=1$.
- Το Z θα παραμείνει 1 όπως και να αλλάξουν μετά τα X και Y.

Κεφάλαιο 1 (Αποθήκευση Δεδομένων)

- Μετατρέψτε το δεκαδικό αριθμό 6.375 στο δυαδικό σύστημα
- Κωδικοποιείστε το δεκαδικό αριθμό 6.375 σε κανονικοποιημένη μορφή των **8 bits**, με 3 bits για τον εκθέτη με υπέρβαση κατά 4 και 4 bits για το σημαινόμενο μέρος

Υπέρβαση κατά 4

Σχήμα μπιτ	Τιμή
111	3
110	2
101	1
100	0
011	-1
010	-2
001	-3
000	-4

Κεφάλαιο 1 (Αποθήκευση Δεδομένων)

- Μετατρέψτε το δεκαδικό αριθμό 6.375 στο δυαδικό σύστημα
- Κωδικοποιείστε το δεκαδικό αριθμό 6.375 σε κανονικοποιημένη μορφή των **8 bits**, με 3 bits για τον εκθέτη με υπέρβαση κατά 4 και 4 bits για το σημαινόμενο μέρος
- Δεκαδικό μέρος 6 = 110

Κεφάλαιο 1 (Αποθήκευση Δεδομένων)

- Μετατρέψτε το δεκαδικό αριθμό 6.375 στο δυαδικό σύστημα
- Κωδικοποιείστε το δεκαδικό αριθμό 6.375 σε κανονικοποιημένη μορφή των **8 bits**, με 3 bits για τον εκθέτη με υπέρβαση κατά 4 και 4 bits για το σημαινόμενο μέρος
- Ακέραιο μέρος 6 = **110**
- $0.375 \times 2 = 0.75$ carry=0
- $0.75 \times 2 = 1.5$ carry=1
- $0.5 \times 2 = 1.0$ carry=1
- Δεκαδικό μέρος **011**
- Άρα 6.375 (δεκαδικό) $\rightarrow$ 110.011 (δυαδικό)

Κεφάλαιο 1 (Αποθήκευση Δεδομένων)

Υπέρβαση κατά 4

- Μετατρέψτε το δεκαδικό αριθμό 6.375 στο δυαδικό σύστημα

- 110.011



Σχήμα μπιτ	Τιμή
111	3
110	2
101	1
100	0
011	-1
010	-2
001	-3
000	-4

- Κωδικοποιείστε το δεκαδικό αριθμό 6.375 σε κανονικοποιημένη μορφή των **8 bits**, με 3 bits για τον εκθέτη με υπέρβαση κατά 4 και 4 bits για το σημαινόμενο μέρος

- $110.011 = 1.10011 \cdot 2^2$

bit προσήμου: 0

εκθέτης: 110

σημαινόμενο μέρος: 1001

Κωδικοποίηση: 01101001

(σφάλμα περικοπής)

Υπέρβαση κατά 4

- Μετατρέψτε το δεκαδικό αριθμό 6.375 στο δυαδικό σύστημα
- 110.011

- Κωδικοποιείστε το δεκαδικό αριθμό 6.375 σε κανονικοποιημένη μορφή των **8 bits**, με 3 bits για τον εκθέτη με υπέρβαση κατά 4 και 4 bits για το σημαινόμενο μέρος

- $110.011 = 1.10011 \cdot 2^2$

bit προσήμου: 0

εκθέτης: 110

σημαινόμενο μέρος: 1001

Κωδικοποίηση: 01101001

(σφάλμα περικοπής)

Κεφάλαιο 1 (Αποθήκευση Δεδομένων)

- Κωδικοποιείστε τη λέξη **BED** χρησιμοποιώντας τον παρακάτω **κώδικα Hamming**, και στη συνέχεια προσθέστε ένα bit ισοτιμίας σε κάθε κωδικοποιημένο γράμμα, ώστε να έχετε **περιττή ισοτιμία**.

Σύμβολο	Κώδικας
A	000000
B	001111
C	010011
D	011100
E	100110
F	101001
G	110101
H	111010

- Έστω ότι σας στάλθηκε ένα γράμμα που κωδικοποιήθηκε όπως παραπάνω και λάβατε τον κώδικα **1101001**. Ποια είναι τα γράμματα (ή το γράμμα) που μπορεί με τη μεγαλύτερη πιθανότητα (δηλαδή λιγότερα σφάλματα) να στάλθηκαν;

Κεφάλαιο 1 (Αποθήκευση Δεδομένων)

- Κωδικοποιείστε τη λέξη **BED** χρησιμοποιώντας τον παρακάτω **κώδικα Hamming**, και στη συνέχεια προσθέστε ένα bit ισοτιμίας σε κάθε κωδικοποιημένο γράμμα, ώστε να έχετε **περιττή ισοτιμία**.

- Κώδικας Hamming:

- B: 001111
- E: 100110
- D: 011100

Τελική κωδικοποιημένη λέξη

1001111 0100110 0011100

- Περιττή ισοτιμία:

- B: 1001111
- E: 0100110
- D: 0011100

Σύμβολο	Κώδικας
A	000000
B	001111
C	010011
D	011100
E	100110
F	101001
G	110101
H	111010

- Έστω ότι σας στάλθηκε ένα γράμμα που κωδικοποιήθηκε όπως παραπάνω και λάβατε τον κώδικα 1101001. Ποια είναι τα γράμματα (ή το γράμμα) που μπορεί με τη μεγαλύτερη πιθανότητα (δηλαδή λιγότερα σφάλματα) να στάλθηκαν;
- Δεν μπορεί να μην έχει συμβεί κάποιο σφάλμα. Γιατί;

Κεφάλαιο 1 (Αποθήκευση Δεδομένων)

- Κωδικοποιείστε τη λέξη **BED** χρησιμοποιώντας τον παρακάτω **κώδικα Hamming**, και στη συνέχεια προσθέστε ένα bit ισοτιμίας σε κάθε κωδικοποιημένο γράμμα, ώστε να έχετε **περιττή ισοτιμία**.

- Κώδικας Hamming:

- B: 001111
- E: 100110
- D: 011100

Τελική κωδικοποιημένη λέξη

1001111 0100110 0011100

- Περιττή ισοτιμία:

- B: 1001111
- E: 0100110
- D: 0011100

Σύμβολο	Κώδικας
A	000000
B	001111
C	010011
D	011100
E	100110
F	101001
G	110101
H	111010

- Έστω ότι σας στάλθηκε ένα γράμμα που κωδικοποιήθηκε όπως παραπάνω και λάβατε τον κώδικα 1101001. Ποια είναι τα γράμματα (ή το γράμμα) που μπορεί με τη μεγαλύτερη πιθανότητα (δηλαδή λιγότερα σφάλματα) να στάλθηκαν;
- Δεν μπορεί να μην έχει συμβεί κάποιο σφάλμα. Γιατί;
- Στη λέξη 1101001 υπάρχει άρτιος αριθμός 1

Κεφάλαιο 1 (Αποθήκευση Δεδομένων)

- Έστω ότι σας στάλθηκε ένα γράμμα που κωδικοποιήθηκε όπως παραπάνω και λάβατε τον κώδικα 1101001. Ποια είναι τα γράμματα (ή το γράμμα) που μπορεί με τη μεγαλύτερη πιθανότητα (δηλαδή λιγότερα σφάλματα) να στάλθηκαν;
- Δεν μπορεί να μην έχει συμβεί κάποιο σφάλμα. Γιατί;
Με 1 σφάλμα μπορεί να είναι μόνο το F (με σφάλμα στο bit ισοτιμίας).

Χαρακτήρας	Κώδικας							Σχήμα που λαμβάνεται							Απόσταση
A	1	0	0	0	0	0	0	1	1	0	1	0	0	1	3
B	1	0	0	1	1	1	1	1	1	0	1	0	0	1	3
C	0	0	1	0	0	1	1	1	1	0	1	0	0	1	5
D	0	0	1	1	1	0	0	1	1	0	1	0	0	1	5
E	0	1	0	0	1	1	0	1	1	0	1	0	0	1	5
F	0	1	0	1	0	0	1	1	1	0	1	0	0	1	1
G	1	1	1	0	1	0	1	1	1	0	1	0	0	1	3
H	1	1	1	1	0	1	0	1	1	0	1	0	0	1	3

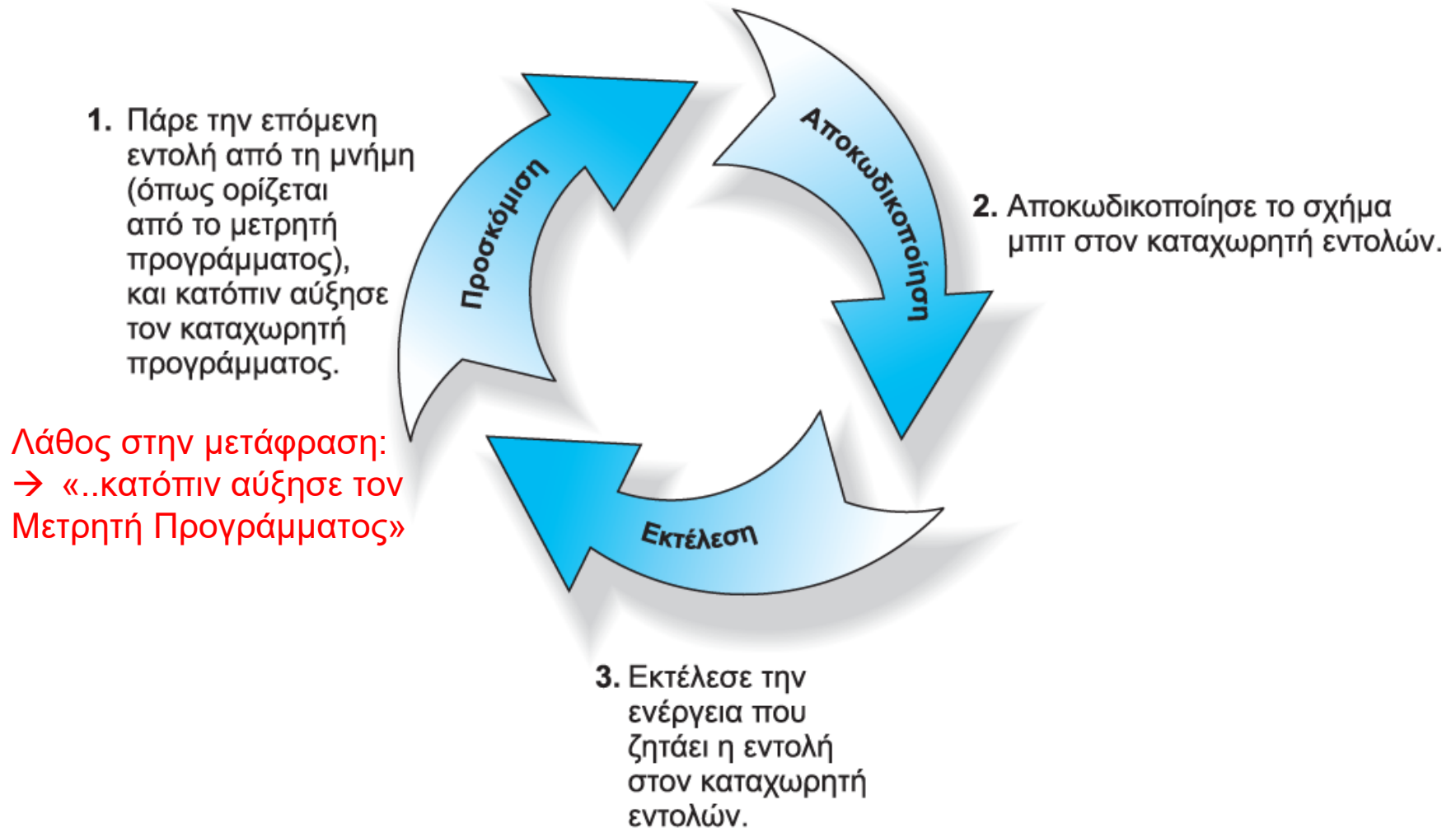
Κεφάλαιο 2 (Χειρισμός Δεδομένων)

- Υποθέστε ότι ο μετρητής προγράμματος έχει την τιμή 00 και ότι η κύρια μνήμη είναι όπως φαίνεται παρακάτω. Εξηγήστε τι αλλαγές θα γίνουν μετά την εκτέλεση του προγράμματος. Τι κάνει το πρόγραμμα;

Κύρια μνήμη	
Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C

Κύρια μνήμη	
Διεύθυνση	Περιεχόμενο
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00

Ο κύκλος εκτέλεσης μιας εντολής



Παράρτημα Γ: Μια Απλή Γλώσσα Μηχανής (1)

Κωδ. Λειτ.	Τελεστέος	Περιγραφή
1	R X Y	Φορτώνει (LOAD) τον καταχωρητή R με την ακολουθία bit που υπάρχει στο κελί μνήμης με διεύθυνση XY.
2	R X Y	Φορτώνει (LOAD) τον καταχωρητή R με την ακολουθία bit XY.
3	R X Y	Αποθηκεύει (STORE) την ακολουθία bit που περιέχει ο καταχωρητής R στο κελί μνήμης με διεύθυνση XY.
4	0 R S	Μετακινεί (MOVE) την ακολουθία bit που περιέχει ο καταχωρητής R στον καταχωρητή S.

Παράρτημα Γ: Μια Απλή Γλώσσα Μηχανής (2)

Κωδ Λειτ.	Τελεστέος	Περιγραφή
5	R S T	Προσθέτει (ADD) τις ακολουθίες bit των καταχωρητών S και T (σε μορφή συμπληρώματος ως προς δύο), και τοποθετεί το αποτέλεσμα στον καταχωρητή R.
6	R S T	Προσθέτει (ADD) τις ακολουθίες bit των καταχωρητών S και T (σε μορφή κινητής υποδιαστολής), και τοποθετεί το αποτέλεσμα στον καταχωρητή R.
7	R S T	Εκτελεί την πράξη OR στις ακολουθίες bit των καταχωρητών S και T και τοποθετεί το αποτέλεσμα στον καταχωρητή R.
8	R S T	Εκτελεί την πράξη AND στις ακολουθίες bit των καταχωρητών S και T και τοποθετεί το αποτέλεσμα στον καταχωρητή R.

Παράρτημα Γ: Μια Απλή Γλώσσα Μηχανής (3)

Κωδ Λειτ.	Τελεστέος	Περιγραφή
9	R S T	Εκτελεί την πράξη XOR στις ακολουθίες bit των καταχωρητών S και T και τοποθετεί το αποτέλεσμα στον καταχωρητή R.
A	R 0 X	Περιστρέφει (rotate) την ακολουθία bit του καταχωρητή R ένα bit προς τα δεξιά X φορές. Το bit του άκρου χαμηλής τάξης (LSB) τοποθετείται στο άκρο υψηλής τάξης (MSB) (κυκλική ολίσθηση)
B	R X Y	Μεταπηδά (JUMP) στην εντολή που βρίσκεται στο κελί μνήμης με διεύθυνση XY, αν η ακολουθία bit του R είναι ίδια με την ακολουθία bit του καταχωρητή 0. Αλλιώς συνεχίζεται η κανονική ροή της εκτέλεσης.
C	0 0 0	Τερματίζει (HALT) την εκτέλεση.

