



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Κατανεμημένα Συστήματα

Δεύτερη Εργαστηριακή Άσκηση:

Επίλυση προβλήματος παραγωγού-καταναλωτή

Διδάσκων: Χρήστος Γκουμόπουλος

Εργαστηριακός Διδάσκοντας: Φακής Αλέξανδρος

Εισαγωγή

Τα νήματα επικοινωνούν κυρίως μέσω κοινόχρηστων (μοιραζόμενων) περιοχών στο χώρο διευθύνσεων. Αυτή η μέθοδος επικοινωνίας είναι εξαιρετικά αποδοτική αλλά επιτρέπει δύο τύπους σφαλμάτων: *συνθήκες ανταγωνισμού (race conditions)* και *συνθήκες προήγησης (precedence conditions)*. Τα σφάλματα αποτρέπονται μέσω του *συγχρονισμού (synchronization)*. Ο συγχρονισμός επιτυγχάνεται με ειδικά εργαλεία και μεθόδους που περιγράφονται παρακάτω.

Ωστόσο ο συγχρονισμός μπορεί να δημιουργήσει ανταγωνισμό μεταξύ των νημάτων, ο οποίος εμφανίζεται όταν δύο ή περισσότερα νήματα προσπαθούν συνδρομικά να αποκτήσουν πρόσβαση στις κοινές περιοχές μνήμης και το αναγκάζουν την Java να εκτελεί ένα ή περισσότερα νήματα πιο αργά είτε ακόμη και να αναστέλλει την εκτέλεσή τους. Το αποτέλεσμα αυτού του ανταγωνισμού για διαμοιραζόμενες περιοχές μνήμης παίζει σημαντικό ρόλο στην ζωτικότητα (liveness) των νημάτων που οδηγεί μερικές φορές στην λιμοκτονία (starvation) τους.

Συχνά τα νήματα πρέπει να συντονίζονται (συγχρονίζονται) την εκτέλεσή τους. Η πιο συνηθισμένη δομή συντονισμού είναι το *φρουρούμενο τμήμα (guarded block)*. Ένα τέτοιο τμήμα ξεκινά με μια συνθήκη (φρουρό, guard). Το νήμα αναστέλλει την εκτέλεσή του μέχρι να ικανοποιηθεί η συνθήκη αυτή.

Θα χρησιμοποιήσουμε το σχήμα αναμονής και ειδοποίησης για την υλοποίηση μιας εφαρμογής *Παραγωγού-Καταναλωτή (Producer-Consumer)*. Μια εφαρμογή τέτοιου τύπου διαμοιράζει δεδομένα μεταξύ δύο νημάτων: τον *παραγωγό*, που δημιουργεί δεδομένα, και τον *καταναλωτή*, που τα επεξεργάζεται. Ο συντονισμός είναι ουσιαστικός: Ο καταναλωτής δεν πρέπει να προσπαθεί να επεξεργαστεί δεδομένα πριν αυτά παραχθούν από τον παραγωγό. Ο παραγωγός πάλι δεν πρέπει να παράγει υπερβολικά πολλά δεδομένα πριν ο καταναλωτής τα επεξεργαστεί.

Σενάριο

Σε μια καφετέρια-εστιατόριο, δουλεύουν 4 άτομα, 1 μπουφετζής, 1 μάγειρας και 2 σερβιτόροι. Κάθε σερβιτόρος έχει διαφορετικό καθήκον. Ο πρώτος (έστω Α) είναι υπεύθυνος για να σερβίρει **μόνο** τα ροφήματα, ενώ ο δεύτερος (έστω Β) είναι υπεύθυνος για το σερβίρισμα **μόνο** των φαγητών. Υπάρχουν 2 είδη ροφημάτων, οι καφέδες και οι χυμοί, και 2 διαφορετικοί τύποι φαγητών, οι μακαρονάδες και τα τοστ. Όλα τα προϊόντα που είναι έτοιμα για σερβίρισμα, τοποθετούνται στον ίδιο πάγκο **με μια σειρά** από τον μάγειρα και τον μπουφετζή, και από το σημείο αυτό παραλαμβάνονται στην συνέχεια από τους σερβιτόρους, ανάλογα με την αρμοδιότητά τους. Για παράδειγμα, εάν στον πάγκο έχει ετοιμαστεί πρώτα ένας χυμός και μετά ένα τόστ, θα πρέπει να σερβιριστεί πρώτα ο χυμός από τον Α, πρώτου σερβιριστεί το τόστ από τον Β.

Παρόλα αυτά υπάρχουν κάποιοι περιορισμοί στην καφετέρια.

- Ένας σερβιτόρος δεν μπορεί να παραλάβει κάτι από τον πάγκο εάν το προϊόν αυτό δεν αντιστοιχεί στο καθήκον του, και στην προκειμένη περίπτωση θα πρέπει να το αγνοήσει, φεύγοντας από τον πάγκο.
- Εάν ο πάγκος είναι κενός ο σερβιτόρος που έφτασε σε αυτόν πρώτος, θα πρέπει να περιμένει έως ότου ετοιμαστεί κάποιο ρόφημα ή φαγητό. Εάν μπορεί να σερβίρει το προϊόν που ετοιμάστηκε, τότε το παίρνει από τον πάγκο. Σε αντίθετη περίπτωση, το αγνοεί και απομακρύνεται από τον πάγκο.
- Εάν ο πάγκος είναι γεμάτος, τότε κανένα προϊόν δεν μπορεί να τοποθετηθεί σε αυτόν. Σε αυτήν την περίπτωση θα πρέπει αυτός που το έχει ετοιμάσει, να περιμένει μέχρι να αδειάσει κάποια θέση στον πάγκο.

