

2^η Άσκηση: Βελτιωμένη Προσομοίωση Ουράς Τράπεζας με χρήση Λιστών-Δεικτών

Οι νέες προδιαγραφές της Ουράς της Τράπεζας

Το αντικείμενο της 2^{ης} άσκησης είναι να επεκτείνετε το πρόγραμμα προσομοίωσης της ουράς πελατών Τράπεζας της 1^{ης} άσκησης. Σκοπός μας είναι :

- Να επιτραπούν οι αποδοτικές διαγραφές (αποχωρήσεις πελατών) από οποιοδήποτε σημείο της ουράς. Για αυτό τον λόγο η υλοποίηση της ουράς απαιτείται να γίνει με δείκτες.
- Η υποστήριξη προτεραιότητας AMEA. Τα AMEA θα μπαίνουν σε ξεχωριστή ουρά που εξυπηρετείται κατά προτεραιότητα. Αφού έρθει πελάτης εξετάζουμε αν είναι AMEA και μπαίνει στην αντίστοιχη ουρά (απαιτείται η πιθανότητα άφιξης AMEA ως είσοδος στο πρόγραμμα προσομοίωσης που είναι πραγματικός αριθμός μεταξύ 0 και 1).
- Όταν έρχεται πελάτης να υπολογίζεται και να εισάγεται στην δομή του η σειρά προτεραιότητάς του (θέση εξυπηρέτησης). Η θέση πρέπει να προστεθεί στο ήδη υπάρχον TStoixeiouOuras. Προσοχή να ληφθεί υπόψη η ύπαρξη ουράς AMEA.
- Να καλυφθούν αποχωρήσεις από την ουρά της κανονικής εξυπηρέτησης. Από την ουρά AMEA δεν γίνονται αποχωρήσεις (για ευκολία προγραμματισμού). Σε κάθε βήμα της προσομοίωσης υπάρχει πιθανότητα αποχώρησης για κάθε πελάτη της κανονικής ουράς, όμως μπορεί να αποχωρήσει το πολύ ένας πελάτης. Όταν κάποιος αποχωρήσει διαγράφεται από την ουρά. Μετά από αποχώρηση μειώνεται η θέση όσων έπονται κατά 1 (απαιτείται η πιθανότητα αποχώρησης ως είσοδος στο πρόγραμμα προσομοίωσης που είναι πραγματικός αριθμός μεταξύ 0 και 1).
- Να διατηρείτε την ακριβή θέση εξυπηρέτησης για κάθε πελάτη στην ουρά ανά πάσα στιγμή. Για τον υπολογισμό της θέσης:
 - ο Στην έναρξη της εξυπηρέτησης ενός πελάτη (απομάκρυνση από την ουρά του) μειώνεται η θέση όλων των υπόλοιπων πελατών κατά 1. Προσοχή στην περίπτωση εξυπηρέτησης AMEA.
 - ο Μετά από άφιξη AMEA αυξάνεται η θέση των ατόμων της κανονικής ουράς κατά 1.
 Για τον έλεγχο του σωστού υπολογισμού της θέσης να εκτυπώνεται επιπλέον η θέση πελάτη κατά α) την είσοδο του στην ουρά στο αρχείο Customers.txt και β) την απομάκρυνση του στο αρχείο Statistics.txt (κατά την έξοδο πρέπει να είναι πάντα 1).

Αρχικός Σχεδιασμός

Στο πρόγραμμα-σκελετό BankQueue , που συνοδεύει την εκφώνηση, δηλώνονται

```
typedef QueueStruct * BankOurasPtr;           // στο BankQueueOA.h
                                              //τα υπόλοιπα στο BankQueueOA.c

typedef struct typos_komvou *typos_deikti;
struct typos_komvou{
    TStoixeiouOuras dedomena; /* τα δεδομένα του κόμβου */
    typos_deikti epomenos; /*δείχνει στον επόμενο κόμβο της λίστας*/
};
typedef struct QueueStruct{
    typos_deikti Head;         /* δείχνει τον πρώτο κόμβο της λίστας */
    typos_deikti Last;        /* δείχνει τον τελευταίο κόμβο της λίστας*/
    int megethos;             /* μήκος ουράς */
} QueueStruct;
```

Ζητείται:

1. Να επεκτείνετε τον τύπο και τις πράξεις του TStoixeiouOuras της προσομοίωσης της 1^{ης} άσκησης με την θέση πελάτη.
2. Να υλοποιήσετε και να δοκιμάσετε τις ακόλουθες πράξεις με τον νέο TStoixeiouOuras :

```
BankOurasPtr BankOura_dimiourgia();
int BankOura_keni(BankOurasPtr oura);
void BankOura_prothesi(BankOurasPtr oura, TStoixeiouOuras stoixeiio,
                      int *yperxeilisi);
void BankOura_apomakrynsh(BankOurasPtr oura, TStoixeiouOuras *stoixeiio,
                          int *ypoxeilisi);

int BankOura_megethos(BankOurasPtr oura);
void BankOura_apoxwrhsh(BankOurasPtr oura, float pithanothta);
void BankOura_updateThesi(BankOurasPtr oura, int ShiftThesi);
void BankOura_katastrofi(BankOurasPtr *oura);
```

- (Ακριβή συμβόλαια (contracts) για τις ανωτέρω πράξεις θα βρείτε στο `BankQueueOA.c`).
3. Να επεκτείνετε το πρόγραμμα `simulation.c` της 1^{ης} άσκησης, ώστε να υλοποιεί τις ανωτέρω προδιαγραφές. Επίσης στο τέλος της προσομοίωσης να καταστρέφεται η ουρά.