Κύρια μνήμη

Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00

Προσκόμιση – λήψη εντολής από τη μνήμη

PC	IR	Επεξήγηση
00	11AA	

Καταχωρητής

0	
1	
2	
3	
4	
5	

Κύρια μνήμη

Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00

Προσκόμιση – αύξηση του PC

PC	IR	Επεξήγηση
02	11AA	

Καταχωρητής

0	
1	
2	
3	
4	
5	

Κύρια μνήμη

Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00
...	
AA	VALUE

Εκτέλεση

PC	IR	Επεξήγηση
02	11AA	LOAD R1 WITH VALUE FROM AA

Καταχωρητής

0	
1	VALUE FROM AA
2	
3	
4	
5	

Κύρια μνήμη

Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00
...	
AA	VALUE

Προσκόμιση – λήψη εντολής από τη μνήμη

PC	IR	Επεξήγηση
02	2001	

Καταχωρητής

0	
1	VALUE FROM AA
2	
3	
4	
5	

Κύρια μνήμη

Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00
...	
AA	VALUE

Προσκόμιση – αύξηση του PC

PC	IR	Επεξήγηση
04	2001	

Καταχωρητής

0	
1	VALUE FROM AA
2	
3	
4	
5	

Κύρια μνήμη

Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00
...	
AA	VALUE

Εκτέλεση

PC	IR	Επεξήγηση
04	2001	LOAD R0 WITH 01

Καταχωρητής

0	01
1	VALUE FROM AA
2	
3	
4	
5	

Κύρια μνήμη

Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00
...	
AA	VALUE

Προσκόμιση – λήψη εντολής από τη μνήμη

PC	IR	Επεξήγηση
04	8210	

Καταχωρητής

0	01
1	VALUE FROM AA
2	
3	
4	
5	

Κύρια μνήμη

Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00
...	
AA	VALUE

Προσκόμιση – αύξηση του PC

PC	IR	Επεξήγηση
06	8210	

Καταχωρητής

0	01
1	VALUE FROM AA
2	
3	
4	
5	

Κύρια μνήμη

Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00
...	
AA	VALUE

Εκτέλεση

PC	IR	Επεξήγηση
06	8210	STORE TO R2 THE RESULT R0 AND R1

Καταχωρητής

0	01
1	VALUE FROM AA
2	R0 AND R1
3	
4	
5	

Κύρια μνήμη

Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00
...	
AA	VALUE

Προσκόμιση – λήψη εντολής από τη μνήμη

PC	IR	Επεξήγηση
06	B20C	

Καταχωρητής

0	01
1	VALUE FROM AA
2	R0 AND R1
3	
4	
5	

Κύρια μνήμη

Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00
...	
AA	VALUE

Προσκόμιση – αύξηση του PC

PC	IR	Επεξήγηση
08	B20C	

Καταχωρητής

0	01
1	VALUE FROM AA
2	R0 AND R1
3	
4	
5	

Κύρια μνήμη

Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00
...	
AA	XXXXXXXX1

Εκτέλεση

PC	IR	Επεξήγηση
08	B20C	IF R2=R0 THEN JUMP TO 0C (where is HALT)

Για να είναι $R2=R0$ θα πρέπει η πράξη $R0 \text{ AND } R1$ να έχει αποτέλεσμα 01. Αυτό συμβαίνει μόνο εάν το τελευταίο bit στη διεύθυνση AA είναι το 1.

Καταχωρητής	
0	01
1	VALUE FROM AA
2	R0 AND R1
3	
4	
5	

Κύρια μνήμη

Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00
...	
AA	XXXXXXXX0

Εκτέλεση

PC	IR	Επεξήγηση
08	B20C	IF R2<>R0 THEN CONTINUE

Αλλοιώς εάν το τελευταίο bit στη διεύθυνση AA είναι το 0 (R0 AND R1 = 0), τότε συνεχίζει.

Καταχωρητής	
0	01
1	VALUE FROM AA
2	R0 AND R1
3	
4	
5	

Κύρια μνήμη

Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00
...	
AA	XXXXXXXX0

Προσκόμιση – λήψη εντολής από τη μνήμη

PC	IR	Επεξήγηση
08	7210	

Καταχωρητής

0	01
1	VALUE FROM AA
2	R0 AND R1
3	
4	
5	

Κύρια μνήμη

Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00
...	
AA	XXXXXXXX0

Προσκόμιση – αύξηση του PC

PC	IR	Επεξήγηση
0A	7210	

Καταχωρητής

0	01
1	VALUE FROM AA
2	R0 AND R1
3	
4	
5	

Κύρια μνήμη

Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00
...	
AA	XXXXXXXX0

Εκτέλεση

PC	IR	Επεξήγηση
0A	7210	STORE TO R2 WITH RESULT R0 OR R1

Η συγκεκριμένη πράξη κάνει το τελευταίο bit στη διεύθυνση AA 1 (εάν δεν είναι ήδη).

Καταχωρητής

0	01
1	VALUE FROM AA
2	R0 OR R1 = XXXXXXXX1
3	
4	
5	

Κύρια μνήμη

Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00
...	
AA	XXXXXXXX0

Προσκόμιση – λήψη εντολής από τη μνήμη

PC	IR	Επεξήγηση
0A	32AA	

Καταχωρητής

0	01
1	VALUE FROM AA
2	R0 OR R1 = XXXXXXXX1
3	
4	
5	

Κύρια μνήμη

Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00
...	
AA	XXXXXXXX0

Προσκόμιση – αύξηση του PC

PC	IR	Επεξήγηση
0C	32AA	

Καταχωρητής

0	01
1	VALUE FROM AA
2	R0 OR R1 = XXXXXXXX1
3	
4	
5	

Κύρια μνήμη

Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00
...	
AA	XXXXXXXX1

Εκτέλεση

PC	IR	Επεξήγηση
0C	32AA	STORE R2 TO MEM AA AND THEN HALT

Καταχωρητής

0	01
1	VALUE FROM AA
2	R0 OR R1 = XXXXXXXX1
3	
4	
5	

Κεφάλαιο 2 (Χειρισμός Δεδομένων)

- Υποθέστε ότι ο μετρητής προγράμματος έχει την τιμή 00 και ότι η κύρια μνήμη είναι όπως φαίνεται παρακάτω. Εξηγήστε τι αλλαγές θα γίνουν μετά την εκτέλεση του προγράμματος. Τι κάνει το πρόγραμμα;

Κύρια μνήμη	
Διεύθυνση	Περιεχόμενο
00	11
01	AA
02	20
03	01
04	82
05	10
06	B2
07	0C

Κύρια μνήμη	
Διεύθυνση	Περιεχόμενο
08	72
09	10
0A	32
0B	AA
0C	C0
0D	00

Το πρόγραμμα ελέγχει το τελευταίο bit από το περιεχόμενο στη διεύθυνση μνήμης AA. Αν είναι 0 το κάνει 1, αλλιώς δεν κάνει τίποτα.

Κεφάλαιο 3 (Λειτουργικά Συστήματα)

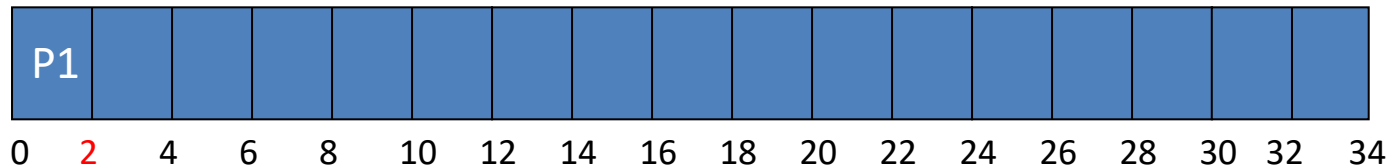
- Δίνονται οι διεργασίες P1, P2, P3, P4 με χρονική στιγμή άφιξης και διάρκεια όπως φαίνεται στον παρακάτω πίνακα. Να υπολογιστεί ο **μέσος χρόνος ολοκλήρωσης** όταν ακολουθείται η πολιτική Round Robin με περίοδο $T=2$ sec, όπου σε κάθε διεργασία που μπαίνει στην ουρά δίνεται η μικρότερη προτεραιότητα. (Υποθέστε μηδενικό χρόνο μεταγωγής, δηλαδή $\delta=0$)

Διεργασία	Χρονική στιγμή άφιξης	Διάρκεια διεργασίας
P1	0 sec	12 sec
P2	5 sec	2 sec
P3	7 sec	14 sec
P4	13 sec	6 sec

Κεφάλαιο 3 (Λειτουργικά Συστήματα) – TIME=2

- Δίνονται οι διεργασίες P1, P2, P3, P4 με χρονική στιγμή άφιξης και διάρκεια όπως φαίνεται στον παρακάτω πίνακα. Να υπολογιστεί ο **μέσος χρόνος ολοκλήρωσης** όταν ακολουθείται η πολιτική Round Robin με περίοδο $T=2$ sec, όπου σε κάθε διεργασία που μπαίνει στην ουρά δίνεται η μικρότερη προτεραιότητα. (Υποθέστε μηδενικό χρόνο μεταγωγής, δηλαδή $\delta=0$)

Διεργασία	Χρονική στιγμή άφιξης	Διάρκεια διεργασίας
P1	0 sec	12 sec, remain=10
P2	5 sec	2 sec
P3	7 sec	14 sec
P4	13 sec	6 sec

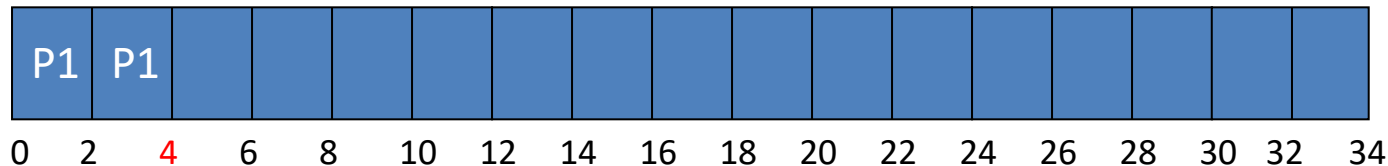


QUEUE = EMPTY

Κεφάλαιο 3 (Λειτουργικά Συστήματα) – TIME=4

- Δίνονται οι διεργασίες P1, P2, P3, P4 με χρονική στιγμή άφιξης και διάρκεια όπως φαίνεται στον παρακάτω πίνακα. Να υπολογιστεί ο **μέσος χρόνος ολοκλήρωσης** όταν ακολουθείται η πολιτική Round Robin με περίοδο $T=2$ sec, όπου σε κάθε διεργασία που μπαίνει στην ουρά δίνεται η μικρότερη προτεραιότητα. (Υποθέστε μηδενικό χρόνο μεταγωγής, δηλαδή $\delta=0$)

Διεργασία	Χρονική στιγμή άφιξης	Διάρκεια διεργασίας
P1	0 sec	12 sec, remain=8
P2	5 sec	2 sec
P3	7 sec	14 sec
P4	13 sec	6 sec

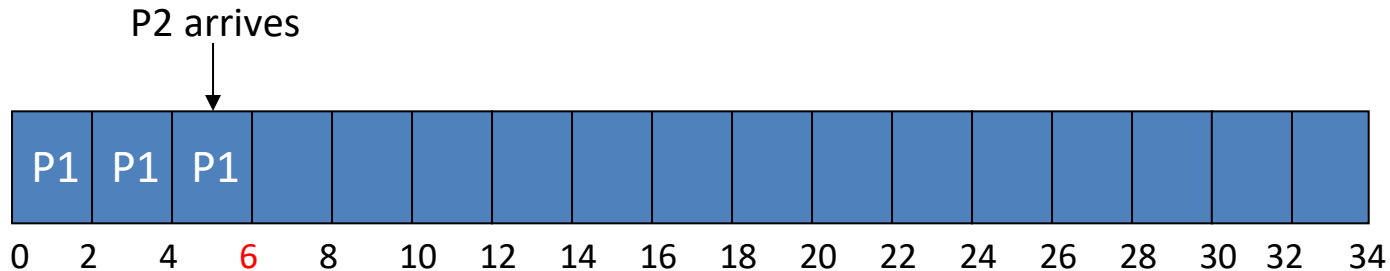


QUEUE = EMPTY

Κεφάλαιο 3 (Λειτουργικά Συστήματα) – TIME=6

- Δίνονται οι διεργασίες P1, P2, P3, P4 με χρονική στιγμή άφιξης και διάρκεια όπως φαίνεται στον παρακάτω πίνακα. Να υπολογιστεί ο **μέσος χρόνος ολοκλήρωσης** όταν ακολουθείται η πολιτική Round Robin με περίοδο $T=2$ sec, όπου σε κάθε διεργασία που μπαίνει στην ουρά δίνεται η μικρότερη προτεραιότητα. (Υποθέστε μηδενικό χρόνο μεταγωγής, δηλαδή $\delta=0$)

Διεργασία	Χρονική στιγμή άφιξης	Διάρκεια διεργασίας
P1	0 sec	12 sec, remain=6
P2	5 sec	2 sec
P3	7 sec	14 sec
P4	13 sec	6 sec

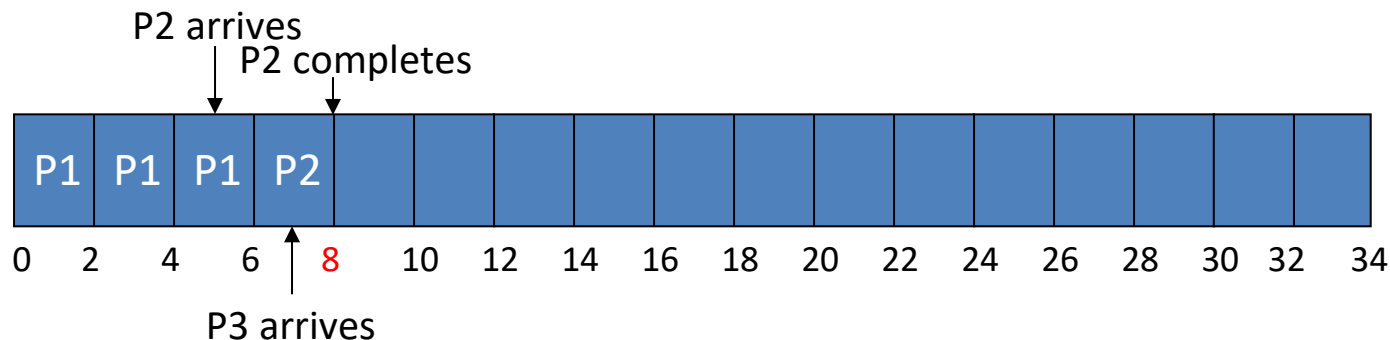


QUEUE = P2

Κεφάλαιο 3 (Λειτουργικά Συστήματα) – TIME=8

- Δίνονται οι διεργασίες P1, P2, P3, P4 με χρονική στιγμή άφιξης και διάρκεια όπως φαίνεται στον παρακάτω πίνακα. Να υπολογιστεί ο **μέσος χρόνος ολοκλήρωσης** όταν ακολουθείται η πολιτική Round Robin με περίοδο $T=2$ sec, όπου σε κάθε διεργασία που μπαίνει στην ουρά δίνεται η μικρότερη προτεραιότητα. (Υποθέστε μηδενικό χρόνο μεταγωγής, δηλαδή $\delta=0$)