Για την εκτέλεση μιας ενέργειας από έναν υπάλληλο απαιτείται κάποιος χρόνος καθυστέρησης. Για παράδειγμα ένας σερβιτόρος για να σερβίρει ένα προϊόν, σπαταλάει κάποιο χρόνο πριν επανέλθει σε αυτόν για να παραλάβει νέο προϊόν. Το ίδιο ισχύει και εάν απομακρυνθεί από τον πάγκο γιατί δεν είναι σε θέση να σερβίρει το επόμενο έτοιμο προϊόν. Αντίστοιχα, ο μπουφетζής και ο μάγειρας χρειάζονται κάποιο χρόνο προετοιμασίας πριν θέσουν ένα προϊόν έτοιμο προς σερβίρισμα.

Θα πρέπει να δημιουργήσετε αντίστοιχο πρόγραμμα σε Java με την χρήση συγχρονισμού νημάτων για να επιλύσετε το παραπάνω πρόβλημα παραγωγού-καταναλωτή.

Τεχνικές Λεπτομέρειες

- Για τον πάγκο της καφετέριας χρησιμοποιήστε έναν κυκλικό πίνακα 10 θέσεων. Τα προϊόντα στην καφετέρια εισέρχονται και εξέρχονται με σειρά FIFO (First In First Out), δηλαδή ότι εισέρχεται πρώτο στον πίνακα θα πρέπει να εξέρχεται και πρώτο. Για την παραπάνω λειτουργία κρατήστε κατάλληλους μετρητές για την εισαγωγή και εξαγωγή προϊόντων στον πίνακα. Εάν τα προϊόντα που βρίσκονται στην αρχή του πίνακα έχουν σερβιριστεί και οι τελευταίες θέσεις είναι γεμάτες, τότε τα νέα προϊόντα θα πρέπει να τοποθετηθούν στις πρώτες θέσεις του πίνακα.
- Δημιουργήστε random χρόνους καθυστέρησης ανάμεσα στις εκτελέσεις ενεργειών από τον ίδιο υπάλληλο με βάση τις δικές σας προτιμήσεις.
- Χαρακτηρίστε με ένα δικό σας τρόπο τα διάφορα προϊόντα της καφετέριας. Μπορείτε για παράδειγμα να ορίσετε διαφορετικούς αριθμούς για το κάθε προϊόν.
- Εκτελέστε το παραπάνω πρόβλημα για συγκεκριμένο αριθμό προϊόντων. Για παράδειγμα, μπορείτε να υποθέσετε πως υπάρχουν 20 παραγγελίες για φαγητό

και 20 για ροφήματα (δώστε στο όριο των επαναλήψεων παραγωγής και κατανάλωσης hard coded τιμές με βάση τους αριθμούς που ορίσατε).

- Εμφανίστε κατάλληλα μηνύματα για να εμφανίζετε το κάθε συμβάν. Για παράδειγμα εμφανίστε μήνυμα για την παραλαβή ενός προϊόντος από κάποιον σερβιτόρο ή την απομάκρυνσή του από τον πάγκο χωρίς κάποιο προϊόν (εάν το επόμενο προς σερβίρισμα προϊόν δεν είναι στην αρμοδιότητά του). Για τον παραπάνω σκοπό, μπορείτε να δώσετε ονόματα στα thread σας ώστε να τα αναγνωρίζετε κατά την εκτέλεσή τους.
- Δημιουργήστε μόνο μια κλάση παραγωγού (για τον μπουφετζή και τον μάγειρα) και μια για τον καταναλωτή (για τους σερβιτόρους) και δώστε κατάλληλες τιμές στα αντικείμενα για να ξεχωρίσετε τις αρμοδιότητες του καθενός.

Οδηγίες Παράδοσης :

Τελική ημερομηνία παράδοσης της εργασίας: **31/03/2018**.

Θα πρέπει να σταλεί ένα αρχείο rar/zip με όνομα ΑριθμόςΜητρώου_Lab02. Στο αρχείο θα περιέχονται τα εξής:

- Μια αναφορά σε μορφή pdf με οθόνες εκτέλεσης του προγράμματός σας. Στην αρχή του αρχείου θα αναγράφεται ο αριθμός μητρώου και το ονοματεπώνυμο σας.
- Τα αρχεία .java των project σας με σχολιασμό στα πιο βασικά σημεία.
- Η υποβολή κοινών απαντήσεων από διαφορετικούς φοιτητές δεν επιτρέπεται και θεωρείται ως ΑΝΤΙΓΡΑΦΗ. Η αντιγραφή έχει ως αποτέλεσμα το ΜΗΔΕΝΙΣΜΟ ΤΗΣ ΕΡΓΑΣΙΑΣ ΣΥΝΟΛΙΚΑ.