Οδηγίες Σχεδίασης και Ανάπτυξης Προγράμματος

Το πρόγραμμά σας πρέπει να είναι οργανωμένο σε ενότητες (modules) και σε πρόγραμμα-πελάτη. Να χρησιμοποιήσετε το `BankQueueOA.h` και να υλοποιήσετε τις πράξεις στο `BankQueueOA.c`. Για την υλοποίηση των πράξεων μπορείτε να χρησιμοποιήσετε τα προγράμματα που σας έχουν ήδη δοθεί για τον ΑΤΔ Ουρά και ΑΤΔ Λίστα.

Παραδοτέα και Οδηγίες Παράδοσης

1. Πηγαίος κώδικας, δηλαδή `simulation.c`, `BankQueueOA.h`, `BankQueueOA.c`, `TStoixeiouOuras.h`, `TStoixeiouOuras.c`. (Τα `main` των δοκιμών δεν παραδίδονται)
2. Τεκμηρίωση (μέγιστο 1 σελίδα) Σύντομο κείμενο (txt, word, pdf) με την εξής δομή:
 - Τα στοιχεία σας: (Όνομα-Επώνυμο-ΑΜ)
 - Λειτουργικότητα: Να περιγράψετε τι κάνει το πρόγραμμά σας (μπορεί να κάνει περισσότερα ή και λιγότερα από τα ζητούμενα της άσκησης).
 - Οδηγίες Χρήσης του προγράμματος σας: π.χ. Διάταξη δεδομένων εισόδου.
 - Περιβάλλον Υλοποίησης και Δοκιμών: πχ. Αναπτύχθηκε σε Dev C++ σε περιβάλλον Windows XP, δοκιμάστηκε επίσης σε gcc και Unix.

Το αρχείο τεκμηρίωσης μαζί με τα πηγαία αρχεία του προγράμματος να τα βάλετε σε έναν φάκελο (directory), τον οποίο θα συμπιέσετε (zip, rar) και θα ανεβάσετε στο eclass. (Όσοι αναπτύσσετε σε dev να συμπεριλάβετε και το `.dev`). Προσοχή ανεβάστε το στην κατάλληλη κατηγορία υλοποίησης (Dev-C++ ή gcc). **Ημερομηνία Παράδοσης 2/5/2011.**

Τρόπος Αξιολόγησης

Οι ασκήσεις είναι **ατομικές** και θα ελεγχθούν για ομοιότητες χρησιμοποιώντας ειδικό σύστημα εντοπισμού ομοιοτήτων/αντιγραφών. Σε περίπτωση μεγάλης «ομοιότητας» όλες οι «παρόμοιες» ασκήσεις θα μηδενιστούν. Θα αξιολογηθούν η λειτουργικότητα, η δομή και η τεκμηρίωση του προγράμματος. Αναλυτικά:

Λειτουργικότητα (80/100)

- | | |
|--|---------|
| 1. <code>BankOura_dimiourgia</code> | (05/80) |
| 2. <code>BankOura_keni</code> και <code>BankOura_megethos</code> | (05/80) |
| 3. <code>BankOura_prothesi</code> | (10/80) |
| 4. <code>BankOura_apomakrynsh</code> | (10/80) |
| 5. <code>BankOura_katastrofi</code> | (10/80) |
| 6. <code>BankOura_apoxwrhsh</code> | (10/80) |
| 7. <code>BankOura_UpdateThesi</code> | (10/80) |
| 8. Επέκταση τύπου στοιχείου και εκτυπώσεις | (05/80) |
| 9. Αρχεία <code>Customer.txt</code> και <code>Statistics.txt</code> (επέκταση θέσης) | (05/80) |
| 10. Επέκταση <code>Simulation.c</code> | (10/80) |

Δομή (15/100)

- | | |
|--|--------|
| Οργάνωση σε Ενότητες (.h, .c) και πρόγραμμα πελάτη | (5/15) |
| Ολική Απόκρυψη Υλοποίησης Ουράς | (5/15) |
| Δομημένο Πρόγραμμα-πελάτη (μορφοποίηση, σχόλια, κλπ) | (5/15) |

Τεκμηρίωση και Παρουσίαση (5/100)

ΠΡΟΣΟΧΗ: Για να αξιολογηθεί το πρόγραμμά σας (έστω για την δομή του) πρέπει τουλάχιστον να μεταγλωττίζεται. Αν δεν μεταγλωττίζεται δεν παίρνει βαθμό. Πριν παραδώσετε το πρόγραμμά σας δοκιμάστε το μια τελευταία φορά και βεβαιωθείτε ότι παραδίδετε τα σωστά αρχεία.

Εκπαιδευτικοί στόχοι

1. Χρήση και υλοποίηση του ΑΤΔ Λίστα
2. Χειρισμός Δεικτών, Δυναμική Διαχείριση Μνήμης