

# ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

## Διεργασίες – Επικοινωνία - Χρονοπρογραμματισμός Ερωτήσεις Επανάληψης

Υλικό από:

Modern Operating Systems, A.S. Tanenbaum

Σύνθεση

Κ.Γ. Μαργαρίτης, Τμήμα Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας

## ΠΡΟΒΛΗΜΑΤΑ

1. Στην Εικόνα 2-2 παρουσιάζονται τρεις καταστάσεις διεργασιών. Θεωρητικά, όταν υπάρχουν τρεις καταστάσεις είναι δυνατές έξι μεταβάσεις, δύο από κάθε κατάσταση. Ωστόσο, παρουσιάζονται μόνο τέσσερις μεταβάσεις. Υπάρχουν κάποιες προϋποθέσεις ώστε να είναι δυνατή η εμφάνιση της μίας ή και των δύο μεταβάσεων που λείπουν;
2. Υποθέστε ότι σχεδιάζετε μια πρωτοποριακή αρχιτεκτονική για υπολογιστές, η οποία επιτελεί εναλλαγή διεργασιών μέσω υλικού, αντί να χρησιμοποιεί διακοπές. Ποιες πληροφορίες χρειάζεται η CPU; Περιγράψτε πώς λειτουργεί η εναλλαγή διεργασιών μέσω υλικού.
3. Σε όλους τους σύγχρονους υπολογιστές, ένα τμήμα (τουλάχιστον) των χειριστών διακοπών είναι γραμμένο σε συμβολική γλώσσα. Γιατί;
4. Όταν μια διακοπή ή μια κλήση συστήματος μεταφέρει τον έλεγχο στο λειτουργικό σύστημα, χρησιμοποιείται γενικά μια περιοχή στοίβας στον πυρήνα, αντί για τη στοίβα της διεργασίας που διακόπηκε. Γιατί;
5. Στο κείμενο αναφέρθηκε ότι το μοντέλο της Εικόνας 2-6(α) δεν είναι κατάλληλο για την υλοποίηση ενός διακομιστή αρχείων ο οποίος χρησιμοποιεί κρυφή μνήμη. Γιατί δεν είναι κατάλληλο; Δε θα μπορούσε κάθε διεργασία να διαθέτει τη δική της κρυφή μνήμη;
6. Στην Εικόνα 2-7 το σύνολο των καταχωρητών αναφέρεται ως οντότητα που ανήκει στα νήματα και όχι στις διεργασίες. Γιατί; Σε τελική ανάλυση, η μηχανή διαθέτει μόνον ένα σύνολο από καταχωρητές.
7. Όταν μια πολυνηματική διεργασία εκτελεί την κλήση fork, αν η θυγατρική διεργασία λάβει αντίγραφο από όλα τα νήματα της μητρικής, προκύπτει το εξής πρόβλημα: ας υποθέσουμε ότι ένα από τα αρχικά νήματα ανέμενε είσοδο από το πληκτρολόγιο. Μετά την εκτέλεση της fork, υπάρχουν πλέον δύο νήματα που αναμένουν είσοδο από το πληκτρολόγιο, ένα σε κάθε διεργασία. Εμφανίζεται το πρόβλημα αυτό σε διεργασίες που διαθέτουν ένα μόνο νήμα;
8. Στην Εικόνα 2-10, παρουσιάζεται ένας πολυνηματικός διακομιστής ιστοσελίδων. Αν ο μοναδικός τρόπος για να γίνει ανάγνωση από αρχείο είναι η συνηθισμένη κλήση συστήματος read (η οποία οδηγεί σε μπλοκάρισμα), πιστεύετε ότι χρησιμοποιούνται στο διακομιστή νήματα επιπέδου χρήστη ή νήματα επιπέδου πυρήνα; Γιατί;

