

HY-345 ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ  
ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2022  
ΑΣΚΗΣΗ 2

Ανάθεση : 25/10

Παράδοση : 8/11

Ωρες γραφείου : Δευτέρα και Παρασκευή στις 14:00-16:00 στην αίθουσα B210

Σκοπός της συγκεκριμένης άσκησης είναι η εξοικείωση με νήματα (Posix Threads). Οι βασικές λειτουργίες των προγραμμάτων που θα υλοποιήσετε είναι δημιουργία, συγχρονισμό και καταστροφή νημάτων.

### Άσκηση - **Narrow road**

Θεωρήστε έναν αρκετά στενό δρόμο, χτισμένο στην κορυφή ασταθών θεμελίων στο Ηράκλειο της Κρήτης. Ο δρόμος είναι τόσο στενός ώστε να χωράει έναν περαστικό αλλά όχι δύο περαστικούς στο πλάτος. Σε περίπτωση που δύο περαστικοί ερχόμενοι από αντίθετες κατευθύνσεις συναντηθούν, ένας (ή περισσότεροι) πρέπει να κάνει στην άκρη. Οι κανόνες έχουν ως εξής.

Υποθέστε ότι οι πεζοί φορούν **κόκκινα** ή **μπλε** ρούχα μόνο.

- Όλοι οι πεζοί επιθυμούν να διασχίσουν τον δρόμο. Όμως ο δρόμος χωράει έναν πεζό σε κάθε θέση. Για να διασχίσει τον δρόμο ο πεζός πρέπει να περάσει από όλες τις θέσεις. Ένας πεζός μπορεί να κάνει ένα βήμα μπροστά (στην επόμενη θέση) ή στο πλάι (για να κάνει στην άκρη) μόνο αν υπάρχει κενή θέση. Ένας πεζός δεν μπορεί να προχωρήσει αν ο μπροστινός του δεν έχει μετακινηθεί ακόμα.
- Στην περίπτωση που συναντηθούν δύο πεζοί ερχόμενοι με διαφορετικές κατευθύνσεις, θα περάσουν πρώτα εκείνοι των οποίων το χρώμα υπερτερεί στο δρόμο από την κατεύθυνση που έχει τους περισσότερους.
- Στην συνέχεια, θα περάσουν οι πεζοί της άλλης κατεύθυνσης με το ίδιο κυρίαρχο χρώμα. Πλέον στο δρόμο έχουμε μόνο ένα χρώμα πεζών.
- Ακολουθούν οι πεζοί της πλευράς που έχει τους περισσότερους πεζούς.
- Τέλος, η άλλη πλευρά με τους υπόλοιπους.

Μπορείτε να υποθέσετε ότι υπάρχει ένας τροχονόμος που διασφαλίζει την ομαλή κυκλοφορία, αφήνοντας τους πεζούς που μπορούν να περάσουν και σταματώντας εκείνους που πρέπει να παραχωρήσουν προτεραιότητα.

- Πρέπει να διατηρείται η ουρά προτεραιότητας από την κάθε πλευρά. Δεν μπορεί κάποιος πεζός ίδιου χρώματος και κατεύθυνσης να προσπεράσει τον επόμενο του και να βγει νωρίτερα από τον δρόμο.

Υλοποιήστε ένα πρόγραμμα που επιτρέπει τους περαστικούς να περνούν τον δρόμο.

Συνοπτικά η λύση σας θα πρέπει να

- Αποφεύγει το starvation (κανένας περαστικός να μην περιμένει για πάντα στο δρόμο).
- Αποφεύγει την διαμάχη των πεζών και την διακοπή της κυκλοφορίας.
- Τηρεί την προτεραιότητα των πεζών. Από την κάθε πλευρά, ο πρώτος που μπαίνει θα βγει πρώτος κοκ. (FIFO)

Για την υλοποίηση θα χρησιμοποιήσετε system calls της ομάδας threads (POSIX Threads) για τη δημιουργία πεζών, και σημαφόρους για τον συγχρονισμό τους.