Διεργασία	Χρονική στιγμή άφιξης	Διάρκεια διεργασίας
P1	0 sec	12 sec, remain=6
P2	5 sec	2 sec, remain=0
P3	7 sec	14 sec
P4	13 sec	6 sec

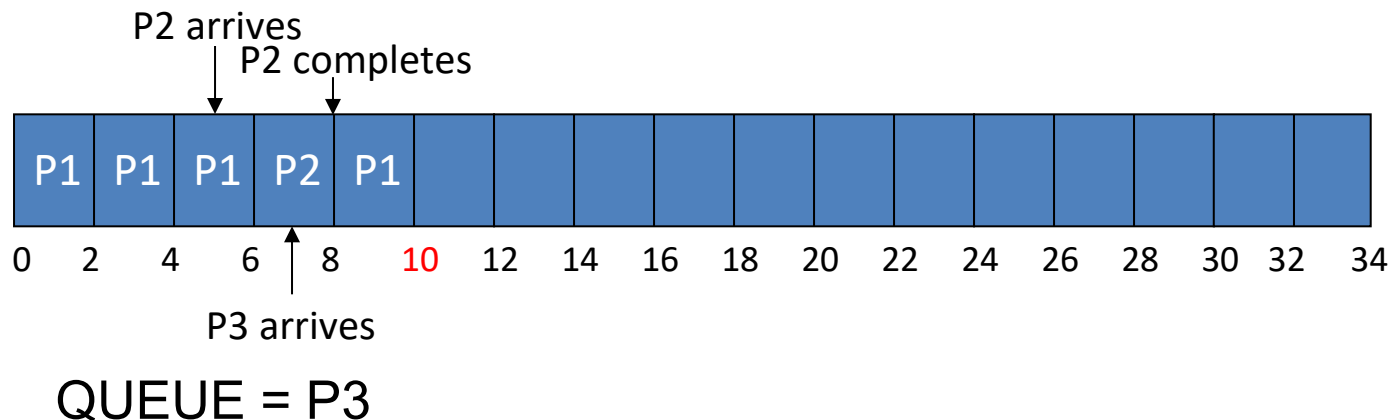


QUEUE = P3, P1

Κεφάλαιο 3 (Λειτουργικά Συστήματα) – TIME=10

- Δίνονται οι διεργασίες P1, P2, P3, P4 με χρονική στιγμή άφιξης και διάρκεια όπως φαίνεται στον παρακάτω πίνακα. Να υπολογιστεί ο **μέσος χρόνος ολοκλήρωσης** όταν ακολουθείται η πολιτική Round Robin με περίοδο $T=2$ sec, όπου σε κάθε διεργασία που μπαίνει στην ουρά δίνεται η μικρότερη προτεραιότητα. (Υποθέστε μηδενικό χρόνο μεταγωγής, δηλαδή $\delta=0$)

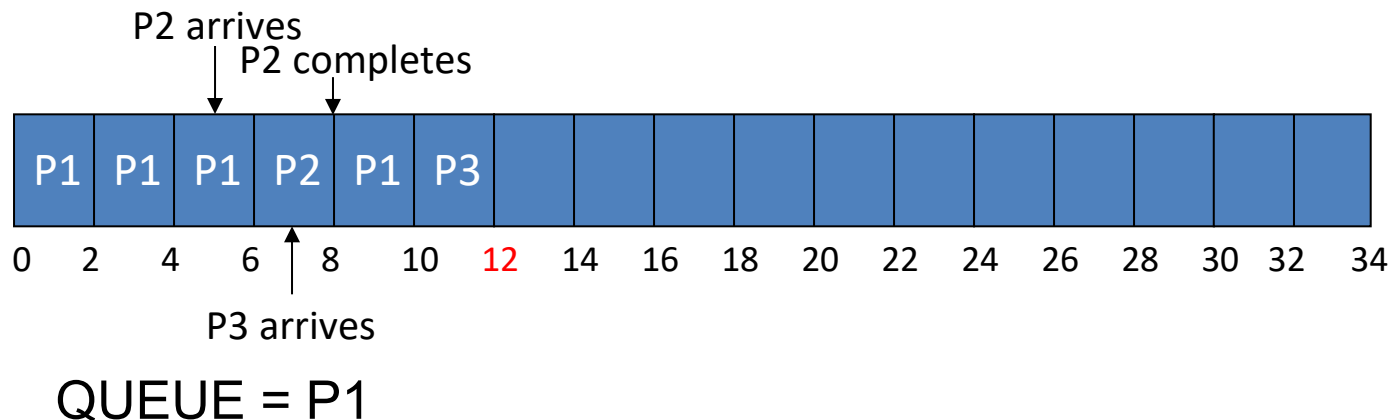
Διεργασία	Χρονική στιγμή άφιξης	Διάρκεια διεργασίας
P1	0 sec	12 sec, remain=4
P2	5 sec	2 sec, remain=0
P3	7 sec	14 sec
P4	13 sec	6 sec



Κεφάλαιο 3 (Λειτουργικά Συστήματα) – TIME=12

- Δίνονται οι διεργασίες P1, P2, P3, P4 με χρονική στιγμή άφιξης και διάρκεια όπως φαίνεται στον παρακάτω πίνακα. Να υπολογιστεί ο **μέσος χρόνος ολοκλήρωσης** όταν ακολουθείται η πολιτική Round Robin με περίοδο $T=2$ sec, όπου σε κάθε διεργασία που μπαίνει στην ουρά δίνεται η μικρότερη προτεραιότητα. (Υποθέστε μηδενικό χρόνο μεταγωγής, δηλαδή $\delta=0$)

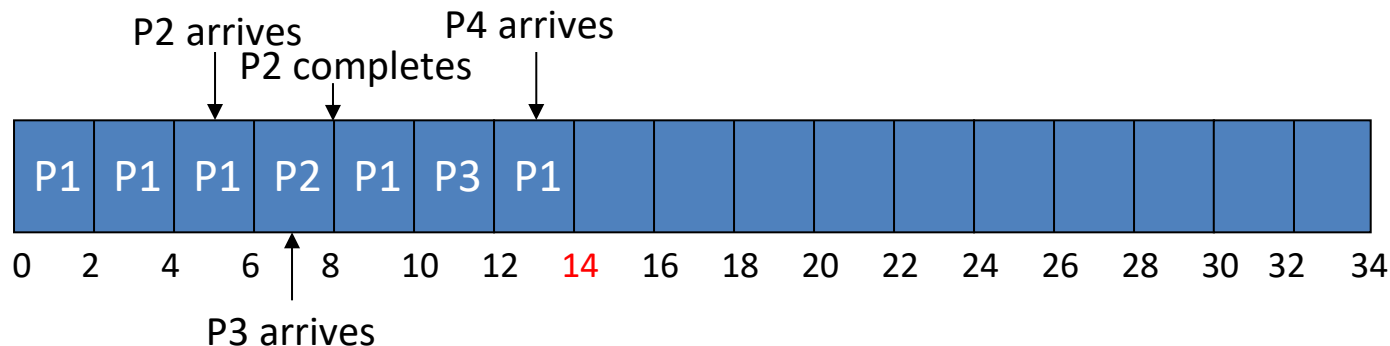
Διεργασία	Χρονική στιγμή άφιξης	Διάρκεια διεργασίας
P1	0 sec	12 sec, remain=4
P2	5 sec	2 sec, remain=0
P3	7 sec	14 sec, remain=12
P4	13 sec	6 sec



Κεφάλαιο 3 (Λειτουργικά Συστήματα) – TIME=14

- Δίνονται οι διεργασίες P1, P2, P3, P4 με χρονική στιγμή άφιξης και διάρκεια όπως φαίνεται στον παρακάτω πίνακα. Να υπολογιστεί ο **μέσος χρόνος ολοκλήρωσης** όταν ακολουθείται η πολιτική Round Robin με περίοδο $T=2$ sec, όπου σε κάθε διεργασία που μπαίνει στην ουρά δίνεται η μικρότερη προτεραιότητα. (Υποθέστε μηδενικό χρόνο μεταγωγής, δηλαδή $\delta=0$)

Διεργασία	Χρονική στιγμή άφιξης	Διάρκεια διεργασίας
P1	0 sec	12 sec, remain=2
P2	5 sec	2 sec, remain=0
P3	7 sec	14 sec, remain=12
P4	13 sec	6 sec

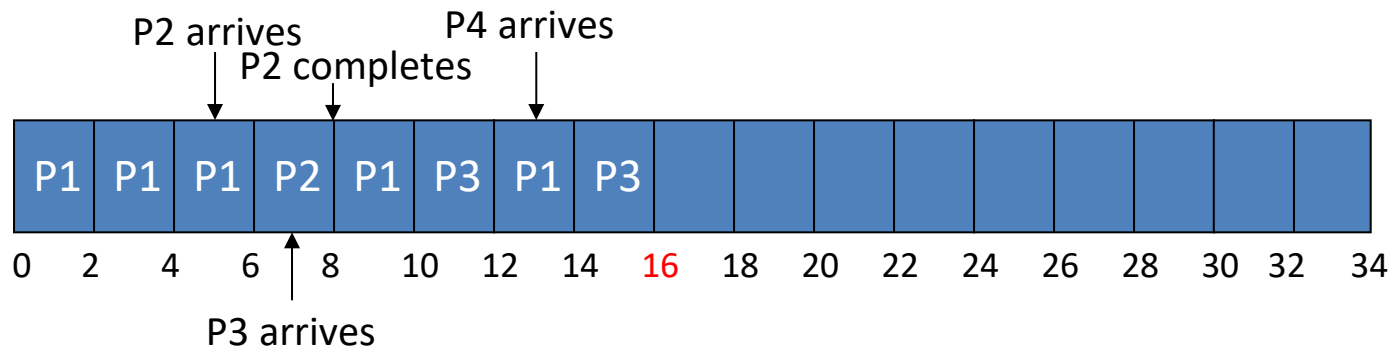


QUEUE = P4, P3

Κεφάλαιο 3 (Λειτουργικά Συστήματα) – TIME=16

- Δίνονται οι διεργασίες P1, P2, P3, P4 με χρονική στιγμή άφιξης και διάρκεια όπως φαίνεται στον παρακάτω πίνακα. Να υπολογιστεί ο **μέσος χρόνος ολοκλήρωσης** όταν ακολουθείται η πολιτική Round Robin με περίοδο $T=2$ sec, όπου σε κάθε διεργασία που μπαίνει στην ουρά δίνεται η μικρότερη προτεραιότητα. (Υποθέστε μηδενικό χρόνο μεταγωγής, δηλαδή $\delta=0$)

Διεργασία	Χρονική στιγμή άφιξης	Διάρκεια διεργασίας
P1	0 sec	12 sec, remain=2
P2	5 sec	2 sec, remain=0
P3	7 sec	14 sec, remain=10
P4	13 sec	6 sec

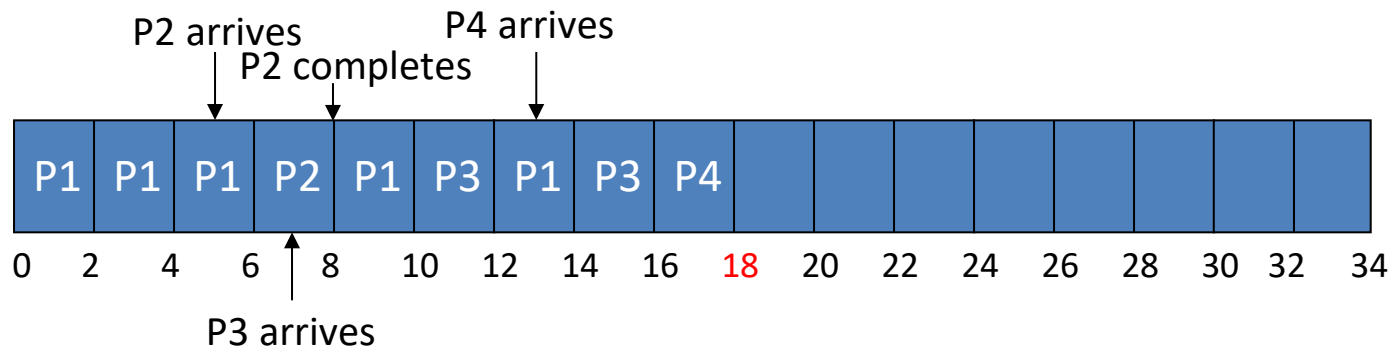


QUEUE = P1, P4

Κεφάλαιο 3 (Λειτουργικά Συστήματα) – TIME=18

- Δίνονται οι διεργασίες P1, P2, P3, P4 με χρονική στιγμή άφιξης και διάρκεια όπως φαίνεται στον παρακάτω πίνακα. Να υπολογιστεί ο **μέσος χρόνος ολοκλήρωσης** όταν ακολουθείται η πολιτική Round Robin με περίοδο $T=2$ sec, όπου σε κάθε διεργασία που μπαίνει στην ουρά δίνεται η μικρότερη προτεραιότητα. (Υποθέστε μηδενικό χρόνο μεταγωγής, δηλαδή $\delta=0$)

Διεργασία	Χρονική στιγμή άφιξης	Διάρκεια διεργασίας
P1	0 sec	12 sec, remain=2
P2	5 sec	2 sec, remain=0
P3	7 sec	14 sec, remain=10
P4	13 sec	6 sec, remain=4

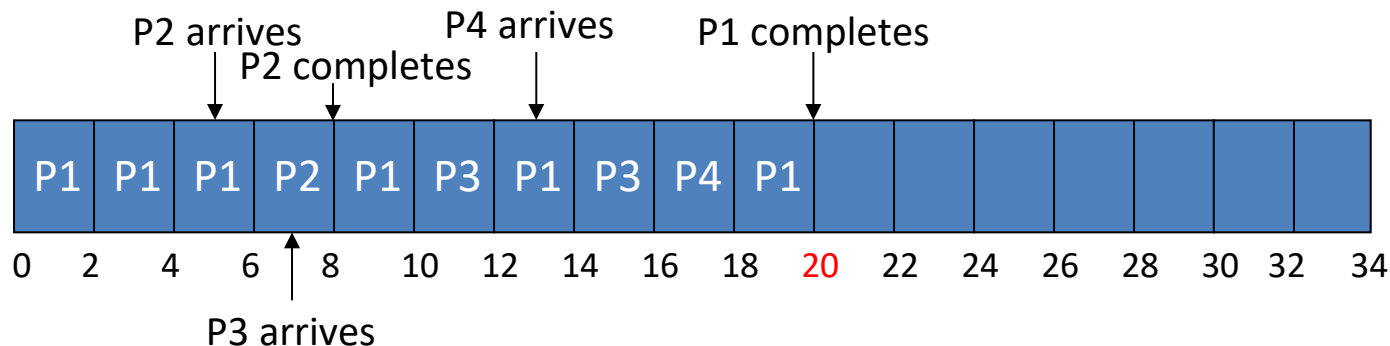


QUEUE = P3, P1

Κεφάλαιο 3 (Λειτουργικά Συστήματα) – TIME=20

- Δίνονται οι διεργασίες P1, P2, P3, P4 με χρονική στιγμή άφιξης και διάρκεια όπως φαίνεται στον παρακάτω πίνακα. Να υπολογιστεί ο **μέσος χρόνος ολοκλήρωσης** όταν ακολουθείται η πολιτική Round Robin με περίοδο $T=2$ sec, όπου σε κάθε διεργασία που μπαίνει στην ουρά δίνεται η μικρότερη προτεραιότητα. (Υποθέστε μηδενικό χρόνο μεταγωγής, δηλαδή $\delta=0$)