9. Γιατί πρέπει ένα νήμα να επιστρέφει εθελοντικά τον έλεγχο της CPU, καλώντας τη `thread_yield`. Σε τελική ανάλυση, από τη στιγμή που δεν υπάρχουν περιοδικές διακοπές ρολογιού, υπάρχει πιθανότητα να μην ξαναπάρει ποτέ τον έλεγχο της CPU.
10. Είναι δυνατόν να προεκτοπιστεί ένα νήμα από διακοπή ρολογιού; Αν ναι, κάτω από ποιες προϋποθέσεις; Αν όχι, γιατί;
11. Σε αυτό το πρόβλημα καλείστε να συγκρίνετε την ανάγνωση ενός αρχείου με τη χρήση μονονηματικού διακομιστή αρχείων, σε σχέση με την αντίστοιχη ενέργεια όταν χρησιμοποιείται πολυνηματικός διακομιστής. Χρειάζονται 15 nsec για να γίνει η αίτηση εργασίας, να περάσει από το διακομιστή, και να γίνει η υπόλοιπη επεξεργασία, υποθέτοντας ότι τα δεδομένα που απαιτούνται βρίσκονται στην κρυφή μνήμη. Αν χρειάζεται ενέργεια που εμπλέκει το δίσκο (πρόγραμμα που συμβαίνει στο ένα τρίτο των περιπτώσεων), απαιτούνται επιπλέον 75 nsec, στη διάρκεια των οποίων το νήμα αναστέλλεται. Πόσες τέτοιες αιτήσεις μπορεί να διαχειριστεί ο διακομιστής αν είναι μονονηματικός και πόσες αν είναι πολυνηματικός;
12. Στο κείμενο περιγράψαμε έναν πολυνηματικό διακομιστή ιστοσελίδων, αποδεικνύοντας γιατί είναι προτιμότερος από έναν αντίστοιχο μονονηματικό και από ένα διακομιστή που χρησιμοποιεί μηχρησιμεπερισμένων καταστάσεων. Υπάρχουν περιπτώσεις όπου πρέπει να προτιμηθεί μονονηματικός διακομιστής; Δώστε ένα παράδειγμα.
13. Στη συζήτηση που αφορούσε τις καθολικές μεταβλητές στα νήματα, χρησιμοποιήσαμε τη διαδικασία `create_global` για να αποκτήσουμε μνήμη για τους δείκτες προς αυτές τις μεταβλητές, αντί να δημιουργήσουμε απευθείας αυτές τις μεταβλητές. Είναι αυτό απαραίτητο, ή οι διαδικασίες έχουν τη δυνατότητα να εργάζονται χρησιμοποιώντας τις ίδιες τις τιμές των μεταβλητών;
14. Θεωρήστε ένα σύστημα στο οποίο τα νήματα υλοποιούνται εξαλοκλήρου στο χώρο του χρήστη, ενώ το σύστημα χρόνου εκτέλεσης προκαλεί μια διακοπή ρολογιού ανά δευτερόλεπτο. Ας υποθέσουμε ότι συμβαίνει μια διακοπή ρολογιού όταν κάποιο νήμα εκτελείται στο σύστημα χρόνου εκτέλεσης. Τι πρόβλημα θα δημιουργηθεί; Μπορείτε να προτείνετε κάποια λύση;
15. Ας υποθέσουμε ότι κάποιο λειτουργικό σύστημα δε διαθέτει κάποια λειτουργία σαν την κλήση συστήματος `select` για να εξακριβώνει εκ των προτέρων αν είναι ασφαλής η ανάγνωση από αρχείο, διαχέτευση, ή συσκευή, αλλά επιτρέπει να χρησιμοποιούνται χρονόμετρα συναγερμού, τα οποία διακόπτουν τις μπλοκαρισμένες κλήσεις συστήματος. Είναι δυνατόν να δημιουργηθεί κάποιο πακέτο νημάτων στο χώρο του χρήστη κάτω από αυτές τις συνθήκες; Ας γίνει συζήτηση πάνω στο ζήτημα.
16. Είναι δυνατόν το πρόβλημα αντιστροφής προτεραιοτήτων που εξετάσαμε στην ενότητα 2.3.4 να συμβεί όταν χρησιμοποιούνται νήματα επιπέδου χρήστη; Ναι ή όχι και γιατί;
17. Σε κάποιο σύστημα που χρησιμοποιεί νήματα, υπάρχει μία στοίβα ανά νήμα, ή μία στοίβα ανά διεργασία, όταν χρησιμοποιούνται νήματα επιπέδου χρήστη; Τι συμβαίνει όταν χρησιμοποιούνται νήματα επιπέδου πυρήνα; Εξηγήστε την απάντησή σας.
18. Τι είναι οι συνθήκες συναγωνισμού;
19. Στη διάρκεια της ανάπτυξης ενός υπολογιστικού συστήματος, συνήθως γίνεται αρχικά προσομοίωση του συστήματος με κάποιο πρόγραμμα που εκτελεί μία εντολή τη φορά. Ακόμη και οι πολυεπεξεργαστές προσομοιώνονται με αυτόν τον τρόπο. Είναι πιθανό να εμφανιστούν συνθήκες συναγωνισμού όταν δεν υπάρχουν συμβάντα που λαμβάνουν χώρα ταυτόχρονα, όπως σε αυτή την περίπτωση;