#### Αρχικοποίηση πεζών

1. Η αρχικοποίηση των πεζών-thread περιλαμβάνει ανάθεση **id**, τον ορισμό της **κατεύθυνσης** και το **χρώμα** των ρούχων τους (**μπλε** ή **κόκκινο**).
2. Κάθε πεζός κινείται προς μια κατεύθυνση κάθε φορά, ανατολικά ή δυτικά. Ορίστε την κατεύθυνση των πεζών καθώς και το χρώμα των ρούχων που φορούν κατά την δημιουργία τους τυχαία. Μπορείτε να χρησιμοποιήσετε την βιβλιοθήκη random(3) για ευκολία.
3. Κάθε στιγμή, οι πεζοί μπορούν να βρίσκονται στο **δρόμο** ή στο **πεζοδρόμιο** (κάνοντας στην άκρη για να περάσουν άλλοι πεζοί).

#### Αρχικοποίηση δρόμου

Το μήκος του δρόμου είναι ο αριθμός των πεζών που χωράει σε παράταξη.

Τοποθετήστε τους πεζούς τυχαία στον δρόμο.

Τα πιθανά σενάρια :

1. ο δρόμος να είναι γεμάτος μόνο με άτομα που θέλουν να πάνε δυτικά (σχεδόν απίθανο σενάριο).
2. ο δρόμος να είναι γεμάτος μόνο με άτομα που θέλουν να πάνε ανατολικά (σχεδόν απίθανο σενάριο).
3. ο δρόμος να είναι γεμάτος με άτομα που θέλουν να πάνε και προς τις δυο κατευθύνσεις.

**Προσοχή:** Η εισαγωγή των πεζών θα γίνεται από την αντίθετη πλευρά από αυτήν που επιθυμούν να προχωρήσουν.

Παράδειγμα πεζού που θέλει να κινηθεί προς τα δυτικά.

ID: 4,Color: RED ,Direction: West,<

Θα γίνει η εισαγωγή του από τα δεξιά του δρόμου και θα κινηθεί προς τα αριστερά(δυτικά).

Ομοίως για ID 1, ID 0.

```
-1 || -1 || -1 || -1 || -1 || -1 || -1 || -1 || -1 || -1 || -1 || -1 || -1 || -1 || -1 || -1 || -1 || -1 || -1 || -1 || -1 ||  
2 > || 3 > || 5 > || 6 > || 8 > || 9 > || 10 > || 12 > || 13 > || 18 > || < 20 || < 19 || < 17 || < 16 || < 15 || < 14 || < 11 || < 7 || < 4 || < 1 || < 0 ||
```

## Flow/Grading/Debugging

Εξασφαλίστε την ομαλή κυκλοφορία του δρόμου τυπώνοντας κατάλληλα μηνύματα κατά την αρχικοποίηση του δρόμου, την κίνηση των πεζών στο δρόμο (αλλαγή θέσεων) αλλά και κατά την έξοδο των πεζών από τον δρόμο.

Προτείνεται να τυπώνεται σε κάθε αλλαγή θέσης ολόκληρος ο δρόμος. Να τυπώνεται δηλαδή το αναγνωριστικό id του κάθε πεζού καθώς και < ή > για ορισμό κατεύθυνσης ή W (west), E (east), D (dytika), A (Anatolika).

```
0 > || 1 > || 4 > || 8 > || 9 > || 10 > || 14 > || 15 > || 16 > || 20 > || < 19 || < 18 || < 17 || < 13 || < 12 || < 11 || < 7 || <
```

Το πρόγραμμα σας θα πρέπει να δέχεται από την γραμμή εντολών την παράμετρο p, τον αριθμό των πεζών.

Μπορείτε να χρησιμοποιήσετε τις έτοιμες συναρτήσεις που σας δίνονται για χρωματισμό του output για καλύτερο debugging αλλά και διόρθωση.

```
#define RED 0  
#define BLUE 1  
void color(int c){  
    if (c)blue();  
    else red();  
}  
void red () {  
    printf("\033[1;31m");  
}  
void blue () {  
    printf("\033[0;34m");  
}  
void reset () {  
    printf("\033[0m");  
}
```

Παράδειγμα δρόμου μετά την αρχικοποίηση.

```
0 > || 1 > || 4 > || 8 > || 9 > || 10 > || 14 > || 15 > || 16 > || 20 > || < 19 || < 18 || < 17 || < 13 || < 12 || < 11 || < 7 || < 6 || < 5 || < 3 || < 2 ||
```

Παράδειγμα εκτέλεσης με 21 πεζούς.