Διεργασία	Χρονική στιγμή άφιξης	Διάρκεια διεργασίας
P1	0 sec	12 sec, remain=0
P2	5 sec	2 sec, remain=0
P3	7 sec	14 sec, remain=10
P4	13 sec	6 sec, remain=4

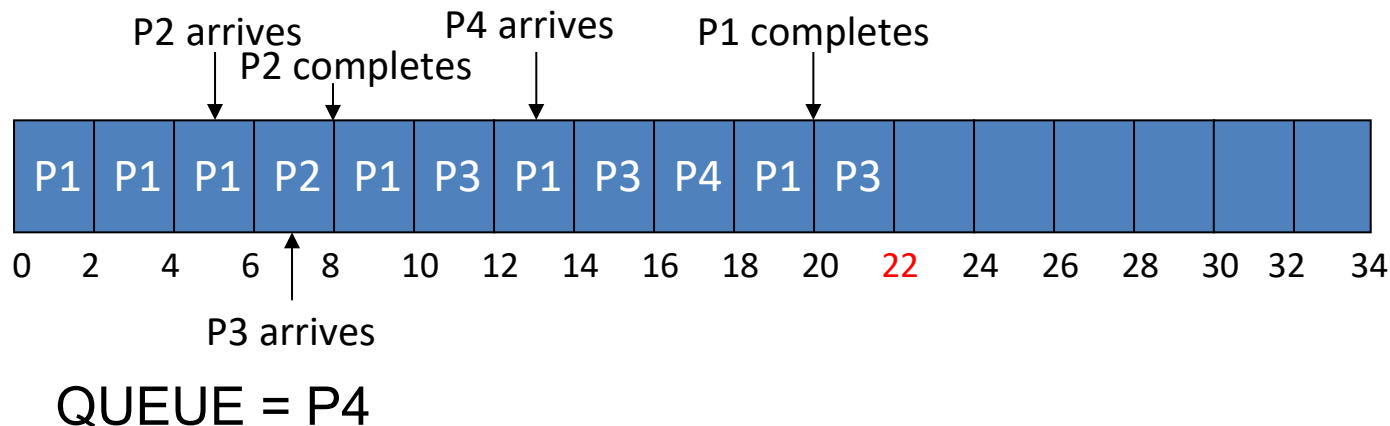


QUEUE = P4, P3

Κεφάλαιο 3 (Λειτουργικά Συστήματα) – TIME=22

- Δίνονται οι διεργασίες P1, P2, P3, P4 με χρονική στιγμή άφιξης και διάρκεια όπως φαίνεται στον παρακάτω πίνακα. Να υπολογιστεί ο **μέσος χρόνος ολοκλήρωσης** όταν ακολουθείται η πολιτική Round Robin με περίοδο $T=2$ sec, όπου σε κάθε διεργασία που μπαίνει στην ουρά δίνεται η μικρότερη προτεραιότητα. (Υποθέστε μηδενικό χρόνο μεταγωγής, δηλαδή $\delta=0$)

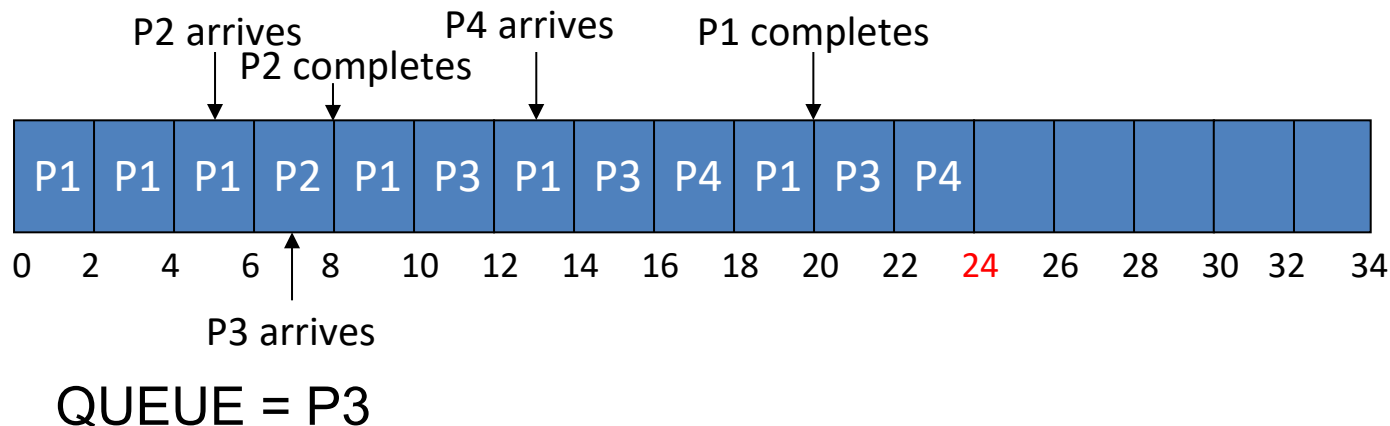
Διεργασία	Χρονική στιγμή άφιξης	Διάρκεια διεργασίας
P1	0 sec	12 sec, remain=0
P2	5 sec	2 sec, remain=0
P3	7 sec	14 sec, remain=8
P4	13 sec	6 sec, remain=4



Κεφάλαιο 3 (Λειτουργικά Συστήματα) – TIME=24

- Δίνονται οι διεργασίες P1, P2, P3, P4 με χρονική στιγμή άφιξης και διάρκεια όπως φαίνεται στον παρακάτω πίνακα. Να υπολογιστεί ο **μέσος χρόνος ολοκλήρωσης** όταν ακολουθείται η πολιτική Round Robin με περίοδο $T=2$ sec, όπου σε κάθε διεργασία που μπαίνει στην ουρά δίνεται η μικρότερη προτεραιότητα. (Υποθέστε μηδενικό χρόνο μεταγωγής, δηλαδή $\delta=0$)

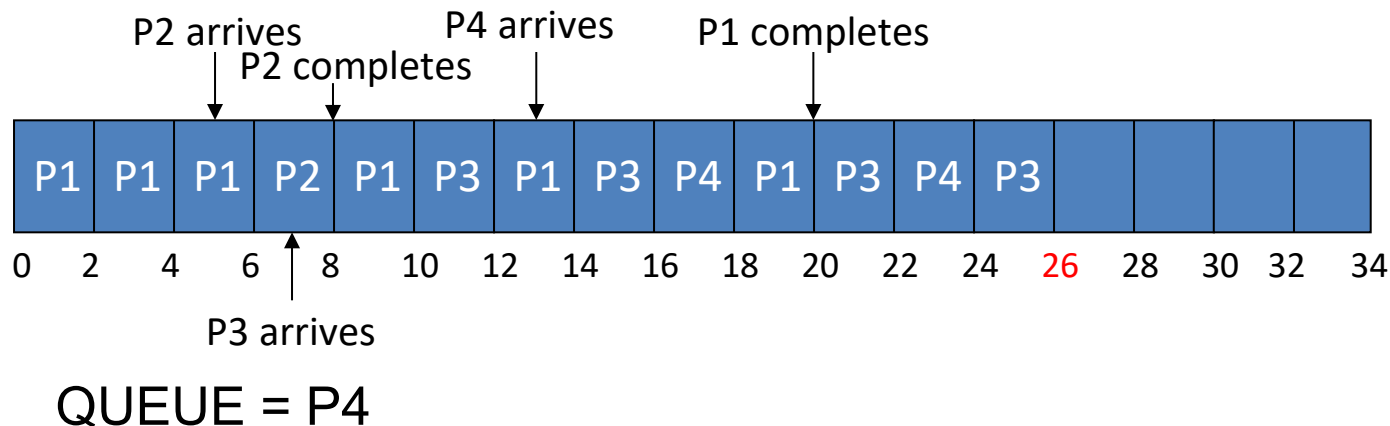
Διεργασία	Χρονική στιγμή άφιξης	Διάρκεια διεργασίας
P1	0 sec	12 sec, remain=0
P2	5 sec	2 sec, remain=0
P3	7 sec	14 sec, remain=6
P4	13 sec	6 sec, remain=2



Κεφάλαιο 3 (Λειτουργικά Συστήματα) – TIME=26

- Δίνονται οι διεργασίες P1, P2, P3, P4 με χρονική στιγμή άφιξης και διάρκεια όπως φαίνεται στον παρακάτω πίνακα. Να υπολογιστεί ο **μέσος χρόνος ολοκλήρωσης** όταν ακολουθείται η πολιτική Round Robin με περίοδο $T=2$ sec, όπου σε κάθε διεργασία που μπαίνει στην ουρά δίνεται η μικρότερη προτεραιότητα. (Υποθέστε μηδενικό χρόνο μεταγωγής, δηλαδή $\delta=0$)

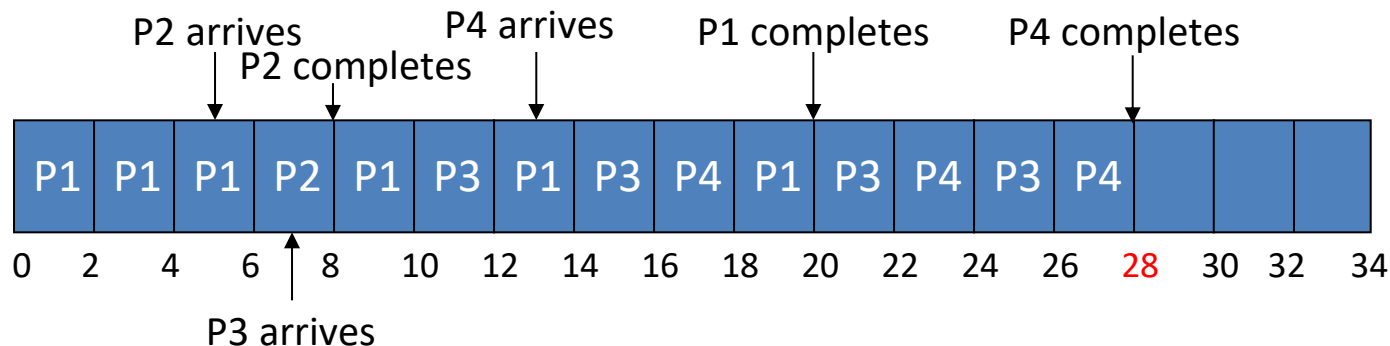
Διεργασία	Χρονική στιγμή άφιξης	Διάρκεια διεργασίας
P1	0 sec	12 sec, remain=0
P2	5 sec	2 sec, remain=0
P3	7 sec	14 sec, remain=6
P4	13 sec	6 sec, remain=2



Κεφάλαιο 3 (Λειτουργικά Συστήματα) – TIME=28

- Δίνονται οι διεργασίες P1, P2, P3, P4 με χρονική στιγμή άφιξης και διάρκεια όπως φαίνεται στον παρακάτω πίνακα. Να υπολογιστεί ο **μέσος χρόνος ολοκλήρωσης** όταν ακολουθείται η πολιτική Round Robin με περίοδο $T=2$ sec, όπου σε κάθε διεργασία που μπαίνει στην ουρά δίνεται η μικρότερη προτεραιότητα. (Υποθέστε μηδενικό χρόνο μεταγωγής, δηλαδή $\delta=0$)

Διεργασία	Χρονική στιγμή άφιξης	Διάρκεια διεργασίας
P1	0 sec	12 sec, remain=0
P2	5 sec	2 sec, remain=0
P3	7 sec	14 sec, remain=6
P4	13 sec	6 sec, remain=0

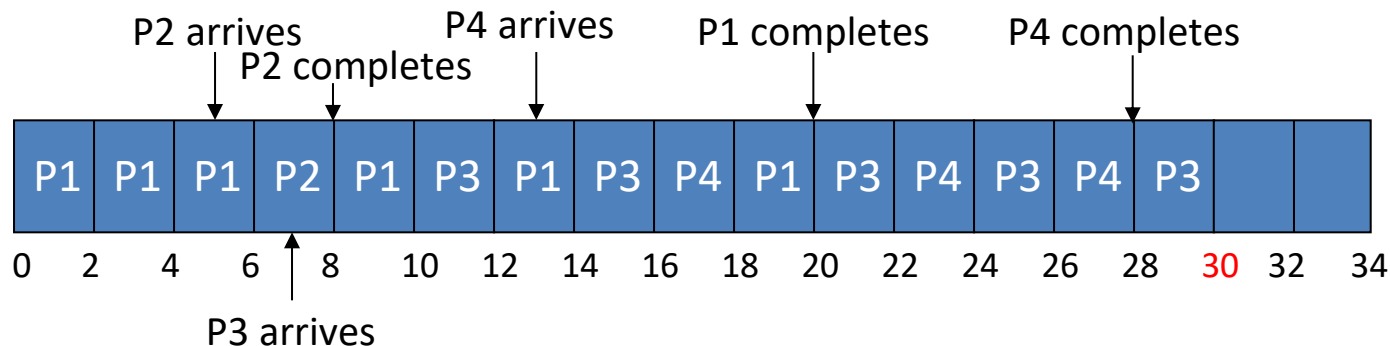


QUEUE = P3

Κεφάλαιο 3 (Λειτουργικά Συστήματα) – TIME=30

- Δίνονται οι διεργασίες P1, P2, P3, P4 με χρονική στιγμή άφιξης και διάρκεια όπως φαίνεται στον παρακάτω πίνακα. Να υπολογιστεί ο **μέσος χρόνος ολοκλήρωσης** όταν ακολουθείται η πολιτική Round Robin με περίοδο $T=2$ sec, όπου σε κάθε διεργασία που μπαίνει στην ουρά δίνεται η μικρότερη προτεραιότητα. (Υποθέστε μηδενικό χρόνο μεταγωγής, δηλαδή $\delta=0$)

Διεργασία	Χρονική στιγμή άφιξης	Διάρκεια διεργασίας
P1	0 sec	12 sec, remain=0
P2	5 sec	2 sec, remain=0
P3	7 sec	14 sec, remain=4
P4	13 sec	6 sec, remain=0