20. Λειτουργεί κανονικά η λύση που χρησιμοποιεί αναμονή με απασχόληση (με χρήση της μεταβλητής *turn*) στην Εικόνα 2-20, όταν οι δύο διεργασίες εκτελούνται σε πολυεπεξεργαστή με κοινή μνήμη, δηλαδή ένα σύστημα με δύο CPU που έχουν κοινή μνήμη;
21. Λειτουργεί κανονικά η λύση του Peterson στο πρόβλημα αμοιβαίου αποκλεισμού που παρουσιάζεται στην Εικόνα 2-21, όταν ο χρονοπρογραμματισμός των διεργασιών είναι προεκτοπιστικός; Τι συμβαίνει όταν είναι μη προεκτοπιστικός;
22. Θεωρήστε έναν υπολογιστή που δε διαθέτει την εντολή TSL, αλλά διαθέτει μια εντολή που ανταλλάσσει τα περιεχόμενα ενός καταχωρητή και μιας λέξης στη μνήμη, ενέργεια που γίνεται αδιαίρετα. Μπορεί η εντολή αυτή να χρησιμοποιηθεί για να γραφεί μια ρουτίνα με το όνομα *enter\_region* όπως αυτή που φαίνεται στην Εικόνα 2-22;
23. Κατασκευάστε ένα σχεδιάγραμμα του τρόπου με τον οποίο ένα λειτουργικό σύστημα, που μπορεί να απενεργοποιεί τις διακοπές, θα μπορούσε να υλοποιήσει τους σηματοφόρους.
24. Δείξτε με ποιο τρόπο οι σηματοφόροι που είναι μετρητές (δηλαδή οι σηματοφόροι που μπορούν να πάρουν οποιαδήποτε τιμή) μπορούν να υλοποιηθούν μόνο με διαδικούς σηματοφόρους και συνηθισμένες εντολές μηχανής.
25. Αν ένα σύστημα διαθέτει δύο μόνο διεργασίες, έχει νόημα να χρησιμοποιεί φράγμα για να τις συγχρονίσει; Ναι ή όχι και γιατί;
26. Στην ενότητα 2.3.4 έχει περιγραφεί μια κατάσταση που περιλαμβάνει μια διεργασία υψηλής προτεραιότητας Y και μια διεργασία χαμηλής προτεραιότητας X, ενώ η X εκτελεί άπειρες επαναλήψεις χωρίς αποτέλεσμα. Εμφανίζεται το ίδιο πρόβλημα αν, αντί του χρονοπρογραμματισμού προτεραιότητας, χρησιμοποιηθεί χρονοπρογραμματισμός εκ περιτροπής; Συζητήστε το θέμα αυτό.
27. Μπορούν δύο νήματα που ανήκουν στην ίδια διεργασία να συγχρονιστούν χρησιμοποιώντας ένα σηματοφόρο του πυρήνα, αν τα νήματα έχουν υλοποιηθεί στον πυρήνα; Τι συμβαίνει αν έχουν υλοποιηθεί στο χώρο του χρήστη; Υποθέστε ότι δεν υπάρχουν νήματα από άλλες διεργασίες τα οποία να έχουν πρόσβαση στο σηματοφόρο. Συζητήστε τις απαντήσεις σας.
28. Ο συγχρονισμός στους ελεγκτές χρησιμοποιεί μεταβλητές συνθήκης, καθώς και δύο ειδικές λειτουργίες *wait* και *signal*. Μια πιο γενική μορφή συγχρονισμού θα περιλάμβανε ένα μόνο γενικό αρχέτυπο (primitive), το *waituntil*, το οποίο θα είχε ως παράμετρο μια αυθαίρετη λογική (Boolean) συνθήκη. Θα μπορούσε, για παράδειγμα, να ήταν η εξής:

`waituntil  $x < 0$  or  $y + z < n$`

Το αρχέτυπο *signal* δε θα χρειαζόταν πια. Η σχεδίαση αυτή, παρόλο που είναι πιο γενική από τις αντίστοιχες των Hoare και Brinch Hansen, δε χρησιμοποιείται στην πράξη. Γιατί συμβαίνει αυτό; Υπόδειξη: σκεφθείτε την υλοποίηση.

