

ΑΣΚΗΣΗ 2

Μάθημα: ΗΥ-345 Λειτουργικά Συστήματα
Χειμερινό Εξάμηνο 2025

Σε αυτήν την άσκηση θα εξοικειωθείτε με:

- τη δημιουργία και διαχείριση **threads** (POSIX threads),
- και τη χρήση **POSIX semaphores** για τον συγχρονισμό πολλαπλών threads που συνεργάζονται σε κοινό περιβάλλον.

Περιγραφή — Εστιατόριο και Ομάδες Πελατών

Θεωρούμε ένα εστιατόριο όπου οι πελάτες καταφθάνουν σε **ομάδες** (groups).

Κάθε ομάδα επιθυμεί να καθίσει **μαζί στο ίδιο τραπέζι** — δεν επιτρέπεται διαχωρισμός μελών.

- Το εστιατόριο διαθέτει **N τραπέζια**, το καθένα με χωρητικότητα έως **X** ατόμων.
- Δημιουργούνται **G ομάδες** πελατών (threads).
- Κάθε ομάδα αποτελείται από **1 έως X** άτομα (τυχαία τιμή).

Ο σερβιτόρος (ένα ξεχωριστό thread) είναι υπεύθυνος να παρακολουθεί την κατάσταση των τραπεζιών και να **αναθέτει ομάδες σε διαθέσιμα τραπέζια**.

Λογική του Προγράμματος

Συμπεριφορά Ομάδας (Group Thread)

Κάθε ομάδα είναι ένα ανεξάρτητο **thread** και εκτελεί τα παρακάτω βήματα:

1. **Αφίξη στο εστιατόριο** και αναφορά μεγέθους (τυχαίο από 1–X).
 - ο Η αφίξεις των ομάδων θα πρέπει να είναι σε τυχαία σειρά.
2. **Αναμονή** μέχρι να υπάρχει διαθέσιμο τραπέζι που να χωράει όλα τα μέλη της.
 - ο Η αναμονή υλοποιείται με χρήση semaphore το οποίο «ξυπνάει» ο σερβιτόρος.
3. **Κάθισμα** στο τραπέζι που της ανατέθηκε.
4. **Φαγητό** για τυχαίο χρονικό διάστημα (5–15 δευτερόλεπτα).
5. **Αποχώρηση** και απελευθέρωση των θέσεων στο τραπέζι.
6. Ενημέρωση του waiter ότι το τραπέζι είναι ελεύθερο.

Συμπεριφορά Σερβιτόρου (Waiter Thread)

Ο σερβιτόρος είναι επίσης ένα thread και εκτελεί **συνεχή βρόχο λειτουργίας**:

1. Παρακολουθεί την ουρά αναμονής (queue) με ομάδες που δεν έχουν κάτσει ακόμα.
2. Για κάθε ομάδα που περιμένει, ελέγχει αν υπάρχει τραπέζι με αρκετές ελεύθερες θέσεις.
3. Αν βρεθεί κατάλληλο τραπέζι:
 - ο «Κρατάει» τις θέσεις (ενημερώνει τις μεταβλητές occupied στο τραπέζι).
 - ο «Ξυπνάει» το συγκεκριμένο group.
4. Αν καμία ομάδα δεν μπορεί να καθίσει, περιμένει μέχρι να αλλάξει κάτι (όταν ένα group αποχωρήσει).
5. Το πρόγραμμα τερματίζει όταν **όλες οι ομάδες έχουν φάει και αποχωρήσει**.

Συγχρονισμός — Semaphores και Mutexes

Η λύση πρέπει να χρησιμοποιεί **και semaphores και mutexes**.