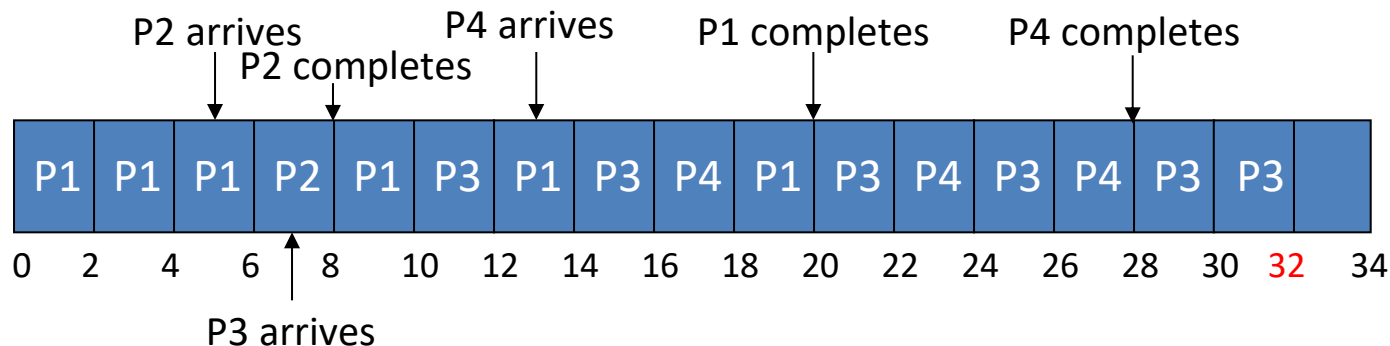


QUEUE = EMPTY

Κεφάλαιο 3 (Λειτουργικά Συστήματα) – TIME=32

- Δίνονται οι διεργασίες P1, P2, P3, P4 με χρονική στιγμή άφιξης και διάρκεια όπως φαίνεται στον παρακάτω πίνακα. Να υπολογιστεί ο **μέσος χρόνος ολοκλήρωσης** όταν ακολουθείται η πολιτική Round Robin με περίοδο $T=2$ sec, όπου σε κάθε διεργασία που μπαίνει στην ουρά δίνεται η μικρότερη προτεραιότητα. (Υποθέστε μηδενικό χρόνο μεταγωγής, δηλαδή $\delta=0$)

Διεργασία	Χρονική στιγμή άφιξης	Διάρκεια διεργασίας
P1	0 sec	12 sec, remain=0
P2	5 sec	2 sec, remain=0
P3	7 sec	14 sec, remain=2
P4	13 sec	6 sec, remain=0

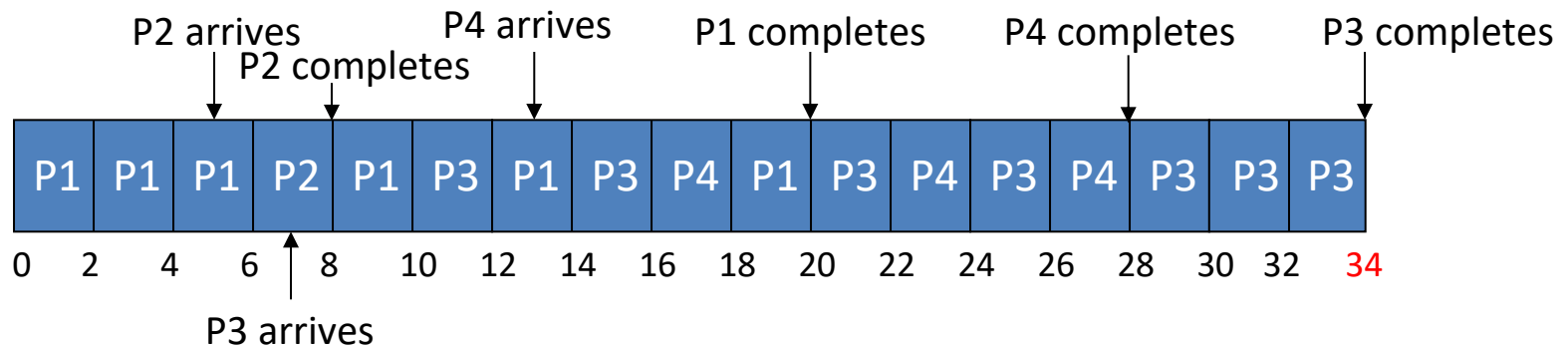


QUEUE = EMPTY

Κεφάλαιο 3 (Λειτουργικά Συστήματα) – TIME=34

- Δίνονται οι διεργασίες P1, P2, P3, P4 με χρονική στιγμή άφιξης και διάρκεια όπως φαίνεται στον παρακάτω πίνακα. Να υπολογιστεί ο **μέσος χρόνος ολοκλήρωσης** όταν ακολουθείται η πολιτική Round Robin με περίοδο $T=2$ sec, όπου σε κάθε διεργασία που μπαίνει στην ουρά δίνεται η μικρότερη προτεραιότητα. (Υποθέστε μηδενικό χρόνο μεταγωγής, δηλαδή $\delta=0$)

Διεργασία	Χρονική στιγμή άφιξης	Διάρκεια διεργασίας
P1	0 sec	12 sec, remain=0
P2	5 sec	2 sec, remain=0
P3	7 sec	14 sec, remain=0
P4	13 sec	6 sec, remain=0

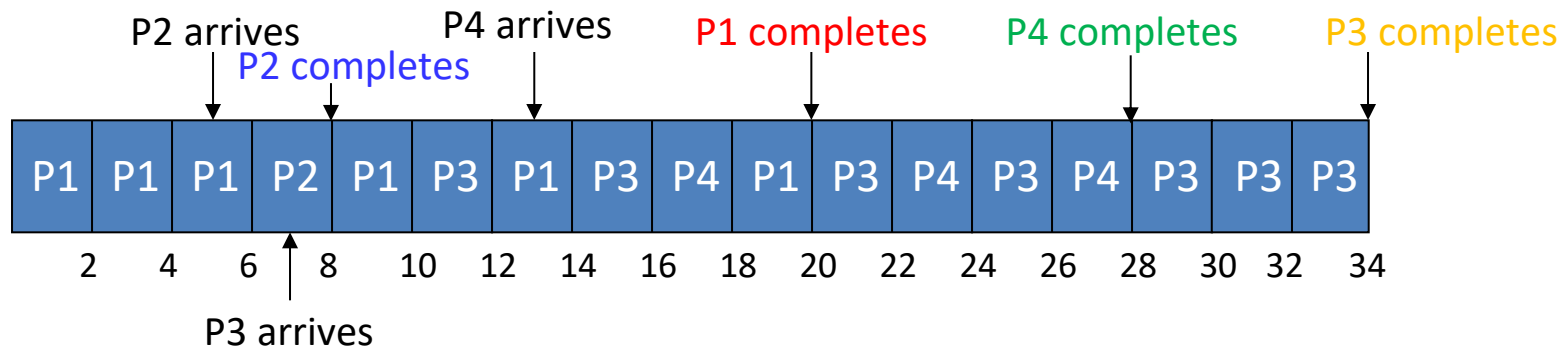


QUEUE = EMPTY

Κεφάλαιο 3 (Λειτουργικά Συστήματα)

- Δίνονται οι διεργασίες P1, P2, P3, P4 με χρονική στιγμή άφιξης και διάρκεια όπως φαίνεται στον παρακάτω πίνακα. Να υπολογιστεί ο **μέσος χρόνος ολοκλήρωσης** όταν ακολουθείται η πολιτική Round Robin με περίοδο $T=2$ sec, όπου σε κάθε διεργασία που μπαίνει στην ουρά δίνεται η μικρότερη προτεραιότητα. (Υποθέστε μηδενικό χρόνο μεταγωγής, δηλαδή $\delta=0$)

Διεργασία	Χρονική στιγμή άφιξης	Διάρκεια διεργασίας
P1	0 sec	12 sec
P2	5 sec	2 sec
P3	7 sec	14 sec
P4	13 sec	6 sec



- Μέσος χρόνος ολοκλήρωσης: $((20-0)+(8-5)+(34-7)+(28-13))/4=16.25$

Κεφάλαιο 4 (Δικτύωση και Διαδίκτυο)

- Ποια είναι η **βασική** διαφορά των πρωτοκόλλων **CSMA/CD** (Carrier Sense Multiple Access / Collision Detection) και **CSMA/CA** (Carrier Sense Multiple Access / Collision Avoidance)

Κεφάλαιο 4 (Δικτύωση και Διαδίκτυο)

- Ποια είναι η **βασική** διαφορά των πρωτοκόλλων **CSMA/CD** (Carrier Sense Multiple Access / Collision Detection) και **CSMA/CA** (Carrier Sense Multiple Access / Collision Avoidance)
- Στο CSMA/CD κάθε κόμβος ακούει συνέχεια το κανάλι και ανιχνεύει τυχόν σύγκρουση καθώς μεταδίδει. Σταματά για ένα τυχαίο σύντομο χρονικό διάστημα πριν ξαναπροσπαθήσει.
- Στο CSMA/CA οι κόμβοι δεν μπορούν να ανιχνεύσουν τις συγκρούσεις, αντ' αυτού τις αποφεύγουν ζητώντας επιβεβαίωση από το δέκτη. Όταν παρουσιαστεί σύγκρουση, το μήνυμα αναμεταδίδεται.

Κεφάλαιο 4 (Δικτύωση και Διαδίκτυο)

- Έστω ένας κόμβος A θέλει να στείλει **100 πακέτα** σε έναν κόμβο B. Έστω ότι ο κόμβος B λαμβάνει τα πακέτα με ρυθμό **5 πακέτα/sec** και έχει μία ουρά που χωράει **30 πακέτα**. Αρχικά ο κόμβος A στέλνει τα πακέτα με ρυθμό **10 πακέτα/sec**. Μετά από **2 sec** ο κόμβος A ενημερώνεται για το ρυθμό λήψης των πακέτων από τον κόμβο B και για τη χωρητικότητα της ουράς του B. Ποιος είναι ο **μέγιστος σταθερός ρυθμός** που πρέπει να στέλνει από δω και πέρα τα πακέτα ο κόμβος A ώστε να μη χαθεί κανένα πακέτο.

Κεφάλαιο 4 (Δικτύωση και Διαδίκτυο)

- Έστω ένας κόμβος A θέλει να στείλει **100 πακέτα** σε έναν κόμβο B. Έστω ότι ο κόμβος B λαμβάνει τα πακέτα με ρυθμό **5 πακέτα/sec** και έχει μία ουρά που χωράει **30 πακέτα**. Αρχικά ο κόμβος A στέλνει τα πακέτα με ρυθμό **10 πακέτα/sec**. Μετά από **2 sec** ο κόμβος A ενημερώνεται για το ρυθμό λήψης των πακέτων από τον κόμβο B και για τη χωρητικότητα της ουράς του B. Ποιος είναι ο **μέγιστος σταθερός ρυθμός** που πρέπει να στέλνει από δω και πέρα τα πακέτα ο κόμβος A ώστε να μη χαθεί κανένα πακέτο.
- Ιδέα: Έστω ότι μετά τα 2 sec ο A στέλνει με ρυθμό $x \geq 5$ πακέτα/sec. Για το μέγιστο x , θα πρέπει η ουρά του B να γεμίσει τελείως όταν ο A στείλει όλα τα πακέτα, έτσι ώστε να μην ξεχειλίσει η ουρά του B.

Κεφάλαιο 4 (Δικτύωση και Διαδίκτυο)

- Έστω ένας κόμβος A θέλει να στείλει **100 πακέτα** σε έναν κόμβο B. Έστω ότι ο κόμβος B λαμβάνει τα πακέτα με ρυθμό **5 πακέτα/sec** και έχει μία ουρά που χωράει **30 πακέτα**. Αρχικά ο κόμβος A στέλνει τα πακέτα με ρυθμό **10 πακέτα/sec**. Μετά από **2 sec** ο κόμβος A ενημερώνεται για το ρυθμό λήψης των πακέτων από τον κόμβο B και για τη χωρητικότητα της ουράς του B. Ποιος είναι ο **μέγιστος σταθερός ρυθμός** που πρέπει να στέλνει από δω και πέρα τα πακέτα ο κόμβος A ώστε να μη χαθεί κανένα πακέτο.
- Ιδέα: Έστω ότι μετά τα 2 sec ο A στέλνει με ρυθμό $x \geq 5$ πακέτα/sec. Για το μέγιστο x , θα πρέπει η ουρά του B να γεμίσει τελείως όταν ο A στείλει όλα τα πακέτα, έτσι ώστε να μην ξεχειλίσει η ουρά του B.
- Μετά από 2 sec η ουρά του B έχει γεμίσει με $(10-5) \cdot 2 = 10$ πακέτα, και έχει περιθώριο άλλα 20 πακέτα.

Κεφάλαιο 4 (Δικτύωση και Διαδίκτυο)

- Έστω ένας κόμβος A θέλει να στείλει **100 πακέτα** σε έναν κόμβο B. Έστω ότι ο κόμβος B λαμβάνει τα πακέτα με ρυθμό **5 πακέτα/sec** και έχει μία ουρά που χωράει **30 πακέτα**. Αρχικά ο κόμβος A στέλνει τα πακέτα με ρυθμό **10 πακέτα/sec**. Μετά από **2 sec** ο κόμβος A ενημερώνεται για το ρυθμό λήψης των πακέτων από τον κόμβο B και για τη χωρητικότητα της ουράς του B. Ποιος είναι ο **μέγιστος σταθερός ρυθμός** που πρέπει να στέλνει από δω και πέρα τα πακέτα ο κόμβος A ώστε να μη χαθεί κανένα πακέτο.
- Ιδέα: Έστω ότι μετά τα 2 sec ο A στέλνει με ρυθμό $x \geq 5$ πακέτα/sec. Για το μέγιστο x , θα πρέπει η ουρά του B να γεμίσει τελείως όταν ο A στείλει όλα τα πακέτα, έτσι ώστε να μην ξεχειλίζει η ουρά του B.
- Μετά από 2 sec η ουρά του B έχει γεμίσει με $(10-5) \cdot 2 = 10$ πακέτα, και έχει περιθώριο άλλα 20 πακέτα.
- Η ουρά του B μετά τα 2 sec γεμίζει με ρυθμό $x-5$ πακέτα/sec.

Κεφάλαιο 4 (Δικτύωση και Διαδίκτυο)

- Έστω ένας κόμβος A θέλει να στείλει **100 πακέτα** σε έναν κόμβο B. Έστω ότι ο κόμβος B λαμβάνει τα πακέτα με ρυθμό **5 πακέτα/sec** και έχει μία ουρά που χωράει **30 πακέτα**. Αρχικά ο κόμβος A στέλνει τα πακέτα με ρυθμό **10 πακέτα/sec**. Μετά από **2 sec** ο κόμβος A ενημερώνεται για το ρυθμό λήψης των πακέτων από τον κόμβο B και για τη χωρητικότητα της ουράς του B. Ποιος είναι ο **μέγιστος σταθερός ρυθμός** που πρέπει να στέλνει από δω και πέρα τα πακέτα ο κόμβος A ώστε να μη χαθεί κανένα πακέτο.
- Ιδέα: Έστω ότι μετά τα 2 sec ο A στέλνει με ρυθμό $x \geq 5$ πακέτα/sec. Για το μέγιστο x , θα πρέπει η ουρά του B να γεμίσει τελείως όταν ο A στείλει όλα τα πακέτα, έτσι ώστε να μην ξεχειλίζει η ουρά του B.
- Μετά από 2 sec η ουρά του B έχει γεμίσει με $(10-5) \cdot 2 = 10$ πακέτα, και έχει περιθώριο άλλα 20 πακέτα.
- Η ουρά του B μετά τα 2 sec γεμίζει με ρυθμό $x-5$ πακέτα/sec.
- Άρα η ουρά θα γεμίσει μετά από $t = 20/(x-5)$ sec, και άρα στα t sec θα πρέπει ο A να έχει στείλει όλα τα πακέτα.