29. Οι υπάλληλοι σε ένα εστιατόριο γρήγορου φαγητού (fast food) χωρίζονται σε τέσσερις κατηγορίες: (1) αποδέκτες παραγγελιών, οι οποίοι παίρνουν τις παραγγελίες των πελατών, (2) μάγειρες, οι οποίοι ετοιμάζουν το φαγητό, (3) συσκευαστές, οι οποίοι τοποθετούν το φαγητό σε πακέτα, και (4) ταμίες, που δίνουν τα πακέτα στους πελάτες και παίρνουν τα χρήματα. Κάθε υπάλληλος μπορεί να θεωρηθεί ως επικοινωνούσα σειριακή διεργασία. Ποια μορφή διαδιεργασιακής επικοινωνίας χρησιμοποιούν; Συζητήστε το μοντέλο αυτό με τις διεργασίες στο UNIX.
30. Θεωρήστε ένα σύστημα μεταβίβασης μηνυμάτων που χρησιμοποιεί γραμματοκιβώτια. Όταν στέλνεται μήνυμα σε γεμάτο γραμματοκιβώτιο ή γίνεται προσπάθεια να ληφθεί μήνυμα από άδειο γραμματοκιβώτιο, οι διεργασίες δεν μπλοκάρονται. Αντί γι' αυτό, τους επιστρέφεται ένας κωδικός

σφάλματος. Η διεργασία αποκρίνεται στον κωδικό λάθους επιχειρώντας απλώς τη ίδια ενέργεια ξανά και ξανά, μέχρι να επιτύχει. Οδηγεί η σχεδίαση αυτή σε συνθήκες συναγωνισμού;