Παράμετροι Εκτέλεσης

Το πρόγραμμα πρέπει να δέχεται **τρεις παραμέτρους από τη γραμμή εντολών**:

`./restaurant <num_tables> <num_groups> <table_capacity>`

Παράδειγμα Εκτέλεσης για 2 tables, 4 groups, 6 each table capacity

```
./restaurant 2 4 6
```

```
[GROUP 3] Arrived with 3 people (after 2 sec)
```

```
[WAITER] Assigned group 3 (size=3) to Table 0 (3/6 occupied)
```

```
[WAITER] Groups waiting: None
```

```
[GROUP 3] Seated at Table 0 with 3 people
```

```
[GROUP 3] Eating (3 people) for 8 seconds...
```

```
[GROUP 1] Arrived with 2 people (after 6 sec)
```

```
[GROUP 0] Arrived with 6 people (after 6 sec)
```

```
[WAITER] Assigned group 1 (size=2) to Table 0 (5/6 occupied)
```

```
[WAITER] Groups waiting: 0
```

```
[GROUP 1] Seated at Table 0 with 2 people
```

```
[GROUP 1] Eating (2 people) for 14 seconds...
```

```
[WAITER] Assigned group 0 (size=6) to Table 1 (6/6 occupied)
```

```
[WAITER] Groups waiting: None
```

```
[GROUP 0] Seated at Table 1 with 6 people
```

```
[GROUP 0] Eating (6 people) for 13 seconds...
```

```
[GROUP 2] Arrived with 3 people (after 8 sec)
```

```
[GROUP 3] Left Table 0 (2/6 occupied)
```

```
[WAITER] Assigned group 2 (size=3) to Table 0 (5/6 occupied)
```

```
[WAITER] Groups waiting: None
```

```
[GROUP 2] Seated at Table 0 with 3 people
```

```
[GROUP 2] Eating (3 people) for 7 seconds...
```

```
[GROUP 2] Left Table 0 (2/6 occupied)
```

```
[GROUP 0] Left Table 1 (0/6 occupied)
```

```
[GROUP 1] Left Table 0 (0/6 occupied)
```

```
All groups have left. Closing restaurant.
```

Τι θα αξιολογηθεί

1. Ορθότητα συγχρονισμού (χωρίς race conditions).
2. Ορθή χρήση mutexes και semaphores.
3. Ρεαλιστική προσομοίωση (σωστός συγχρονισμός waiter-groups).
4. Να προσέξετε ώστε το output στη κονσόλα να είναι ευανάγνωστο και να παρέχει τις πληροφορίες που χρειάζονται.
5. Πληρότητα σχολίων και README.

Παράδοση

Η παράδοση της άσκησης θα γίνει μέσω E-Learn.

Θα πρέπει να φτιάξετε ένα zip αρχείο το οποίο θα περιέχει:

- Τον κώδικα σας, ο οποίος θα περιέχει σχόλια που να τον εξηγούν.
- Ένα Makefile το οποίο με την χρήση των εντολών:
 - make all να παράγεται το εκτελέσιμο
 - make clean να διαγράφονται τα αρχεία που δημιουργήθηκαν.
- Ένα README στο οποίο θα περιγράφετε την υλοποίησή σας, και ότι σημειώσεις θεωρείτε ότι είναι σημαντικές.

Μία προτεινόμενη δομή για τα αρχεία που θα παραδώσετε:

- o main.c
- o restaurant.c
- o restaurant.h
- o Makefile
- o README.md

Παρατηρήσεις

- Η άσκηση αυτή είναι καθαρά ατομική. Τα παραδοτέα αρχεία θα περαστούν από σύστημα ανίχνευσης αντιγραφών. Τυχούσες αντιγραφές συνεπάγονται με αυτόματο μηδενισμό.
- Στην αρχή του κώδικά σας, συμπεριλάβετε σε σχόλια το ονοματεπώνυμο και τον αριθμό μητρώου σας.
- Κρατήστε περιποιημένο τον κώδικα σας και προσθέστε σχόλια που να εξηγούν τη λειτουργία του όπου χρειάζεται.
- Η χρήση κώδικα ο οποίος έχει παραχθεί από AI απαγορεύεται ρητά