Κεφάλαιο 4 (Δικτύωση και Διαδίκτυο)

- Έστω ένας κόμβος A θέλει να στείλει **100 πακέτα** σε έναν κόμβο B. Έστω ότι ο κόμβος B λαμβάνει τα πακέτα με ρυθμό **5 πακέτα/sec** και έχει μία ουρά που χωράει **30 πακέτα**. Αρχικά ο κόμβος A στέλνει τα πακέτα με ρυθμό **10 πακέτα/sec**. Μετά από **2 sec** ο κόμβος A ενημερώνεται για το ρυθμό λήψης των πακέτων από τον κόμβο B και για τη χωρητικότητα της ουράς του B. Ποιος είναι ο **μέγιστος σταθερός ρυθμός** που πρέπει να στέλνει από δω και πέρα τα πακέτα ο κόμβος A ώστε να μη χαθεί κανένα πακέτο.
- Ιδέα: Έστω ότι μετά τα 2 sec ο A στέλνει με ρυθμό $x \geq 5$ πακέτα/sec. Για το μέγιστο x , θα πρέπει η ουρά του B να γεμίσει τελείως όταν ο A στείλει όλα τα πακέτα, έτσι ώστε να μην ξεχειλίζει η ουρά του B.
- Μετά από 2 sec η ουρά του B έχει γεμίσει με $(10-5) \cdot 2 = 10$ πακέτα, και έχει περιθώριο άλλα 20 πακέτα.
- Η ουρά του B μετά τα 2 sec γεμίζει με ρυθμό $x-5$ πακέτα/sec.
- Άρα η ουρά θα γεμίσει μετά από $t = 20/(x-5)$ sec, και άρα στα t sec θα πρέπει ο A να έχει στείλει όλα τα πακέτα.
- Ο A πρέπει να στείλει ακόμα 80 πακέτα, άρα θέλει χρόνο $80/x$ sec. Άρα $t = 80/x$.

Κεφάλαιο 4 (Δικτύωση και Διαδίκτυο)

- Έστω ένας κόμβος A θέλει να στείλει **100 πακέτα** σε έναν κόμβο B. Έστω ότι ο κόμβος B λαμβάνει τα πακέτα με ρυθμό **5 πακέτα/sec** και έχει μία ουρά που χωράει **30 πακέτα**. Αρχικά ο κόμβος A στέλνει τα πακέτα με ρυθμό **10 πακέτα/sec**. Μετά από **2 sec** ο κόμβος A ενημερώνεται για το ρυθμό λήψης των πακέτων από τον κόμβο B και για τη χωρητικότητα της ουράς του B. Ποιος είναι ο **μέγιστος σταθερός ρυθμός** που πρέπει να στέλνει από δω και πέρα τα πακέτα ο κόμβος A ώστε να μη χαθεί κανένα πακέτο.
- Ιδέα: Έστω ότι μετά τα 2 sec ο A στέλνει με ρυθμό $x \geq 5$ πακέτα/sec. Για το μέγιστο x , θα πρέπει η ουρά του B να γεμίσει τελείως όταν ο A στείλει όλα τα πακέτα, έτσι ώστε να μην ξεχειλίζει η ουρά του B.
- Μετά από 2 sec η ουρά του B έχει γεμίσει με $(10-5) \cdot 2 = 10$ πακέτα, και έχει περιθώριο άλλα 20 πακέτα.
- Η ουρά του B μετά τα 2 sec γεμίζει με ρυθμό $x-5$ πακέτα/sec.
- Άρα η ουρά θα γεμίσει μετά από $t = 20/(x-5)$ sec, και άρα στα t sec θα πρέπει ο A να έχει στείλει όλα τα πακέτα.
- Ο A πρέπει να στείλει ακόμα 80 πακέτα, άρα θέλει χρόνο $80/x$ sec. Άρα $t = 80/x$.
- **Λύνοντας $20/(x-5) = 80/x$ βρίσκουμε $x = 20/3$!**

Κεφάλαιο 5 (Αλγόριθμοι)

- Πόσους αριθμούς εκτυπώνει η παρακάτω συνάρτηση με είσοδο κάποιο $n=2^k$ για κάποιο ακέραιο $k>0$; Ποια είναι η πολυπλοκότητά της;

Συνάρτηση $f(n)$

if $n \geq 1$

for $i=1$ to n do

print(i)

$f(n/2)$

Κεφάλαιο 5 (Αλγόριθμοι)

- Πόσους αριθμούς εκτυπώνει η παρακάτω συνάρτηση με είσοδο κάποιο $n=2^k$ για κάποιο ακέραιο $k>0$; Ποια είναι η πολυπλοκότητά της;

Συνάρτηση $f(n)$

```
if  $n \geq 1$ 
    for  $i=1$  to  $n$  do
        print( $i$ )
    f( $n/2$ )
```

- Η συνάρτηση καλείται μία φορά για το n , 1 φορά για το $n/2$, 1 φορά για το $n/4$, κ.ο.κ. και συνολικά, έως ότου ισχύει $n \geq 1$, άρα 1 η τελευταία τιμή του n που θα εκτελέσει το for και η τελευταία τιμή είναι $\frac{1}{2}$ που δεν εκτελέσει το for.

Κεφάλαιο 5 (Αλγόριθμοι)

- Πόσους αριθμούς εκτυπώνει η παρακάτω συνάρτηση με είσοδο κάποιο $n=2^k$ για κάποιο ακέραιο $k>0$; Ποια είναι η πολυπλοκότητά της;

Συνάρτηση $f(n)$

if $n \geq 1$

for $i=1$ to n do

print(i)

$f(n/2)$

- Η συνάρτηση καλείται μία φορά για το n , 1 φορά για το $n/2$, 1 φορά για το $n/4$, κ.ο.κ. και συνολικά, έως ότου ισχύει $n \geq 1$, άρα 1 η τελευταία τιμή του n που θα εκτελέσει το for και η τελευταία τιμή είναι $\frac{1}{2}$ που δεν εκτελέσει το for.
- Το n θα γίνει 1 όταν $2^k/2^k=2^0=1$ δηλ. την k φορά. Επομένως καλείται για 2^k , $2^k/2=2^{k-1}$, $2^{k-1}/2=2^{k-2}$... , $2^{k-(k-1)}/2=2^0$, δηλαδή 1 φορά με είσοδο το $n/2^r = 2^{k-r}$ για r από 0 έως k (καλείται και με είσοδο $\frac{1}{2}$, αλλά δεν εκτυπώνει τίποτα).

Κεφάλαιο 5 (Αλγόριθμοι)

- Πόσους αριθμούς εκτυπώνει η παρακάτω συνάρτηση με είσοδο κάποιο $n=2^k$ για κάποιο ακέραιο $k>0$; Ποια είναι η πολυπλοκότητά της;

Συνάρτηση $f(n)$

if $n \geq 1$

for $i=1$ to n do

print(i)

$f(n/2)$

- Η συνάρτηση καλείται μία φορά για το n , 1 φορά για το $n/2$, 1 φορά για το $n/4$, κ.ο.κ. και συνολικά, έως ότου ισχύει $n \geq 1$, άρα 1 η τελευταία τιμή του n που θα εκτελέσει το for και η τελευταία τιμή είναι $\frac{1}{2}$ που δεν εκτελέσει το for.
- Το n θα γίνει 1 όταν $2^k/2^k=2^0=1$ δηλ. την k φορά. Επομένως καλείται για 2^k , $2^k/2=2^{k-1}$, $2^{k-1}/2=2^{k-2}$... , $2^{k-(k-1)}/2=2^0$, δηλαδή για $2^k/2^r = 2^{k-r}$ για r από 0 έως k (καλείται και με είσοδο $\frac{1}{2}$, αλλά δεν εκτυπώνει τίποτα).
- Κάθε φορά που καλείται η συνάρτηση με είσοδο 2^{k-r} εκτυπώνει 2^{k-r} αριθμούς. Επομένως, οι αριθμοί που εκτυπώνονται συνολικά είναι:

$$\sum_{r=0}^k 2^{k-r} = \sum_{i=0}^k 2^i = 2^{k+1} - 1$$

Κεφάλαιο 5 (Αλγόριθμοι)

- Πόσους αριθμούς εκτυπώνει η παρακάτω συνάρτηση με είσοδο κάποιο $n=2^k$ για κάποιο ακέραιο $k>0$; Ποια είναι η πολυπλοκότητά της;

Συνάρτηση $f(n)$

if $n \geq 1$

for $i=1$ to n do

print(i)

$f(n/2)$

Το for εκτελείται **n φορές**

Υπάρχει **μία αναδρομική κλήση** με όρισμα $n/2$

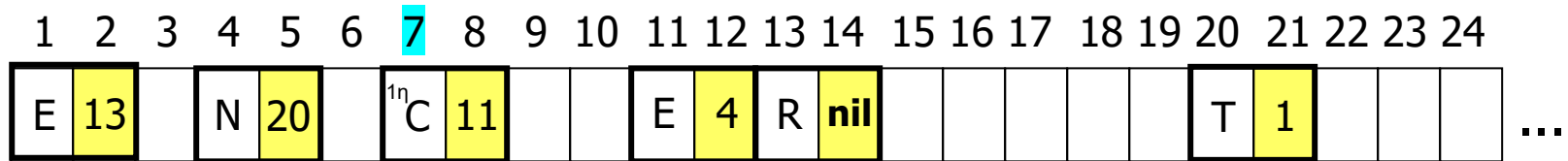
Την 1^η φορά το for θα κάνει n κύκλους, τη 2^η $n/2$, ...

Άρα ο χρόνος εκτέλεσης είναι:

$T(n) = n + n/2 + n/4 + n/8 + \dots = 2n$, δηλ. **$O(n)$**

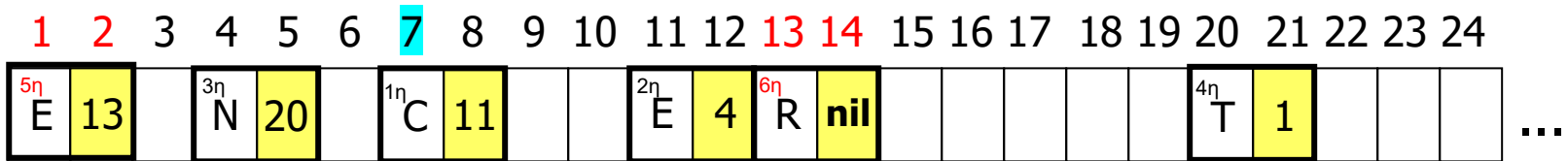
Κεφάλαιο 8 (Δομές Δεδομένων)

- Έστω ότι σας δίνεται η παρακάτω συνδεδεμένη λίστα όπου στα κίτρινα κελιά είναι ο δείκτης της κάθε καταχώρησης. Υποθέστε ότι ο δείκτης κεφαλής είναι στη θέση μνήμης 7.
 - Περιγράψτε πώς μπορούμε να ανταλλάξουμε τη θέση των δύο τελευταίων καταχωρήσεων.
 - Περιγράψτε πώς μπορούμε μετά να προσθέσουμε το σύμβολο “D” σαν τελευταία καταχώρηση.



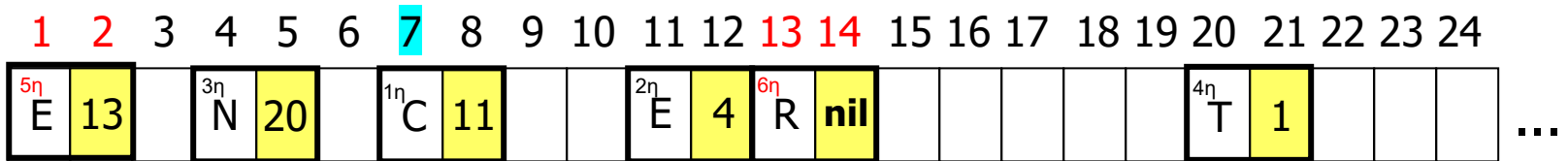
Κεφάλαιο 8 (Δομές Δεδομένων)

- Έστω ότι σας δίνεται η παρακάτω συνδεδεμένη λίστα όπου στα κίτρινα κελιά είναι ο δείκτης της κάθε καταχώρησης. Υποθέστε ότι ο δείκτης κεφαλής είναι στη θέση μνήμης 7.
 - Περιγράψτε πώς μπορούμε να ανταλλάξουμε τη θέση των δύο τελευταίων καταχωρήσεων.
 - Περιγράψτε πώς μπορούμε μετά να προσθέσουμε το σύμβολο “D” σαν τελευταία καταχώρηση.

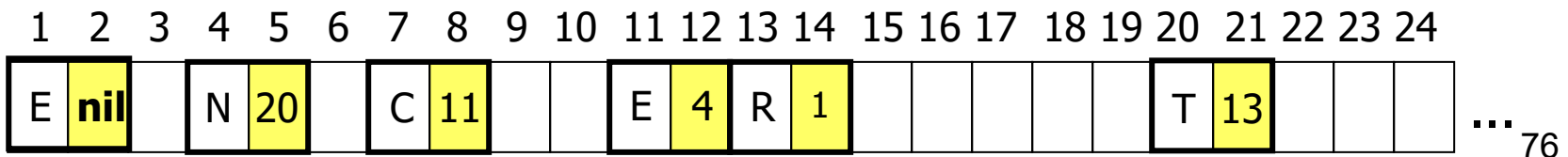


Κεφάλαιο 8 (Δομές Δεδομένων)

- Έστω ότι σας δίνεται η παρακάτω συνδεδεμένη λίστα όπου στα κίτρινα κελιά είναι ο δείκτης της κάθε καταχώρησης. Υποθέστε ότι ο δείκτης κεφαλής είναι στη θέση μνήμης 7.
 - Περιγράψτε πώς μπορούμε να ανταλλάξουμε τη θέση των δύο τελευταίων καταχωρήσεων.
 - Περιγράψτε πώς μπορούμε μετά να προσθέσουμε το σύμβολο “D” σαν τελευταία καταχώρηση.

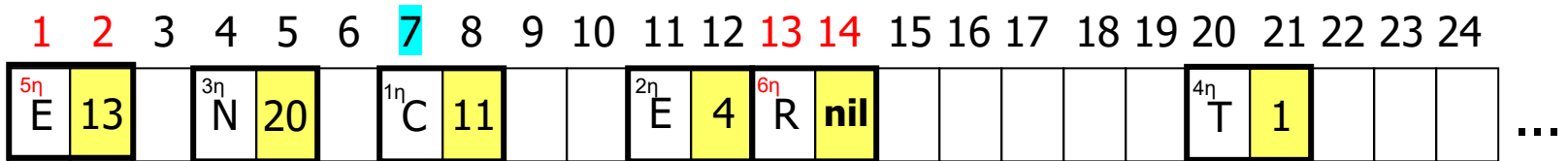


- Στο δείκτη της 4^{ης} καταχώρησης (θέση 21) θα βάλουμε το δείκτη της 5^{ης} καταχώρησης (δηλαδή το 13), έτσι ώστε να δείχνει στην 6^η καταχώρηση. Έτσι η 6^η καταχώρηση γίνεται 5^η. Επίσης στο δείκτη της 6^{ης} καταχώρησης, που έχει γίνει τώρα 5^η, θα βάλουμε το δείκτη που είχε η 4^η καταχώρηση, έτσι ώστε η 5^η καταχώρηση να γίνει 6^η. Επιπλέον στο δείκτη 5^{ης} καταχώρησης που έχει γίνει 6^η θα βάλουμε nil ώστε να τελειώσει η λίστα.