31. Στη λύση του προβλήματος του δείχνου των φιλοσόφων (Εικόνα 2-20), εξηγήστε γιατί η μεταβλητή κατάστασης παίρνει αρχική τιμή *HUNGRY* στη διαδικασία *take\_forks*.
32. Κοιτάξτε τη διαδικασία *put\_forks* στην Εικόνα 2-20. Υποθέστε ότι η μεταβλητή *state[i]* παίρνει την τιμή *THINKING* μετά τις δύο κλήσεις στην *test* και όχι πριν από αυτές. Πώς επηρεάζει αυτή η αλλαγή τη λύση του προβλήματος;
33. Το πρόβλημα των αναγνώστων-συγγραφέων μπορεί να διατυπωθεί με πολλούς τρόπους, ανάλογα με το ποια κατηγορία διεργασιών εκκινείται και πότε. Περιγράψτε προσεκτικά τρεις διαφορετικές παραλλαγές του προβλήματος, όπου η κάθε μία να ευνοεί ή να αδικεί μια κατηγορία διεργασιών. Για κάθε παραλλαγή καθορίστε τι συμβαίνει, όταν ένας αναγνώστης ή ένας συγγραφέας γίνεται έτοιμος να προσπελάσει τη βάση δεδομένων, καθώς και όταν μια διεργασία τερματίζεται ενώ χρησιμοποιεί τη βάση δεδομένων.
34. Ο υπολογιστής CDC 6600 μπορεί να χειριστεί ταυτόχρονα μέχρι 10 διεργασίες εισόδου/εξόδου το πολύ, χρησιμοποιώντας μια ενδιαφέρουσα εκδοχή του χρονοπρογραμματισμού με εξυπηρέτηση εκ περιτροπής, η οποία ονομάζεται **μερισμός επεξεργαστή** (processor sharing). Οι εναλλαγές διεργασιών παρουσιάζονται μετά από κάθε εντολή, συνεπώς η εντολή 1 προέρχεται από τη διεργασία 1, η εντολή 2 από τη διεργασία 2, κ.ο.κ. Η εναλλαγή διεργασιών γίνεται από εξειδικευμένο υλικό και η επιβάρυνση είναι μηδενική. Αν μια διεργασία χρειάζεται  $T$  sec για να ολοκληρωθεί (υποτίθεται ότι δεν υπάρχει συνιγωνισμός), πόσο χρόνο θα χρειαστεί αν χρησιμοποιηθεί μερισμός επεξεργαστή και υπάρχουν συνολικά  $N$  διεργασίες;
35. Οι χρονοπρογραμματιστές εκ περιτροπής διατηρούν συνήθως μια λίστα με όλες τις εκτελέσιμες διεργασίες, στην οποία κάθε διεργασία εμφανίζεται ακριβώς μία φορά. Τι θα συνέβαινε, αν μια διεργασία παρουσιαζόταν δύο φορές στη λίστα; Μπορείτε να σκεφθείτε κάποιο λόγο, για τον οποίο θα επιτρεπόταν κάτι τέτοιο;
36. Μπορεί να μετρηθεί αν μια διεργασία είναι υπολογιστικά φραγμένη ή φραγμένη από E/E, αν αναλυθεί ο πηγαίος κώδικάς της; Πώς μπορεί να προσδιοριστεί το θέμα αυτό κατά το χρόνο της εκτέλεσης;
37. Στην ενότητα "Πότε γίνεται ο χρονοπρογραμματισμός" αναφέρθηκε ότι ο χρονοπρογραμματισμός μπορεί να βελτιωθεί αν κάποια σημαντική διεργασία έχει τη δυνατότητα να παίζει κάποιο ρόλο στην επιλογή της επόμενης διεργασίας που θα εκτελεστεί, μόλις η ίδια μπλοκαριστεί. Περιγράψτε μια κατάσταση στην οποία η ιδέα αυτή μπορεί να εφαρμοστεί και εξηγήστε τον τρόπο εφαρμογής της.
38. Μετρήσεις σε συγκεκριμένο σύστημα έχουν δείξει ότι ο μέσος όρος του χρόνου εκτέλεσης για μια διεργασία είναι  $T$ , πριν αυτή μπλοκαριστεί από είσοδο/έξοδο. Η εναλλαγή διεργασίας απαιτεί χρόνο  $S$ , ο οποίος λογαριάζεται ως επιβάρυνση και συνεπώς θεωρείται χαμένος χρόνος. Για χρονοπρογραμματισμό εκ περιτροπής με κβάντο χρόνου  $Q$ , δώστε το μαθηματικό τύπο για την αποδοτικότητα της CPU, για κάθε μία από τις ακόλουθες περιπτώσεις.
  - (α)  $Q = \infty$
  - (β)  $Q > T$
  - (γ)  $S < Q < T$
  - (δ)  $Q = S$
  - (ε)  $Q$  σχεδόν 0

