**Θ ΕΜΑ 1ο :**

Α) Χωρίς αδιέξοδο: Όλες οι διεργασίες καλούν με την ίδια σειρά τους R1-R5. Αν εκτελεστούν με τη σειρά, αν η A καλέσει τον R1 και διακοπεί, οι άλλες 3 διεργασίες καλούν την R1 και πάνε για ύπνο, η Α όμως δεν κοιμάται.

Β) Με αδιέξοδο: Η Α δεσμεύει διαδοχικά τους R1,R5 και μετά με μία σειρά τους άλλους 3 πόρους. Η Β δεσμεύει πρώτα τον R2 και μετά με κάποια σειρά τους άλλους 4. Η C δεσμεύει πρώτα τον R3 και μετά με κάποια σειρά τους άλλους 4. Η D δεσμεύει πρώτα τον R4 και μετά με κάποια σειρά τους άλλους 4. Αν η Α κοπεί μετά τη δέσμευση του R5 και οι άλλες μετά τη δέσμευση των R2, R3, R4 αντίστοιχα, όλο το σύστημα οδηγείται σε αδιέξοδο.

**ΠΑΡΑΤΗΡΗΣΗ ΓΙΑ ΤΙΣ ΛΥΣΕΙΣ:** Πλήρης θεωρείται η λύση που παρουσιάζει σενάρια όπου όλες οι διεργασίες προσπαθούν να δεσμεύσουν όλους τους υπάρχοντες πόρους.

**ΘΕΜΑ 2ο:**

Α) Μ=128/4 = 32 πλαίσια, Ν= 4M/4K=1024 σελίδες

Β) Offset=12 bit, 4M=222, άρα το PN για τις ιδεατές σελίδες είναι 22-12=10 bit . Επίσης, για τα φυσικά πλαίσια, 128Κ=217 άρα το PN είναι 17-12=5 bit.

Γ) Η ιδεατή σελίδα 40 δεν υπάρχει ούτε στο TLB ούτε στο PMT, γράφεται στο φυσικό πλαίσιο 0. H εγγραφή θα είναι 40|0. H εγγραφή γίνεται ΚΑΙ στο PMT ΚΑΙ στο ΤLB. Ο χρόνος που απαιτείται είναι 20 μονάδες από τον δίσκο προς τη μνήμη + 10 από τη μνήμη προς τη CPU άρα 30 μονάδες. Οι σελίδες 6,7 υπάρχουν στο TLB άρα απαιτείται απλή ανάγνωση από τη μνήμη, 10 μονάδες. Η σελίδα 21 υπάρχει στο PMT στη θέση 21, αλλά όχι στο TLB. Άρα, γίνεται διπλή ανάγνωση από τη μνήμη, 20 μονάδες και η εγγραφή 21|21 θα περάσει στη θέση 1 του TLB. Τέλος, η σελίδα 50 δεν υπάρχει ούτε στο TLB ούτε στο PMT. Αυτή θα γραφτεί στο πλαίσιο 1, άρα η εγγραφή 50|1 θα πάει και στο PMT και στο TLB. O χρόνος ανάγνωσης των δεδομένων είναι 20 μονάδες από το δίσκο στη μνήμη και 10 από τη μνήμη στη CPU, σύνολο 30.

Άρα, ο συνολικός χρόνος ανάγνωσης είναι 30+10+10+20+30 =100 χρονικές μονάδες.

Δ) Η τελική εικόνα του TLB είναι:

40|0 (αντί της 0|0)

21|21 (αντί της 1|1)

50|1 (αντί της 2|2)

3|3

4|4

5|5

6|6

7|7

Ε) Η διεύθυνση 40Κ βρίσκεται στην ιδεατή σελίδα 10, δεν έχει αντικατασταθεί και βρίσκεται στο φυσικό πλαίσιο 10. Άρα απλώς γράφουμε 01010|000000000000

**ΘΕΜΑ 3ο**

Α) Καταρχήν, κάθε παραγωγός ξυπνάει τον εαυτό του. Άρα, όταν, έστω ο P1 μειώσει την empty και την κάνει 0, θα κοιμηθεί και αυτός και ο P2. Άρα, πρέπει ο P1 να ξυπνά τον P2 Και ο P2 τον καταναλωτή. Ακόμη και με αυτή τη διόρθωση όμως, μπορεί να συμβεί το εξής. Τρέχει ο P1 και κόβεται όταν μειώσει το empty, ενώ και ο P2 είναι ξυπνητός. Αν ο P2 ξεκινήσει, θα κατεβάσει το empty άλλη μία μονάδα, άρα θα υπάρχει αναντιστοιχία ανάμεσα στις κενές θέσεις και τα δεδομένα που έχουν γραφτεί, δηλαδή το empty θα έχει μειωθεί κατά 2 ενώ έχει γίνει μία εγγραφή. Η λύση σε αυτό το πρόβλημα είναι να κλειδωθεί η μνήμη πριν το down (empty), δηλαδή το down(mutex) να ανεβεί πάνω από το down(empty), ώστε ο ένας παραγωγός να μπορεί να γράψει μόνον αφότου ολοκληρωθεί η εγγραφή του άλλου. Αντίστοιχα, το up(mutex) θα γίνει μετά το up(full).

B) O κώδικας του καταναλωτή είναι όπως έχει παρουσιαστεί στο μάθημα, απλώς στο τέλος του πρέπει να ξυπνάει τον P1.

**ΘΕΜΑ 4ο**

Η Α θα καταλάβει το τμήμα από 0-70Κ. Τα υπόλοιπα 58 θα διασπαστούν σε 14 κομμάτια των τεσσάρων και ένα των 2. Η Β θα καταλάβει από 70Κ μέχρι 106Κ (1Κ κατακερματισμός).

Ερχόμενη η C, θα καταλάβει το κομμάτι από 128Κ-256Κ (αφήνοντας υπόλοιπο 48Κ). Όταν τελειώσει η Α, θα ελευθερώσει 70Κ, τα οποία όμως δεν μπορούν να ενωθούν με άλλο κομμάτι. Η D θα καταλάβει το τμήμα από 256-320 αφήνοντας κατακερματισμένο ένα κομμάτι 4Κ.