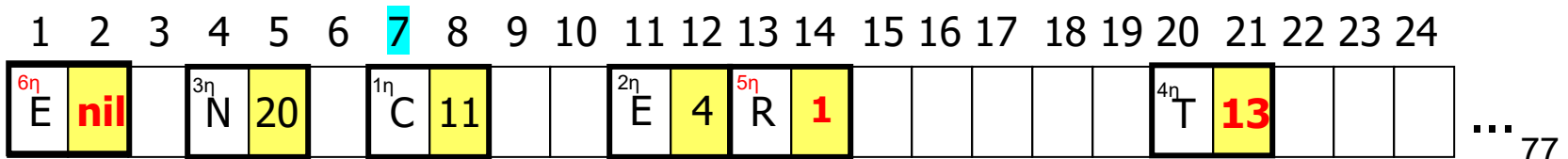


Κεφάλαιο 8 (Δομές Δεδομένων)

- Έστω ότι σας δίνεται η παρακάτω συνδεδεμένη λίστα όπου στα κίτρινα κελιά είναι ο δείκτης της κάθε καταχώρησης. Υποθέστε ότι ο δείκτης κεφαλής είναι στη θέση μνήμης 7.
 - Περιγράψτε πώς μπορούμε να ανταλλάξουμε τη θέση των δύο τελευταίων καταχωρήσεων.
 - Περιγράψτε πώς μπορούμε μετά να προσθέσουμε το σύμβολο “D” σαν τελευταία καταχώρηση.



- Στο δείκτη της 4^{ης} καταχώρησης (θέση 21) θα βάλουμε το δείκτη της 5^{ης} καταχώρησης (δηλαδή το 13), έτσι ώστε να δείχνει στην 6^η καταχώρηση. Έτσι η 6^η καταχώρηση γίνεται 5^η. Επίσης στο δείκτη της 6^{ης} καταχώρησης, που έχει γίνει τώρα 5^η, θα βάλουμε το δείκτη που είχε η 4^η καταχώρηση, έτσι ώστε η 5^η καταχώρηση να γίνει 6^η. Επιπλέον στο δείκτη 5^{ης} καταχώρησης που έχει γίνει 6^η θα βάλουμε nil ώστε να τελειώσει η λίστα.



Κεφάλαιο 9 (Βάσεις Δεδομένων)

Περιγράψτε τι επιστρέφουν οι παρακάτω εντολές σε SQL.

Σχέση EMPLOYEE (ΥΠΑΛΛΗΛΟΣ)

Empl Id (Κωδικός υπαλλήλου)	Name (Όνομα)	Address (Διεύθυνση)	SSN (ΑΦΜ)
25X15	Γιάννης Αθανασίου	Ακαδημίας 1234	111223333
34Y70	Μαρία Γεωργιάδου	Πανεπιστημίου 982	999009999
23Y34	Στέλιος Μπέλλος	Σταδίου 724	111005555
•	•	•	•
•	•	•	•
•	•	•	•

```
select EmplId
from Employee
where Address IS NULL
```

Σχέση JOB (ΘΕΣΗ)

Job Id (Κωδικός θέσης)	Job Title (Τίτλος θέσης)	Skill Code (Βαθμός ικανότητας)	Dept (Τμήμα)
S25X	Γραμματέας	T5	Προσωπικού
S26Z	Γραμματέας	T6	Λογιστήριο
F5	Εμποπτεών ορόφου	FM3	Πωλήσεων
•	•	•	•
•	•	•	•
•	•	•	•

```
select EmplId
from Assignment A, Job J
where A.JobId=J.JobId and
SkillCode='T5' and
Dept='Προσωπικού'
```

Σχέση ASSIGNMENT (ΑΝΑΘΕΣΗ)

Empl Id (Κωδικός υπαλλήλου)	Job Id (Κωδικός θέσης)	Start Date (Ημερομηνία έναρξης)	Term Date (Ημερομηνία τέλους)
23Y34	S25X	1-3-1999	30-4-2001
34Y70	F5	1-10-2002	*
23Y34	S26Z	1-5-2001	*
•	•	•	•
•	•	•	•
•	•	•	•

Κεφάλαιο 9 (Βάσεις Δεδομένων)

Περιγράψτε τι επιστρέφουν οι παρακάτω εντολές σε SQL.

Σχέση EMPLOYEE (ΥΠΑΛΛΗΛΟΣ)

Empl Id (Κωδικός υπαλλήλου)	Name (Όνομα)	Address (Διεύθυνση)	SSN (ΑΦΜ)
25X15	Γιάννης Αθανασίου	Ακαδημίας 1234	111223333
34Y70	Μαρία Γεωργιάδου	Πανεπιστημίου 982	999009999
23Y34	Στέλιος Μπέλλος	Σταδίου 724	111005555
•	•	•	•
•	•	•	•
•	•	•	•

Σχέση JOB (ΘΕΣΗ)

Job Id (Κωδικός θέσης)	Job Title (Τίτλος θέσης)	Skill Code (Βαθμός ικανότητας)	Dept (Τμήμα)
S25X	Γραμματέας	T5	Προσωπικού
S26Z	Γραμματέας	T6	Λογιστήριο
F5	Εμποτεύων ορόφου	FM3	Πωλήσεων
•	•	•	•
•	•	•	•
•	•	•	•

Σχέση ASSIGNMENT (ΑΝΑΘΕΣΗ)

Empl Id (Κωδικός υπαλλήλου)	Job Id (Κωδικός θέσης)	Start Date (Ημερομηνία έναρξης)	Term Date (Ημερομηνία τέλους)
23Y34	S25X	1-3-1999	30-4-2001
34Y70	F5	1-10-2002	*
23Y34	S26Z	1-5-2001	*
•	•	•	•
•	•	•	•
•	•	•	•

```
select EmplId  
from Employee  
where Address IS NULL
```

Επιστρέφει τα Id των υπαλλήλων για τους οποίους δεν είναι καταχωρημένη η διεύθυνσή τους.

```
select EmplId  
from Assignment A, Job J  
where A.JobId=J.JobId and  
SkillCode='T5' and Dept='Προσωπικού'
```

Επιστρέφει τα Id των υπαλλήλων που δουλεύουν ή έχουν δουλέψει στο τμήμα προσωπικού με το skill T5.

Κεφάλαιο 9 (Βάσεις Δεδομένων)

Περιγράψτε τι επιστρέφουν οι παρακάτω εντολές σε SQL.

Σχέση EMPLOYEE (ΥΠΑΛΛΗΛΟΣ)

Empl Id (Κωδικός υπαλλήλου)	Name (Όνομα)	Address (Διεύθυνση)	SSN (ΑΦΜ)
25X15	Γιάννης Αθανασίου	Ακαδημίας 1234	111223333
34Y70	Μαρία Γεωργιάδου	Πανεπιστημίου 982	999009999
23Y34	Στέλιος Μπέλλος	Σταδίου 724	111005555
•	•	•	•
•	•	•	•
•	•	•	•

```
select Dept
from Job J, Assignment A
where J.JobId=A.JobId
```

Σχέση JOB (ΘΕΣΗ)

Job Id (Κωδικός θέσης)	Job Title (Τίτλος θέσης)	Skill Code (Βαθμός ικανότητας)	Dept (Τμήμα)
S25X	Γραμματέας	T5	Προσωπικού
S26Z	Γραμματέας	T6	Λογιστήριο
F5	Εμποπτεύων ορόφου	FM3	Πωλήσεων
•	•	•	•
•	•	•	•
•	•	•	•

```
select Dept
from Job J
where J.JobId NOT IN (select A.JobId
from Assignment A
where TermDate IS NULL)
```

Σχέση ASSIGNMENT (ΑΝΑΘΕΣΗ)

Empl Id (Κωδικός υπαλλήλου)	Job Id (Κωδικός θέσης)	Start Date (Ημερομηνία έναρξης)	Term Date (Ημερομηνία τέλους)
23Y34	S25X	1-3-1999	30-4-2001
34Y70	F5	1-10-2002	*
23Y34	S26Z	1-5-2001	*
•	•	•	•
•	•	•	•
•	•	•	•

Κεφάλαιο 9 (Βάσεις Δεδομένων)

Περιγράψτε τι επιστρέφουν οι παρακάτω εντολές σε SQL.

Σχέση EMPLOYEE (ΥΠΑΛΛΗΛΟΣ)

Empl Id (Κωδικός υπαλλήλου)	Name (Όνομα)	Address (Διεύθυνση)	SSN (ΑΦΜ)
25X15	Γιάννης Αθανασίου	Ακαδημίας 1234	111223333
34Y70	Μαρία Γεωργιάδου	Πανεπιστημίου 982	999009999
23Y34	Στέλιος Μπέλλος	Σταδίου 724	111005555
•	•	•	•
•	•	•	•
•	•	•	•

Σχέση JOB (ΘΕΣΗ)

Job Id (Κωδικός θέσης)	Job Title (Τίτλος θέσης)	Skill Code (Βαθμός ικανότητας)	Dept (Τμήμα)
S25X	Γραμματέας	T5	Προσωπικού
S26Z	Γραμματέας	T6	Λογιστήριο
F5	Εμποπτεύων ορόφου	FM3	Πωλήσεων
•	•	•	•
•	•	•	•
•	•	•	•

Σχέση ASSIGNMENT (ΑΝΑΘΕΣΗ)

Empl Id (Κωδικός υπαλλήλου)	Job Id (Κωδικός θέσης)	Start Date (Ημερομηνία έναρξης)	Term Date (Ημερομηνία τέλους)
23Y34	S25X	1-3-1999	30-4-2001
34Y70	F5	1-10-2002	*
23Y34	S26Z	1-5-2001	*
•	•	•	•
•	•	•	•
•	•	•	•

```
select Dept
from Job J, Assignment A
where J.JobId=A.JobId
```

Επιστρέφει τα τμήματα στα οποία δουλεύει ή έχει δουλέψει τουλάχιστον ένας υπάλληλος.

```
select Dept
from Job J
where J.JobId NOT IN (select A.JobId
from Assignment A
where TermDate IS NULL)
```

Επιστρέφει τα τμήματα στα οποία υπάρχει θέση στην οποία δεν δουλεύει κάποιος υπάλληλος τώρα.

Κεφάλαιο 9 (Βάσεις Δεδομένων)

	Βάση	T1	T2
A	3		
B	5		

Υποθέστε ότι δύο συναλλαγές T1 και T2 εκτελούνται παράλληλα όπως στον παρακάτω πίνακα. Αν αρχικά $A=3$ και $B=5$, ποιες θα είναι οι τελικές τιμές των A, B στη βάση δεδομένων και ποια τιμή θα έχει στο τέλος το product;

Time	T1	T2
1	Read(A)	
2	$A=A-2$	
3		product=1
4		Read(A)
5	Write(A)	
6		product=product*A
7		$A=A-1$
8	Read(B)	
9	$B=B+1$	
10		Write(A)
11	Write(B)	
12		Read(B)
13		product=product*B
14		$B=B-1$

Κεφάλαιο 9 (Βάσεις Δεδομένων)

	Βάση	T1	T2
A	3	1	
B	5		

Υποθέστε ότι δύο συναλλαγές T1 και T2 εκτελούνται παράλληλα όπως στον παρακάτω πίνακα. Αν αρχικά A=3 και B=5, ποιες θα είναι οι τελικές τιμές των A, B στη βάση δεδομένων και ποια τιμή θα έχει στο τέλος το product;

Time	T1	T2
1	Read(A)	
2	A=A-2	
3		product=1
4		Read(A)
5	Write(A)	
6		product=product*A
7		A=A-1
8	Read(B)	
9	B=B+1	
10		Write(A)
11	Write(B)	
12		Read(B)
13		product=product*B
14		B=B-1

Επικοινωνία με τη βάση:

Time 1: Η T1 διαβάζει A=3

Συναλλαγή T1:

Time 2: A=A-2=1

Κεφάλαιο 9 (Βάσεις Δεδομένων)

	Βάση	T1	T2
A	3	1	3
B	5		

Υποθέστε ότι δύο συναλλαγές T1 και T2 εκτελούνται παράλληλα όπως στον παρακάτω πίνακα. Αν αρχικά A=3 και B=5, ποιες θα είναι οι τελικές τιμές των A, B στη βάση δεδομένων και ποια τιμή θα έχει στο τέλος το product;

Time	T1	T2
1	Read(A)	
2	A=A-2	
3		product=1
4		Read(A)
5	Write(A)	
6		product=product*A
7		A=A-1
8	Read(B)	
9	B=B+1	
10		Write(A)
11	Write(B)	
12		Read(B)
13		product=product*B
14		B=B-1

Επικοινωνία με τη βάση:

Time 1: Η T1 διαβάζει A=3

Time 4: Η T2 διαβάζει A=3

Συναλλαγή T1:

Time 2: A=A-2=1

Συναλλαγή T2:

Time 3: product = 1

Κεφάλαιο 9 (Βάσεις Δεδομένων)

	Βάση	T1	T2
A	1	1	3
B	5		

Υποθέστε ότι δύο συναλλαγές T1 και T2 εκτελούνται παράλληλα όπως στον παρακάτω πίνακα. Αν αρχικά A=3 και B=5, ποιες θα είναι οι τελικές τιμές των A, B στη βάση δεδομένων και ποια τιμή θα έχει στο τέλος το product;

Time	T1	T2
1	Read(A)	
2	A=A-2	
3		product=1
4		Read(A)
5	Write(A)	
6		product=product*A
7		A=A-1
8	Read(B)	
9	B=B+1	
10		Write(A)
11	Write(B)	
12		Read(B)
13		product=product*B
14		B=B-1

Επικοινωνία με τη βάση:

Time 1: Η T1 διαβάζει A=3

Time 4: Η T2 διαβάζει A=3

Time 5: Η T1 γράφει A=1 (**update**)

Συναλλαγή T1:

Time 2: A=A-2=1

Συναλλαγή T2:

Time 3: product = 1

Κεφάλαιο 9 (Βάσεις Δεδομένων)

	Βάση	T1	T2
A	1	1	2
B	5		

Υποθέστε ότι δύο συναλλαγές T1 και T2 εκτελούνται παράλληλα όπως στον παρακάτω πίνακα. Αν αρχικά A=3 και B=5, ποιες θα είναι οι τελικές τιμές των A, B στη βάση δεδομένων και ποια τιμή θα έχει στο τέλος το product;

Time	T1	T2
1	Read(A)	
2	A=A-2	
3		product=1
4		Read(A)
5	Write(A)	
6		product=product*A
7		A=A-1
8	Read(B)	
9	B=B+1	
10		Write(A)
11	Write(B)	
12		Read(B)
13		product=product*B
14		B=B-1

Επικοινωνία με τη βάση:

Time 1: Η T1 διαβάζει A=3

Time 4: Η T2 διαβάζει A=3

Time 5: Η T1 γράφει A=1 (**update**)

Συναλλαγή T1:

Time 2: A=A-2=1

Συναλλαγή T2:

Time 3: product = 1

Time 6: product = 3 (είναι 1 * A της T2)

Time 7: A=A-1=2

Κεφάλαιο 9 (Βάσεις Δεδομένων)