39. Πέντε εργασίες περιμένουν να εκτελεστούν. Οι αναμενόμενοι χρόνοι εκτέλεσής τους είναι 9, 6, 3, 5, και  $X$ . Με ποια σειρά πρέπει να εκτελεστούν ώστε να ελαχιστοποιηθεί ο μέσος χρόνος απόκρισης; (Η απάντησή σας πρέπει να είναι συνάρτηση του  $X$ .)
40. Πέντε εργασίες δέσμης, από  $A$  μέχρι  $E$ , καταφθάνουν σε υπολογιστικό κέντρο την ίδια περίπου χρονική στιγμή. Οι χρόνοι εκτέλεσής τους εκτιμώνται σε 10, 6, 2, 4, και 8 λεπτά αντίστοιχα. Οι προτεραιότητές τους (που καθορίστηκαν εξωτερικά) είναι 3, 5, 2, 1, και 4 αντίστοιχα, με το 5 να αποτελεί την υψηλότερη προτεραιότητα. Για τον καθέναν από τους επόμενους αλγορίθμους χρονοπρογραμματισμού, υπολογίστε το μέσο χρόνο δικτυακής διακίνησης των διεργασιών. Αγνοήστε την επιβάρυνση που συμβαίνει κατά τις εναλλαγές των διεργασιών.
- (α) Εξυπηρέτηση εκ περιτροπής.  
 (β) Χρονοπρογραμματισμός προτεραιοτήτων.  
 (γ) Εξυπηρέτηση με βάση τη σειρά άφιξης (εκτέλεση με τη σειρά 10, 6, 2, 4, 8).  
 (δ) Εξυπηρέτηση με βάση τη μικρότερη διάρκεια.
- Για την περίπτωση (α), υποθέστε ότι το σύστημα είναι πολυπρογραμματιζόμενο και ότι κάθε εργασία παίρνει δίκαιο μερίδιο του χρόνου της CPU. Για τις περιπτώσεις από (β) μέχρι (δ), υποθέστε ότι εκτελείται μόνο μία εργασία κάθε φορά, και εκτελείται μέχρι να ολοκληρωθεί. Όλες οι εργασίες είναι υπολογιστικά φραγμένες στο μέγιστο βαθμό.
41. Μια διεργασία που εκτελείται στο σύστημα CTSS χρειάζεται 30 κβάντα χρόνου για να ολοκληρωθεί. Πόσες φορές θα εισαχθεί στη μνήμη (συμπεριλαμβάνεται και η πρώτη φορά, όταν ξεκινάει να εκτελείται);
42. Μπορείτε να σκεφθείτε έναν τρόπο για να προστατεύσετε το σύστημα CTSS, από το πονηρό κόλπο που εφαρμόζουν οι χρήστες πατώντας πολλές φορές το πλήκτρο *Enter*;
43. Ο αλγόριθμος γήρανσης με  $a = 1/2$ , χρησιμοποιείται για την πρόβλεψη των χρόνων εκτέλεσης. Οι τέσσερις προηγούμενες εκτελέσεις, από την παλαιότερη προς την πιο πρόσφατη, έδωσαν χρόνους 40, 20, 40, και 15 msec αντίστοιχα. Ποια είναι η πρόβλεψη χρόνου για την επόμενη εκτέλεση;
44. Σε ήπιο σύστημα πραγματικού χρόνου υπάρχουν τέσσερα συμβάντα, με περιόδους 50, 100, 200, και 250 msec αντίστοιχα. Υποθέστε ότι τα τέσσερα συμβάντα χρειάζονται 35, 20, 10, και  $X$  msec χρόνου CPU αντίστοιχα. Ποια είναι η μεγαλύτερη τιμή του  $X$ , για την οποία το σύστημα είναι χρονοπρογραμματιζόμενο;
45. Εξηγήστε γιατί ο χρονοπρογραμματισμός δύο επιπέδων χρησιμοποιείται ευρύτατα.
46. Θεωρήστε ένα σύστημα στο οποίο είναι επιθυμητός ο διαχωρισμός της πολιτικής από το μηχανισμό, κατά το χρονοπρογραμματισμό των νημάτων πυρήνα. Προτείνετε μια μέθοδο για να επιτευχθεί ο στόχος αυτός.
47. Γράψτε ένα σενάριο κελύφους (shell script) το οποίο να δημιουργεί ένα αρχείο συνεχόμενων αριθμών με τον εξής τρόπο: διαβάζει τον τελευταίο αριθμό από το αρχείο, προσθέτει 1 στον αριθμό αυτόν, και τοποθετεί το άθροισμα στο τέλος του αρχείου. Εκτελέστε μία εκδοχή του σεναρίου στο παρασκήνιο και μία στο προσκήνιο, καθορίζοντας ότι οι δύο διεργασίες προσπελάζουν το ίδιο αρχείο. Πόσος χρόνος περνάει πριν την εμφάνιση συνθήκης συναγωνισμού ανάμεσα στις δύο διεργασίες; Ποια είναι η κρίσιμη περιοχή; Τροποποιήστε το σενάριο ώστε να αποφευχθεί η συνθήκη συναγωνισμού.
48. Υποθέστε ότι εργάζεστε σε ένα λειτουργικό σύστημα που υποστηρίζει σηματοφόρους. Υλοποιήστε ένα σύστημα μηνυμάτων. Γράψτε τις διαδικασίες αποστολής και παραλαβής μηνυμάτων.



49. Λύστε το πρόβλημα του δείχνου των φιλοσόφων χρησιμοποιώντας ελεγκτές αντί για σηματοφόρους.