./nr -p 21

Κυρίαρχο χρώμα μπλε

Κυρίαρχη κατεύθυνση >

```
||01 >||04 >|| ||06 >|| || ||10 >||13 >||14 >||16 >||17 >||18 >|| || || || || || || || || ||  
|| || ||05 >|| ||08 >|| ||12 >|| || || || || || ||19 >||20 >||< 15||< 11||< 09||< 07||< 03||< 02||
```

Μετακίνηση ID 13,20 ενώ τα υπόλοιπα ακίνητα.

```
|| ||02 >|| ||03 >||05 >|| ||07 >|| ||08 >|| ||11 >|| ||13 >|| || || || ||20 >|| || ||  
|| || || || || || || ||09 >|| || ||12 >|| ||14 >|| || || || ||< 19|| || ||  
|| ||02 >|| ||03 >||05 >|| ||07 >|| ||08 >|| || ||11 >|| ||13 >|| || || || ||20 >|| ||  
|| || || || || || || ||09 >|| || ||12 >|| ||14 >|| || || || ||< 19|| || ||
```

**Παράδειγμα λάθους χειρισμού του δρόμου.**

Βλέπουμε σε σειρά τα κόκκινα > [6,9,10,13,15,17,19]

Βγαίνουν τα 17,19.

Όμως ανάμεσα στο 10 - 15, υπήρχε το 13 το οποίο χάθηκε ή προσπέρασε.

Το 13 δεν θα έπρεπε να έχει εξαφανιστεί από τον δρόμο, γι' αυτό η λύση είναι λάθος.

```
|| ||06 >||09 >|| ||10 >||13 >|| ||15 >||17 >||19 >|| || || || || || || || || || || || || | |
||05 >|| || || || || || ||14 >|| || || ||< 20||< 18||< 16||< 11||< 08||< 07||< 04||< 02||< 01||< 00||  
|| || ||06 >|| || || ||10 >|| || || ||15 >|| || || || || || || || || || || || ||  
||05 >|| || || || || || ||14 >|| || || || ||< 20||< 18||< 16||< 11||< 08||< 07||< 04||< 02||< 01||< 00||
```

## Παρατηρήσεις

1. Η άσκηση είναι ατομική. Τυχόν αντιγραφές μπορούν να ανιχνευθούν εύκολα από κατάλληλο πρόγραμμα και να μηδενιστούν. Συμπεριλάβετε το όνομα σας, τον αριθμό μητρώου σας, και το username σας σε όλα τα αρχεία.
2. Η μεταγλώττιση πρέπει να γίνει με την παράμετρο **-pthread** ώστε ο linker να μπορεί να βρει τις συναρτήσεις στην pthread βιβλιοθήκη.
3. Κατασκευάστε ένα αρχείο Makefile, έτσι ώστε πληκτρολογώντας make all να γίνεται η μεταγλώττιση (compilation) του προγράμματος και να παράγεται το εκτελέσιμο αρχείο. Επίσης πληκτρολογώντας make clean να καθαρίζονται όλα τα περιττά αρχεία, και να μείνουν μόνο τα αρχεία που χρειάζονται για τη μεταγλώττιση.
4. Γράψτε ένα αρχείο README.txt, το πολύ 30 γραμμών, με επεξηγήσεις για τον τρόπο υλοποίησης.
5. Τοποθετήστε σε ένα κατάλογο όλα τα αρχεία που χρειάζονται για την άσκηση 2. Παραδώστε τα παραπάνω αρχεία χρησιμοποιώντας το πρόγραμμα turnin (πληκτρολογήστε turnin assignment\_2@hy345 directory\_name από τον κατάλογο που περιέχει τον κατάλογο directory\_name με τα αρχεία της άσκησης).
6. Το πρόγραμμά σας μπορεί να γραφεί μόνο για λειτουργικά συστήματα που υπάρχουν στα εργαστήρια (μηχανήματα σχολής) και σε γλώσσα προγραμματισμού C χρησιμοποιώντας system calls της ομάδας threads (POSIX Threads).
7. Για καλύτερη κατανόηση - ανάγνωση του προγράμματος μπορείτε να χρησιμοποιήσετε την συνάρτηση sleep(3) π.χ. προσπέλαση δρόμου, είσοδο, πορεία πεζών.
8. Ο έλεγχος και ο συγχρονισμός ταυτόχρονα εκτελούμενων διεργασιών μπορεί να επιτευχθεί με την χρήση σημαφόρων (semaphore).