	Βάση	T1	T2
A	1	1	2
B	5	6	

Υποθέστε ότι δύο συναλλαγές T1 και T2 εκτελούνται παράλληλα όπως στον παρακάτω πίνακα. Αν αρχικά A=3 και B=5, ποιες θα είναι οι τελικές τιμές των A, B στη βάση δεδομένων και ποια τιμή θα έχει στο τέλος το product;

Time	T1	T2
1	Read(A)	
2	A=A-2	
3		product=1
4		Read(A)
5	Write(A)	
6		product=product*A
7		A=A-1
8	Read(B)	
9	B=B+1	
10		Write(A)
11	Write(B)	
12		Read(B)
13		product=product*B
14		B=B-1

Επικοινωνία με τη βάση:

Time 1: Η T1 διαβάζει A=3

Time 4: Η T2 διαβάζει A=3

Time 5: Η T1 γράφει A=1 (**update**)

Time 8: Η T1 διαβάζει B=5

Συναλλαγή T1:

Time 2: A=A-2=1

Time 9: B=B+1=6

Συναλλαγή T2:

Time 3: product = 1

Time 6: product = 3

Time 7: A=A-1=2

Κεφάλαιο 9 (Βάσεις Δεδομένων)

	Βάση	T1	T2
A	2	1	2
B	5	6	

Υποθέστε ότι δύο συναλλαγές T1 και T2 εκτελούνται παράλληλα όπως στον παρακάτω πίνακα. Αν αρχικά A=3 και B=5, ποιες θα είναι οι τελικές τιμές των A, B στη βάση δεδομένων και ποια τιμή θα έχει στο τέλος το product;

Time	T1	T2
1	Read(A)	
2	A=A-2	
3		product=1
4		Read(A)
5	Write(A)	
6		product=product*A
7		A=A-1
8	Read(B)	
9	B=B+1	
10		Write(A)
11	Write(B)	
12		Read(B)
13		product=product*B
14		B=B-1

Επικοινωνία με τη βάση:

Time 1: Η T1 διαβάζει A=3

Time 4: Η T2 διαβάζει A=3

Time 5: Η T1 γράφει A=1 (**update**)

Time 8: Η T1 διαβάζει B=5

Time 10: Η T2 γράφει A=2 (**update**)

Συναλλαγή T1:

Time 2: A=A-2=1

Time 9: B=B+1=6

Συναλλαγή T2:

Time 3: product = 1

Time 6: product = 3

Time 7: A=A-1=2

Κεφάλαιο 9 (Βάσεις Δεδομένων)

	Βάση	T1	T2
A	2	1	2
B	6	6	

Υποθέστε ότι δύο συναλλαγές T1 και T2 εκτελούνται παράλληλα όπως στον παρακάτω πίνακα. Αν αρχικά A=3 και B=5, ποιες θα είναι οι τελικές τιμές των A, B στη βάση δεδομένων και ποια τιμή θα έχει στο τέλος το product;

Time	T1	T2
1	Read(A)	
2	A=A-2	
3		product=1
4		Read(A)
5	Write(A)	
6		product=product*A
7		A=A-1
8	Read(B)	
9	B=B+1	
10		Write(A)
11	Write(B)	
12		Read(B)
13		product=product*B
14		B=B-1

Επικοινωνία με τη βάση:

Time 1: Η T1 διαβάζει A=3

Time 4: Η T2 διαβάζει A=3

Time 5: Η T1 γράφει A=1 (**update**)

Time 8: Η T1 διαβάζει B=5

Time 10: Η T2 γράφει A=2 (**update**)

Time 11: Η T1 γράφει B=6 (**update**)

Συναλλαγή T1:

Time 2: A=A-2=1

Time 9: B=B+1=6

Συναλλαγή T2:

Time 3: product = 1

Time 6: product = 3

Time 7: A=A-1=2

Κεφάλαιο 9 (Βάσεις Δεδομένων)

	Βάση	T1	T2
A	2	1	2
B	6	6	5

Υποθέστε ότι δύο συναλλαγές T1 και T2 εκτελούνται παράλληλα όπως στον παρακάτω πίνακα. Αν αρχικά A=3 και B=5, ποιες θα είναι οι τελικές τιμές των A, B στη βάση δεδομένων και ποια τιμή θα έχει στο τέλος το product;

Time	T1	T2
1	Read(A)	
2	A=A-2	
3		product=1
4		Read(A)
5	Write(A)	
6		product=product*A
7		A=A-1
8	Read(B)	
9	B=B+1	
10		Write(A)
11	Write(B)	
12		Read(B)
13		product=product*B
14		B=B-1

Επικοινωνία με τη βάση:

Time 1: Η T1 διαβάζει A=3

Time 4: Η T2 διαβάζει A=3

Time 5: Η T1 γράφει A=1 (**update**)

Time 8: Η T1 διαβάζει B=5

Time 10: Η T2 γράφει A=2 (**update**)

Time 11: Η T1 γράφει B=6 (**update**)

Time 12: Η T2 διαβάζει B=6

Συναλλαγή T1:

Time 2: A=A-2=1

Time 9: B=B+1=6

Συναλλαγή T2:

Time 3: product = 1

Time 6: product = 3

Time 7: A=A-1=2

Time 13: product = 18 (τελική τιμή = 3 X 6)

Time 14: B=B-1=5

Κεφάλαιο 12 (Θεωρία Υπολογισμού)

- Δώστε τον κώδικα σε στοιχειώδη γλώσσα μηχανής που εκτελεί το διπλασιασμό μιας μεταβλητής X .

Οι μόνες εντολές της Γλώσσας είναι:

```
clear όνομα;  
incr όνομα;  
decr όνομα;  
while όνομα not 0 do;  
... end;
```

Κεφάλαιο 12 (Θεωρία Υπολογισμού)

- Δώστε τον κώδικα σε στοιχειώδη γλώσσα μηχανής που εκτελεί το διπλασιασμό μιας μεταβλητής X.

```
clear Z;  
while X not 0 do;  
    incr Z;  
    incr Z;  
    decr X;  
end;
```

Οι μόνες εντολές της Γλώσσας
είναι:

```
clear όνομα;  
incr όνομα;  
decr όνομα;  
while όνομα not 0 do;  
... end;
```

Κεφάλαιο 12 (Θεωρία Υπολογισμού)

- Τι κάνει το παρακάτω πρόγραμμα σε στοιχειώδη γλώσσα μηχανής;

```
clear Z;  
while X not 0 do;  
  clear W;  
  while X not 0 do;  
    incr Z;  
    incr W;  
    decr X;  
  end;  
  while W not 0 do;  
    incr X;  
    decr W;  
  end;  
  decr X;  
end;
```

Κεφάλαιο 12 (Θεωρία Υπολογισμού)

- Τι κάνει το παρακάτω πρόγραμμα σε στοιχειώδη γλώσσα μηχανής;

```
clear Z;
while X not 0 do;
  clear W;
  while X not 0 do;
    incr Z;       $Z \leftarrow Z + X$ 
    incr W;       $W \leftarrow X$ 
    decr X;
  end;
  while W not 0 do;
    incr X;       $X \leftarrow W$  (X regains the value it had at
    decr W;      the beginning of the outer while)
  end;
  decr X;      X decreases by 1 before the next outer while
end;
```

Σε κάθε κύκλο του outer while
προστίθεται στο Z η τιμή του X.
Την 1^η φορά η αρχική τιμή του X.
Την 2^η φορά η αρχική τιμή του $X - 1$
...
Την ν^η φορά το 1.

Κεφάλαιο 12 (Θεωρία Υπολογισμού)

- Τι κάνει το παρακάτω πρόγραμμα σε στοιχειώδη γλώσσα μηχανής;

```
clear Z;
while X not 0 do;
  clear W;
  while X not 0 do;
    incr Z;
    incr W;
    decr X;
  end;
  while W not 0 do;
    incr X;
    decr W;
  end;
  decr X;
end;
```

$Z \leftarrow Z + X$
 $W \leftarrow X$

Σε κάθε κύκλο του outer while
προστίθεται στο Z η τιμή του X.
Την 1^η φορά η αρχική τιμή του X.
Την 2^η φορά η αρχική τιμή του X - 1
...
Την ν^η φορά το 1.

$X \leftarrow W$ (X regains the value it had at
the beginning of the outer while)

X decreases by 1 before the next outer while

$$\text{Final result:}$$
$$Z = \sum_{i=1}^X i = \frac{X \cdot (X + 1)}{2}$$

Κεφάλαιο 12 (Θεωρία Υπολογισμού)

Έστω ότι το πρόβλημα Π_1 ανάγεται πολυωνυμικά στο πρόβλημα Π_2 ($\Pi_1 \leq_p \Pi_2$). Ποιες από τις παρακάτω προτάσεις είναι **πάντα** σωστές;

- Αν Π_1 ανήκει στην κλάση NP, τότε Π_1 ανήκει και στην κλάση P.
- Αν Π_1 ανήκει στην κλάση P, τότε Π_1 ανήκει και στην κλάση NP.
- Αν Π_1 ανήκει στην κλάση NP, τότε Π_1 είναι NP-complete.
- Αν Π_1 είναι NP-complete, τότε Π_1 ανήκει στην κλάση NP.

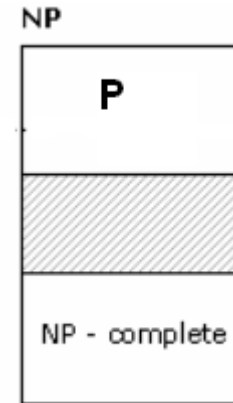
- Αν Π_1 ανήκει στην κλάση P, τότε και Π_2 ανήκει στην κλάση P.
- Αν Π_2 ανήκει στην κλάση P, τότε και Π_1 ανήκει στην κλάση P.
- Αν Π_1 είναι NP-complete, τότε και Π_2 είναι NP-complete.
- Αν Π_2 είναι NP-complete, τότε και Π_1 είναι NP-complete.

- Αν για το πρόβλημα Π_3 ισχύει $\Pi_3 \leq_p \Pi_1$, τότε $\Pi_3 \leq_p \Pi_2$.
- Αν Π_2 είναι NP-complete και Π_3 ανήκει στην κλάση NP, τότε $\Pi_2 \leq_p \Pi_3$.
- Αν Π_2 είναι NP-complete και Π_3 ανήκει στην κλάση NP, τότε $\Pi_3 \leq_p \Pi_2$.
- Αν Π_2 είναι NP-complete και Π_3 ανήκει στην κλάση P, τότε $\Pi_3 \leq_p \Pi_2$.

Κεφάλαιο 12 (Θεωρία Υπολογισμού)

Έστω ότι το πρόβλημα Π_1 ανάγεται πολυωνυμικά στο πρόβλημα Π_2 ($\Pi_1 \leq_P \Pi_2$). Ποιες από τις παρακάτω προτάσεις είναι **πάντα** σωστές;

- Αν Π_1 ανήκει στην κλάση NP, τότε Π_1 ανήκει και στην κλάση P.
- Αν Π_1 ανήκει στην κλάση P, τότε Π_1 ανήκει και στην κλάση NP.
- Αν Π_1 ανήκει στην κλάση NP, τότε Π_1 είναι NP-complete.
- Αν Π_1 είναι NP-complete, τότε Π_1 ανήκει στην κλάση NP.
- Αν Π_1 ανήκει στην κλάση P, τότε και Π_2 ανήκει στην κλάση P.
- Αν Π_2 ανήκει στην κλάση P, τότε και Π_1 ανήκει στην κλάση P.
- Αν Π_1 είναι NP-complete, τότε και Π_2 είναι NP-complete.
- Αν Π_2 είναι NP-complete, τότε και Π_1 είναι NP-complete.
- Αν για το πρόβλημα Π_3 ισχύει $\Pi_3 \leq_P \Pi_1$, τότε $\Pi_3 \leq_P \Pi_2$.
- Αν Π_2 είναι NP-complete και Π_3 ανήκει στην κλάση NP, τότε $\Pi_2 \leq_P \Pi_3$.
- Αν Π_2 είναι NP-complete και Π_3 ανήκει στην κλάση NP, τότε $\Pi_3 \leq_P \Pi_2$.
- Αν Π_2 είναι NP-complete και Π_3 ανήκει στην κλάση P, τότε $\Pi_3 \leq_P \Pi_2$.
- **Κλάση προβλημάτων P:** σύνολο των πολυωνυμικών προβλημάτων. Αν ένα πρόβλημα ανήκει στην κλάση P σημαίνει ότι η λύση του μπορεί να υπολογιστεί σε **σχετικά σύντομο χρόνο** (πρακτική για διάφορες εφαρμογές).
- Κάθε πρόβλημα που μπορώ να **ελέγξω την ορθότητα** μιας προτεινόμενης λύσης σε **πολυωνυμικό** χρόνο ανήκει στην **κλάση NP**.
 1. Προσοχή: δεν μπορώ να ελέγξω τη μη ορθότητα, **μόνο** την ορθότητα!
 2. Η κλάση P είναι υποσύνολο της κλάσης NP
- Ένα πρόβλημα Π είναι NP-Complete αν και μόνο αν
 1. Π ανήκει στην κλάση NP
 2. Κάθε πρόβλημα Π' της NP **ανάγεται πολυωνυμικά** στο Π (Δηλαδή μπορώ σε πολυωνυμικό αριθμό βημάτων να χρησιμοποιήσω τη λύση του Π για να λύσω το Π')



Κεφάλαιο 12 (Θεωρία Υπολογισμού)

Έστω ότι το πρόβλημα Π_1 ανάγεται πολυωνυμικά στο πρόβλημα Π_2 ($\Pi_1 \leq_p \Pi_2$). Ποιες από τις παρακάτω προτάσεις είναι **πάντα** σωστές;

- Αν Π_1 ανήκει στην κλάση NP, τότε Π_1 ανήκει και στην κλάση P. Λάθος
- Αν Π_1 ανήκει στην κλάση P, τότε Π_1 ανήκει και στην κλάση NP. Σωστό
- Αν Π_1 ανήκει στην κλάση NP, τότε Π_1 είναι NP-complete. Λάθος
- Αν Π_1 είναι NP-complete, τότε Π_1 ανήκει στην κλάση NP. Σωστό

- Αν Π_1 ανήκει στην κλάση P, τότε και Π_2 ανήκει στην κλάση P. Λάθος
- Αν Π_2 ανήκει στην κλάση P, τότε και Π_1 ανήκει στην κλάση P. Σωστό
- Αν Π_1 είναι NP-complete, τότε και Π_2 είναι NP-complete. Σωστό
- Αν Π_2 είναι NP-complete, τότε και Π_1 είναι NP-complete. Λάθος

- Αν για το πρόβλημα Π_3 ισχύει $\Pi_3 \leq_p \Pi_1$, τότε $\Pi_3 \leq_p \Pi_2$. Σωστό
- Αν Π_2 είναι NP-complete και Π_3 ανήκει στην κλάση NP, τότε $\Pi_2 \leq_p \Pi_3$. Λάθος
- Αν Π_2 είναι NP-complete και Π_3 ανήκει στην κλάση NP, τότε $\Pi_3 \leq_p \Pi_2$. Σωστό
- Αν Π_2 είναι NP-complete και Π_3 ανήκει στην κλάση P, τότε $\Pi_3 \leq_p \Pi_2$. Σωστό

ΚΑΛΗ ΕΞΕΤΑΣΤΙΚΗ!!!