50. Υποθέστε ότι ένα πανεπιστήμιο θέλει να δείξει πόσο "πολιτικά ορθό" είναι, εφαρμόζοντας το δόγμα του Ανωτάτου Δικαστηρίου των Η.Π.Α. "απομονωμένοι και ίσοι σημαίνει στην πράξη άνισοι" και στα δύο φύλα εκτός από τις ανθρώπινες φυλές, τερματίζοντας την πατροπαράδοτη πρακτική να διαχωρίζονται οι τουαλέτες σε ανδρικές και γυναικείες. Ωστόσο, κάνοντας μια παραχώρηση στην παράδοση, ορίζει ότι όταν βρίσκεται γυναίκα μέσα σε κάποια τουαλέτα, έχουν δικαίωμα να μπουν μόνο γυναίκες και, αντίστροφα, όταν βρίσκεται άνδρας, έχουν δικαίωμα να εισέλθουν μόνο άνδρες. Μια φωτεινή επιγραφή σε κάθε πόρτα τουαλέτας προσδιορίζει ποια από τις τρεις πιθανές καταστάσεις υπάρχει στο εσωτερικό της τουαλέτας:

- Άδεια
- Υπάρχουν μέσα γυναίκες
- Υπάρχουν μέσα άνδρες

Γράψτε τις επόμενες διαδικασίες στη γλώσσα προγραμματισμού που προτιμάτε: *woman\_wants\_to\_enter*, *man\_wants\_to\_enter*, *woman\_leaves*, *man\_leaves*. Μπορείτε να χρησιμοποιήσετε μετρητές και οποιεσδήποτε τεχνικές συγχρονισμού θέλετε.

51. Ξαναγράψτε το πρόγραμμα της Εικόνας 2-20, ώστε να μπορεί να χειριστεί περισσότερες από δύο διεργασίες.

52. Γράψτε ένα πρόβλημα παραγωγού-καταναλωτή, το οποίο να χρησιμοποιεί νήματα και μια κοινόχρηστη περιοχή προσωρινής αποθήκευσης. Όμως, μη χρησιμοποιήσετε σηματοφόρους ή άλλες μεθόδους συγχρονισμού για να προστατεύσετε τις κοινές δομές δεδομένων. Αφήστε απλώς κάθε νήμα να τις προσπελάζει όπως θέλει. Χρησιμοποιήστε τις κλήσεις *sleep* και *wakeup* για να χειριστείτε τις περιπτώσεις γεμάτης ή άδειας περιοχής προσωρινής αποθήκευσης. Παρατηρήστε πόσος χρόνος περνάει μέχρι να εμφανιστεί μοιραία συνθήκη συναγωνισμού. Για παράδειγμα, θα μπορούσατε να καθορίσετε ότι ο παραγωγός τυπώνει έναν αριθμό σε τακτά χρονικά διαστήματα. Μην τυπώνετε περισσότερους από έναν αριθμούς ανά λεπτό, επειδή η Ε/Ε θα μπορούσε να επηρεάσει τις συνθήκες συναγωνισμού.

53. Μπορούμε να προσθέσουμε μια διεργασία περισσότερο από μία φορές σε ουρά εκ περιτροπής, ώστε να της δώσουμε υψηλότερη προτεραιότητα. Η εκτέλεση πολλών παρουσιών (*multiple instances*) ενός προγράμματος, με την κάθε παρουσία να εργάζεται σε διαφορετικό τμήμα των δεδομένων, οδηγεί στο ίδιο αποτέλεσμα. Γράψτε αρχικά ένα πρόγραμμα που εξετάζει μια σειρά από αριθμούς για να διαπιστώσει ποιοι από αυτούς είναι πρώτοι. Στη συνέχεια επινοήστε μια μέθοδο για να επιτρέψετε σε πολλές παρουσίες του προγράμματος να εκτελούνται ταυτόχρονα και με τέτοιο τρόπο, ώστε να εξασφαλίζεται ότι κανένα ζεύγος διεργασιών δε θα επεξεργαστεί τον ίδιο αριθμό. Μπορείτε τελικά να διατρέξετε ταχύτερα τη σειρά εκτελώντας πολλές παρουσίες του ίδιου προγράμματος; Ας σημειωθεί ότι τα αποτελέσματά σας εξαρτώνται άμεσα από τις υπόλοιπες εργασίες που εκτελεί ο υπολογιστής σας. Αν χρησιμοποιείτε προσωπικό υπολογιστή, ο οποίος εκτελεί μόνο παρουσίες του συγκεκριμένου προγράμματος, μην περιμένετε να δείτε βελτίωση· ωστόσο, σε σύστημα που εκτελούνται πολλές άλλες διεργασίες, θα καταφέρετε να πάρετε μεγαλύτερο μερίδιο από το χρόνο της CPU, αν εκτελέσετε πολλές παρουσίες.